Proceedings

IWSLT 2012

International Workshop on Spoken Language Translation

HKUST 6–7 December 2012
The Hong Kong University of Science and Technology

hltc.cs.ust.hk/iwslt

Proceedings of the

International Workshop on Spoken Language Translation

December 6 and 7, 2012
Hong Kong

Edited by

Eiichiro Sumita
Dekai Wu
Michael Paul
Chengqing Zong
Chiori Hori

Table of Contents

Evaluation Campaign

Technical Papers

Foreword

The International Workshop on Spoken Language Translation (IWSLT) is an annually-held scientific workshop, associated with an open evaluation campaign on spoken language translation, where both scientific papers and system descriptions are presented. The 9th International Workshop on Spoken Language Translation takes place in Hong Kong on December 6 and 7, 2012.

IWSLT includes scientific papers in dedicated technical sessions, with both oral or poster presentations. The contributions cover theoretical and practical issues in the field of Machine Translation (MT) in general, and Spoken Language Translation (SLT) in particular:

Speech and text MT
Integration of ASR and MT
MT and SLT approaches
MT and SLT evaluation
Language resources for MT and SLT
Open source software for MT and SLT
Adaptation in MT
Simultaneous speech translation
Speech translation of lectures
Efficiency in MT
Stream-based algorithms for MT
Multilingual ASR and TTS
Rich transcription of speech for MT
Translation of on-verbal events

Submitted manuscripts were carefully peer-reviewed by two members of the program committee and papers were selected based on their technical merit and relevance to the conference. The large number of submissions as well as the high quality of the submitted papers indicates the interest on Spoken Language Translation as a research field and the growing interest in these technologies and their practical applications. The high quality of submissions to this year's workshop enabled us to accept a total of 20 technical papers from around the world.

The results of the spoken language translation evaluation campaigns organized in the framework of the workshop are also an important part of IWSLT. Those evaluations are not organized for the sake of competition, but their goal is to foster cooperative work and scientific exchange. While participants compete for achieving the best result in the evaluation, they come together afterwards and discuss and share their techniques that they used in their systems. In this respect, IWSLT proposes challenging research tasks and an open experimental infrastructure for the scientific community working on spoken and written language translation. The IWSLT 2012 Evaluation Campaign includes the following tasks:

- ASR track (**TED Task**): automatic transcription of talks from audio to text (in English)
- SLT track: speech translation of talks from audio (or ASR output) to text (from English to French)
- MT track: text translation of talks for two language pairs plus ten optional language pairs)
- HIT track (**Olympics Task**): text translation of the sentences taken from the Olympics domain (Chinese to English)

For each task, monolingual and bilingual language resources, as needed, are provided to participants in order to train their systems, as well as sets of manual and automatic speech transcripts (with n-best and lattices) and reference translations, allowing researchers working only on written language translation to also participate. Moreover, blind test sets are released and all translation outputs produced by the participants are evaluated using several automatic translation quality metrics. For the primary submissions of all MT and SLT tasks a human evaluation was carried out as well. Each participant in the evaluation campaign has been requested to submit a paper describing his system, the utilized resources.

A survey of the evaluation campaigns is presented by the organizers.

We would like to thank the IWSLT Steering Committee, Marcello Federico (FBK-irst, Italy) and Alex Waibel (CMU, USA / Karlsruhe Institute of Technology (KIT), Germany), with the former member, Satoshi Nakamura (NAIST, Japan). We would also like to thank the co-chairs of the Evaluation Committee, Marcello Federico, Tiejun Zhao (Harbin Institute of Technology, China), and Michael Paul (NICT, Japan), the co-chairs of the Program Committee, Chengqing Zong (National Laboratory of Pattern Recognition,

Chinese Academy of Sciences, China) and Chiori Hori (National Institute of Information and Communications Technology, Japan) and the local organizing committee members. Finally, we would like to warmly thank the all members of the Program Committee, who made a wonderful work in the selection of the technical papers, and the three keynote speakers (Dr. Dong Yu, Microsoft Research, USA, Prof. Hideki Isozaki, Okayama Prefectural University, Japan, Dr. Chai Wutiwiwatchai, National Electronics and Computer Technology Center (NECTEC), Thailand), who kindly accepted to give an invited talk at the conference.

Welcome to Hong Kong!

Dekai WU and Eiichiro SUMITA, Workshop Chairs IWSLT 2012

Organizers

Steering Committee

Eiichiro Sumita (NICT, Japan)

Marcello Federico (FBK-irst, Italy)

Alex Waibel (CMU, USA / KIT, Germany)

Workshop Chairs

Eiichiro Sumita (NICT, Japan)

Dekai Wu (HKUST, Hong Kong)

Evaluation Chairs

Marcello Federico (FBK, Italy)

Michael Paul (NICT, Japan)

Tiejun Zhao (HIT, China)

Technical Program Chairs

Chengqing Zong (CAS, China)

Chiori Hori (NICT, Japan)

Local Arrangement

Dekai Wu (HKUST, Hong Kong)

Program Committee

Alexandre Allauzen (LIMSI-CNRS, France)
Andrew Finch (NICT, Japan)
Arne Mauser (RWTH, Germany)
Boxing Chen (NRC-IIT, Canada)
Chien-Lin Huang (NICT, Japan)
Christian Saam (KIT, Germany)
Conghui Zhu (HIT, China)
Eunah Cho (KIT, Germany)
Florian Kraft (KIT, Germany)
Francisco Casacuberta (ITI-UPV, Spain)
Gopala Anumanchipalli (CMU, USA)
Graham Neubig (NAIST, Japan)
Hailon Cao (HIT, China)
Hajime Tsukada (NTT, Japan)
Haifeng Wang (Baidu, China)
Holger Schwenk (LIUM, France)
Hwee Tou Ng (NUS, Singapore)
Isabel Trancoso (INESC-ID, Portugal)
Jan Niehues (KIT, Germany)
Jiajun Zhang (CAS, China)
Joel Ilao (UPD, Philippines)
Kevin Duh (NAIST, Japan)
Laurent Besacier (LIG, France)
Le Sun (CAS, China)
Loic Barrault (LIUM, France)
Mauro Cettolo (FBK, Italy)
Min Zhang (I2R, Singapore)
Ming Zhou (Microsoft Research, China)
Mohammed Mediani (KIT, Germany)
Philippe Langlais (UdeM, Canada)
Patrik Lambert (University of Le Mans, France)
Qun Liu (ICT, China)
Sebastian Stueker (KIT, Germany)
Shigeki Matsuda (NICT, Japan)
Taro Watanabe (NICT, Japan)
Teresa Herrmann (KIT, Germany)
Wade Shen (MIT/LL, USA)
Xiaodong He (Microsoft Research, China)
Xiaodong Shi (Xiamen University, China)
Youzhen Wu (NICT, Japan)
Yves Lepage (Waseda University, Japan)

Program

Thursday, December 6th, 2012

08:30-09:15h	**Workshop Registration**
09:15-09:30h	Welcome remarks
09:30-10:15h	**Keynote Speech I**
	Dr. Chai Wutiwiwatchai - Toward Universal Network-based Speech Translation
10:15-10:35h	Coffee Break
10:35-12:00h	**Evaluation Campaign I**
10:35-11:20h	**Overview of the IWSLT 2012 Evaluation Campaign -** *Marcello Federico, Mauro Cettolo, Luisa Bentivogli, Michael Paul, Sebastian Stüker*
11:20-11:40h	**The NICT ASR System for IWSLT2012 -** *Hitoshi Yamamoto, Youzheng Wu, Chien-Lin Huang, Xugang Lu, Paul R. Dixon, Shigeki Matsuda, Chiori Hori, Hideki Kashioka*
11:40-12:00h	**The KIT Translation systems for IWSLT 2012 -** *Mohammed Mediani, Yuqi Zhang, Thanh-Le Ha, Jan Niehues, Eunah Cho, Teresa Herrmann, Rainer Kargel, Alexander Waibel*
12:00-14:00h	Lunch Break
14:00-16:00h	**Evaluation Campaign II**
14:00-14:20h	**The UEDIN Systems for the IWSLT 2012 Evaluation -** *Eva Hasler, Peter Bell, Arnab Ghoshal, Barry Haddow, Philipp Koehn, Fergus McInnes, Steve Renals, Pawel Swietojanski*
14:20-14:40h	**The NAIST MachineTranslation System for IWSLT2012 -** *Graham Neubig, Kevin Duh, Masaya Ogushi, Takatomo Kano, Tetsuo Kiso, Sakriani Sakti, Tomoki Toda, Satoshi Nakamura*
14:40-15:00h	**FBK's Machine Translation Systems for IWSLT 2012's TED Lectures -** *Nicholas Ruiz, Arianna Bisazza, Roldano Cattoni, Marcello Federico*
15:00-15:20h	**Coffee Break**
15:20-15:40h	**The RWTH Aachen Speech Recognition and Machine Translation System for IWSLT 2012 -** *Stephan Peitz, Saab Mansour, Markus Freitag, Minwei Feng, Matthias Huck, Joern Wuebker, Malte Nuhn, Markus Nußbaum-Thom, Hermann Ney*
15:40-16:00h	**The HIT- LTRC Machine Translation System for IWSLT 2012 -** *Xiaoning Zhu, Yiming Cui, Conghui Zhu, Tiejun Zhao, Hailong Cao*
16:00-17:30h	**Poster Session I**
	FBK @ IWSLT 2012 - ASR track - *Daniele Falavigna, Roberto Gretter, Fabio Brugnara, Diego Giuliani* **The 2012 KIT and KIT-NAIST English ASR Systems for the IWSLT Evaluation -** *Christian Saam, Christian Mohr, Kevin Kilgour, Michael Heck, Matthias Sperber, Keigo Kubo, Sebastian Stüker, Sakriani Sakti, Graham Neubig, Tomoki Toda, Satoshi Nakamura, Alex Waibel* **The KIT-NAIST (Contrastive) English ASR System for IWSLT 2012 -** *Michael Heck, Keigo Kubo, Matthias Sperber, Sakriani Sakti, Sebastian Stüker, Christian Saam, Kevin Kilgour, Christian Mohr, Graham Neubig, Tomoki Toda, Satoshi Nakamura, Alex Waibel* **EBMT System of Kyoto University in OLYMPICS Task at IWSLT 2012 -** *Chenhui Chu, Toshiaki Nakazawa, Sadao Kurohashi* **The LIG English to French Machine Translation System for IWSLT 2012 -** *Laurent Besacier, Benjamin Lecouteux, Marwen Azouzi, Quang Luong Ngoc* **The MIT-LL/AFRL IWSLT-2012 MT System -** *Jennifer Drexler, Wade Shen, Timothy Anderson, Brian Ore, Ray Slyh, Eric Hansen, Terry Gleason* **Minimum Bayes-Risk Decoding Extended with Two Methods: NAIST-NICT at IWSLT 2012 -** *Hiroaki Shimizu, Masao Utiyama, Eiichiro Sumita, Satoshi Nakamura* **The NICT Translation System for IWSLT 2012 -** *Andrew Finch, Ohnmar Htun, Eiichiro Sumita* **TED Polish-to-English translation system for the IWSLT 2012 -** Krzysztof Marasek **Forest-to-String Translation using Binarized Dependency Forest for IWSLT 2012 OLYMPICS Task -** *Hwidong Na, Jong-Hyeok Lee* **Romanian to English Automatic MT Experiments at IWSLT12 -** *Stefan Dumitrescu, Radu Ion, Dan Ştefănescu, Tiberiu Boroş, Dan Tufis* **The TUBITAK Statistical Machine Translation System for IWSLT 2012 -** *Coskun Mermer, Hamza Kaya, Ilknur Durgar El-Kahlout, Mehmet Ugur Dogan* **+ 7 poster presentations of the papers presented in the oral sessions**
18:00h	**Social Event**

Friday, December 7th, 2012

09:30-10:15h	**Keynote Speech II**
	Dr. Dong Yu - Who Can Understand Your Speech Better — Deep Neural Network or Gaussian Mixture Model?
10:15-10:40h	**Coffee Break**
10:40-12:00h	**Technical Papers I**
10:40-11:00h	**Active Error Detection and Resolution for Speech-to-Speech Translation -** *Rohit Prasad, Rohit Kumar, Sankaranarayanan Ananthakrishnan, Wei Chen, Sanjika Hewavitharana, Matthew Roy, Frederick Choi, Aaron Challenner, Enoch Kan, Arvind Neelakantan, Prem Natarajan*
11:00-11:20h	**A Method for Translation of Paralinguistic Information -** *Takatomo Kano, Sakriani Sakti, Shinnosuke Takamichi, Graham Neubig, Tomoki Toda, Satoshi Nakamura*
11:20-11:40h	**Continuous Space Language Models using Restricted Boltzmann Machines -** *Jan Niehues, Alex Waibel*
11:40-12:00h	**Focusing Language Models For Automatic Speech Recognition -** *Daniele Falavigna, Roberto Gretter*
12:00-13:30h	Lunch Break
13:30-14:15h	**Keynote Speech III**
	Prof. Hideki Isozaki - Head Finalization: Translation from SVO to SOV
14:15-15:35h	**Technical Papers II**
14:15-14:35h	**Simulating Human Judgment in Machine Translation Evaluation Campaigns -** *Philipp Koehn*
14:35-14:55h	**Semi-supervised Transliteration Mining from Parallel and Comparable Corpora -** *Walid Aransa, Holger Schwenk, Loic Barrault*
14:55-15:15h	**A Simple and Effective Weighted Phrase Extraction for Machine Translation Adaptation -** *Saab Mansour, Hermann Ney*
15:15-15:35h	**Applications of Data Selection via Cross-Entropy Difference for Real-World Statistical Machine Translation -** *Amittai Axelrod, QingJun Li, William D. Lewis*
15:35-16:00h	**Coffee Break**
16:00-17:30h	**Poster Session II**
	A Universal Approach to Translating Numerical and Time Expressions - *Mei Tum, Yu Zhou, Chengqing Zong* **Evaluation of Interactive User Corrections for Lecture Transcription -** *Henrich Kolkhorst, Kevin Kilgour, Sebastian Stuker, Alex Waibel* **Factored Recurrent Neural Network Language Model in TED Lecture Transcription -** *Youzheng Wu, Hitoshi Yamamoto, Xugang Lu, Shigeki Matsuda, Chiori Hori, Hideki Kashioka* **Incremental Adaptation Using Translation Information and Post-Editing Analysis -** *Frederic Blain, Holger Schwenk, Jean Senellart* **Interactive-Predictive Speech-Enabled Computer-Assisted Translation -** *Shahram Khadivi, Zeinab Vakil* **MDI Adaptation for the Lazy: Avoiding Normalization in LM Adaptation for Lecture Translation -** *Nick Ruiz, Marcello Federico* **Segmentation and Punctuation Prediction in Speech Language Translation Using a Monolingual Translation System -** *Eunah Cho, Jan Niehues and Alex Waibel* **Sequence Labeling-based Reordering Model for Phrase-based SMT -** *Minwei Feng, Jan-Thorsten Peter, Hermann Ney* **Sparse Lexicalised Features and Topic Adaptation for SMT -** *Eva Hasler, Barry Haddow, Philipp Koehn* **Spoken Language Translation Using Automatically Transcribed Text in Training -** *Stephan Peitz, Simon Wiesler, Markus Nußbaum-Thom, Hermann Ney* **Towards a Better Understanding of Statistical Post-Edition Usefulness -** *Marion Potet, Laurent Besacier, Herve Blanchon, Marwen Azouzi* **Towards Contextual Adaptation for Any-text Translation -** *Li Gong, Aurelien Max, Francois Yvon*
17:30-18:00h	**Closing Ceremony - Best Paper Awards**

Keynotes

Toward Universal Network-based Speech Translation

Dr. Chai Wutiwiwatchai,
National Electronics and Computer Technology Center (NECTEC), Thailand

Abstract: The speech translation technology has been widely expected to play an important role in today global communication. This talk will address activities of a recently developed international consortium, called Universal Speech Translation Advanced Research (U-STAR), which composes 26 research organizations from 23 Asian and European countries. This largest research consortium has jointly developed a network-based speech translation service which supports translation among 23 languages and accepts up to 17 languages speech input. The service has been developed based on shared language resources in travel and sport domains. Users are able to access the service via a freely available iPhone application, namely VoiceTra4U-M. This talk will start by describing the initiation of the U-STAR consortium, followed by summarizing the development issues on both language resource and system engineering parts. Some statistics and analyses of the global usage during a few months field-testing after service launching will be revealed. Finally, challenging issues to improve the service accuracy and to extend the number of supported languages and translation domains will be discussed.

Bio: Chai Wutiwiwatchai received his BEng (the first honor) and MEng degrees of electrical engineering from Thammasat and Chulalongkorn University, Thailand in 1994 and 1997 respectively. He received his PhD in Computer Science from Tokyo Institute of Technology in 2004 under the Japanese Governmental scholarship. He is now the Head of Speech and Audio Technology Laboratory, National Electronics and Computer Technology Center (NECTEC), Thailand. His research work includes several international collaborative projects in a wide area of speech and language processing including Universal Speech Translation Advanced Research (U-STAR), PAN Localization Network (PANL10N), and ASEAN Machine Translation. He is a member of International Speech Communication Association (ISCA), Institute of Electronics, Information and Communication Engineers (IEICE), and has served as a country representative in the ISCA international affair committee during 2007-2009.

Who Can Understand Your Speech Better — Deep Neural Network or Gaussian Mixture Model?

Dr. Dong Yu,
Microsoft Research

Abstract: Recently we have shown that the context-dependent deep neural network (DNN) hidden Markov model (CD-DNN-HMM) can do surprisingly well for large vocabulary speech recognition (LVSR) as demonstrated on several benchmark tasks. Since then, much work has been done to understand its potential and to further advance the state of the art. In this talk I will share some of these thoughts and introduce some of the recent progresses we have made.

In the talk, I will first briefly describe CD-DNN-HMM and bring some insights on why DNNs can do better than the shallow neural networks and Gaussian mixture models. My discussion will be based on the fact that DNN can be considered as a joint model of a complicated feature extractor and a log-linear model. I will then describe how some of the obstacles, such as training speed, decoding speed, sequence-level training, and adaptation, on adopting CD-DNN-HMMs can be removed thanks to recent advances. After that, I will show ways to further improve the DNN structures to achieve better recognition accuracy and to support new scenarios. I will conclude the talk by indicating that DNNs not only do better but also are simpler than GMMs.

Bio: Dr. Dong Yu joined Microsoft Corporation in 1998 and Microsoft Speech Research Group in 2002, where he is currently a senior researcher. He holds a PhD degree in computer science from University of Idaho, an MS degree in computer science from Indiana University at Bloomington, an MS degree in electrical engineering from Chinese Academy of Sciences, and a BS degree (with honors) in electrical engineering from Zhejiang University. His recent work focuses on deep neural network and its applications to large vocabulary speech recognition. Dr. Dong Yu has published over 100 papers in speech processing and machine learning and is the inventor/co-inventor of around 50 granted/pending patents. He is currently serving as an associate editor of *IEEE transactions on audio, speech, and language processing* (2011-) and has served as an associate editor of *IEEE signal processing magazine* (2008-2011) and the lead guest editor of *IEEE Transactions on Audio, Speech, and Language Processing* special issue on deep learning for speech and language processing (2010-2011).

Head Finalization: Translation from SVO to SOV

Prof. Hideki Isozaki,
Okayama Prefectural University

Abstract: Asian languages such as Japanese and Korean follow Subject-Object-Verb (SOV) word order, which is completely different from European languages such as English and French that follow Subject-Verb-Object word. The difference is not limited to the position of "Object" or the accusative case, and the former is also called head-final and the latter is also called head-initial. Because of the difference, phrase-based SMT between SVO and SOV does not work well. This talk introduces Head Finalization that reorders sentences into the head-final word order. According to the result of the NTCIR-9 workshop, Head Finalization was quite effective for English-to-Japanese patent translation.

Bio: Hideki Isozaki is a professor of Okayama Prefectural University, Japan. He received B.E., M.E., and Ph.D. from the University of Tokyo in 1983, 1986, and 1998 respectively. After joining Nippon Telegraph and Telephone Corporation (NTT) in 1986, he has worked on logical inference, information extraction, named entity recognition, question answering, summarization, and machine translation. From 1990 to 1991, he was a visiting scholar at Stanford University. He has authored or coauthored over 100 papers and Japanese books including *LaTeX with Complete Control* and *Question Answering Systems*.

Evaluation Campaign

Overview of the IWSLT 2012 Evaluation Campaign

M. Federico *M. Cettolo*
FBK
via Sommarive 18,
38123 Povo (Trento), Italy
{federico,cettolo}@fbk.eu

L. Bentivogli
CELCT
Via alla Cascata 56/c,
38123 Povo (Trento), Italy
bentivo@fbk.eu

M. Paul
NICT
Hikaridai 3-5,
619-0289 Kyoto, Japan
michael.paul@nict.go.jp

S. Stüker
KIT
Adenauerring 2,
76131 Karlsruhe, Germany
sebastian.stueker@kit.edu

Abstract

We report on the ninth evaluation campaign organized by the IWSLT workshop. This year, the evaluation offered multiple tracks on lecture translation based on the TED corpus, and one track on dialog translation from Chinese to English based on the Olympic trilingual corpus. In particular, the TED tracks included a speech transcription track in English, a speech translation track from English to French, and text translation tracks from English to French and from Arabic to English. In addition to the official tracks, ten unofficial MT tracks were offered that required translating TED talks into English from either Chinese, Dutch, German, Polish, Portuguese (Brazilian), Romanian, Russian, Slovak, Slovene, or Turkish. 16 teams participated in the evaluation and submitted a total of 48 primary runs. All runs were evaluated with objective metrics, while runs of the official translation tracks were also ranked by crowd-sourced judges. In particular, subjective ranking for the TED task was performed on a progress test which permitted direct comparison of the results from this year against the best results from the 2011 round of the evaluation campaign.

1. Introduction

The International Workshop on Spoken Language Translation (IWSLT) offers challenging research tasks and an open experimental infrastructure for the scientific community working on the automatic translation of spoken and written language. The focus of the 2012 IWSLT Evaluation Campaign was the translation of lectures and dialogs. The task of translating lectures was built around the TED[1] talks, a collection of public lectures covering a variety of topics. The TED Task offered three distinct tracks addressing automatic speech recognition (ASR) in English, spoken language translation (SLT) from English to French, and machine translation (MT) from English to French and from Arabic to English. In addition to the official MT language pairs, ten other unofficial translation directions were offered, with English as the target language and the source language being either Chinese, Dutch, German, Polish, Portuguese (Brazilian), Romanian, Russian, Slovak, Sloven, or Turkish.

[1] http://www.ted.com

This year, we also launched the so-called OLYMPICS Task, which addressed the MT of transcribed dialogs, in a limited domain, from Chinese to English.

For each track, a schedule and evaluation specifications, as well as language resources for system training, development and evaluation were made available through the IWSLT website. After the official evaluation deadline, automatic scores for all submitted runs we provided to the participants. In this edition, we received run submissions by 16 teams from 11 countries. For all the official SLT and MT tracks we also computed subjective rankings of all primary runs via crowd-sourcing. For the OLYMPICS Task, system ranking was based on a round-robin tournament structure, following the evaluation scheme adopted last year. For the TED task, as a novelty for this year, we introduced a double-elimination tournament, which previous experiments showed to provide rankings very similar to the more exhaustive but more costly round-robin scheme. Moreover, for the TED Task we run the subjective evaluation on a progress test—i.e., the evaluation set from 2011 that we never released to the participants. This permitted the measure of progress of SLT and MT against the best runs of the 2011 evaluation campaign.

In the rest of the paper, we introduce the TED and OLYMPICS tasks in more detail by describing for each track the evaluation specifications and the language resources supplied. For the TED MT track, we also provide details for the reference baseline systems that we developed for all available translation directions. Then, after listing the participants, we describe how the human evaluation was organized for the official SLT and MT tracks. Finally, we present the main findings of this year's campaign and give an outlook on the next edition of IWSLT. The paper concludes with two appendices, which present detailed results of the objective and subjective evaluations.

2. TED Task

2.1. Task Definition

The translation of TED talks was introduced for the first time at IWSLT 2010. TED is a nonprofit organization that "invites the world's most fascinating thinkers and doers [...] to give the talk of their lives". Its website makes the video

recordings of the best TED talks available under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 license[2]. All talks have English captions, which have also been translated into many other languages by volunteers worldwide.

This year we proposed three challenging tracks involving TED talks:

`ASR track`: automatic transcription of the talks' English audio;

`SLT track`: speech translation of talks from audio (or ASR output) to text, from English to French;

`MT track`: text translation of talks from:

`official`: English to French and Arabic to English

`unofficial`: German, Dutch, Polish, Portuguese-Brazil, Romanian, Russian, Slovak, Slovenian, Turkish and Chinese to English

In the following sections, we give an overview of the released language resources and provide more details about these three tracks.

2.2. Supplied Textual Data

Starting this year, TED data sets for the IWSLT evaluations are distributed through the WIT^3 web repository [1].[3] The aim of this repository is to make the collection of TED talks effectively usable by the NLP community. Besides offering ready-to-use parallel corpora, the WIT^3 repository also offers MT benchmarks and text-processing tools designed for the TED talks collection.

The language resources provided to the participants of IWSLT 2012 comprise monolingual and parallel training corpora of TED talks (`train`). Concerning the two official language pairs, the development and evaluation data sets (`dev2010` and `tst2010`), used in past editions, were provided for development and testing purposes. For evaluation purposes, two data sets were released: a new test set (`tst2012`) and the official test set of 2011 (`tst2011`) that was used as the `progress test` set to compare the results of this year against the best results achieved in 2011.

For the unofficial language pairs similar development/test set were prepared, most of them overlapping with the dev/test sets prepared for Arabic-English.

As usual, only the source part of the evaluation sets was released to the participants. All texts were UTF-8 encoded, case-sensitive, included punctuation marks, and were not tokenized. Parallel corpora were aligned at sentence level, even though the original subtitles were aligned at sub-sentence level. Details on the supplied monolingual and parallel data for the two official language pairs are given in Tables 1 and 2; the figures reported refer to tokenized texts.

[2]http://creativecommons.org/licenses/by-nc-nd/3.0/

[3]http://wit3.fbk.eu

Table 1: Monolingual resources for official language pairs

data set	lang	sent	token	voc
train	En	142k	2.82M	54.8k
	Fr	143k	3.01M	67.3k

Table 2: Bilingual resources for official language pairs

task	data set	lang	sent	token	voc	talks
MT$_{EnFr}$	train	En	141k	2.77M	54.3k	1029
		Fr		2.91M	66.9k	
	dev2010	En	934	20.1k	3.4k	8
		Fr		20.3k	3.9k	
	tst2010	En	1,664	32.0k	3.9k	11
		Fr		33.8k	4.8k	
	tst2011	En	818	14.5k	2.5k	8
		Fr		15.6k	3.0k	
	tst2012	En	1,124	21,5k	3.1k	11
		Fr		23,5k	3.7k	
MT$_{ArEn}$	train	Ar	138k	2.54M	89.7k	1015
		En		2.73M	53.9k	
	dev2010	Ar	934	18.3k	4.6k	8
		En		20.1k	3.4k	
	tst2010	Ar	1,664	29.3k	6.0k	11
		En		32.0k	3.9k	
	tst2011	Ar	1,450	25.6k	5.6k	16
		En		27.0k	3.7k	
	tst2012	Ar	1,704	27.8k	6.1k	15
		En		30.8k	4.1k	

Similar to last year, several out-of-domain parallel corpora, including texts from the United Nations, European Parliament, and news commentaries, were supplied to the participants. These corpora were kindly provided by the organizers of the *7th Workshop on Statistical Machine Translation*[4] and the *EuroMatrixPlus project* [5].

2.3. Speech Recognition

The goal of the *Automatic Speech Recognition* (ASR) track for IWSLT 2012 was to transcribe the English recordings of the `tst2011` and `tst2012` MT$_{EnFr}$ test sets (Table 2) for the TED task. This task reflects the recent increase of interest in automatic subtitling and audiovisual content indexing.

Speech in TED lectures is in general planned, well articulated, and recorded in high quality. The main challenges for ASR in these talks are to cope with a large variability of topics, the presence of non-native speakers, and the rather informal speaking style.

Table 3 provides statistics on the two sets; the counts of reference transcripts refer to lower-cased text without punctuation after the normalization described in detail in Section 2.6.

[4]http://www.statmt.org/wmt12/translation-task.html

[5]http://www.euromatrixplus.net/

Table 3: Statistics of ASR evaluation sets

task	data set	duration	sent	token	voc	talks
ASR_{En}	tst2011	1h07m28s	818	12.9k	2.3k	8
	tst2012	1h45m04s	1124	19.2k	2.8k	11

2.3.1. Language Resources

For acoustic model training, no specific data was provided by the evaluation campaign. Instead, just as last year, participants were allowed to use any data available to them, but recorded before December 31^{st}, 2010.

For language model training, the training data was restricted to the English monolingual texts and the English part of the provided parallel texts as described in Section 2.2.

2.4. Spoken Language Translation

The SLT track required participants to translate the English TED talks of `tst2011` and `tst2012` into French, starting from the audio signal (see Section 2.3). The challenge of this translation task over the MT track is the necessity to deal with automatic, and in general error prone, transcriptions of the audio signal, instead of correct human transcriptions.

Participants not using their own ASR system could resort to automatic transcriptions distributed by the organizers. These were the primary runs submitted by three participants to the ASR track:

Table 4: WER of ASR runs released for the SLT track

system num.	system name	tst2011	tst2012
1	NICT	10.9	12.1
2	MITLL	11.1	13.3
3	UEDIN	12.4	14.4

Table 4 shows their WERs. Participants could freely choose which set of transcriptions to translate; they were allowed even to create a new transcription, e.g., by means of system combination methods. Details on the specifications for this track are given in Section 2.6.

2.4.1. Language Resources

For the SLT task the language resources available to participants are the union of those of the ASR track, described in Section 2.3.1, and of the English-to-French MT track, described in Section 2.2.

2.5. Machine Translation

The MT TED track basically corresponds to a subtitling translation task. The natural translation unit considered by the human translators volunteering for TED is indeed the single caption—as defined by the original transcript—which in general does not correspond to a sentence, but to fragments of it that fit the caption space. While translators can look at the context of the single captions, arranging the MT task in this way would make it particularly difficult, especially when word re-ordering across consecutive captions occurs. For this reason, we preprocessed all the parallel texts to re-build the original sentences, thus simplifying the MT task.

Reference results from baseline MT systems on the official evaluation set (`tst2012`) are provided via the WIT^3 repository. This helps participants and MT scientists to assess their experimental outcomes, but also to set reference systems for the human evaluation experiments (Section 5).

MT baselines were trained from TED data only, i.e,. no additional out-of-domain resources were used. Preprocessing was applied as follows: Arabic and Chinese words were segmented by means of AMIRA [2] and the Stanford Chinese Segmenter [3], respectively; while for all the other languages the `tokenizer` script released with the Europarl corpus [4] was applied.

The baselines were developed with the Moses toolkit [5]. Translation and lexicalized reordering models were trained on the parallel training data; 5-gram LMs with improved Kneser-Ney smoothing [6] were estimated on the target side of the training parallel data with the IRSTLM toolkit [7]. The weights of the log-linear interpolation model were optimized on `dev2010` with the MERT procedure provided with Moses. Performance scores were computed with the `MultEval` script implemented by [8].

Table 5 collects the %BLEU, METEOR, and TER scores ("case sensitive+punctuation" mode) of all the baseline systems developed for all language pairs. In addition to the scores obtained on `dev2010` after the last iteration of the tuning algorithm, we also report the scores measured on the second development set (`tst2010`) and on the official test sets of the evaluation campaign (`tst2011`, `tst2012`). Note that the tokenizers and the scorer applied here are different from those used for official evaluation.

2.6. Evaluation Specifications

ASR—For the evaluation of ASR submissions, participants had to provide automatic transcripts of test talk recordings. The talks were accompanied by an UEM file that marked the portion of each talk that needed to be transcribed. Specifically excluded were the beginning portions of each talk containing a jingle and possibly introductory applause, and the applause and jingle at the end of each file after the speaker has concluded his talk. Also excluded were larger portions of the talks that did not contain the lecturer's speech.

In addition, the UEM file also provides a segmentation of each talk into sentence-like units. The segmentation was that at sentence-level used in the MT track (Section 2.2). While giving human-defined segmentation makes the transcription task easier than it would be in real life, the use of it facilitates the speech translation evaluation since the segmentation of the input language perfectly matches the segmentation of the reference translation used in evaluating the translation task.

Participants were required to provide the results of the au-

	%bleu	σ	mtr	σ	ter	σ
En-Fr						
dev2010	26.28	0.59	47.57	0.47	56.80	0.70
tst2010	28.74	0.47	49.63	0.37	51.30	0.47
tst2011	34.95	0.70	54.53	0.51	44.11	0.60
tst2012	34.89	0.61	54.68	0.44	43.35	0.50
Ar-En						
dev2010	24.70	0.54	48.66	0.39	55.41	0.59
tst2010	23.64	0.45	47.61	0.34	57.16	0.50
tst2011	22.66	0.49	46.37	0.37	60.27	0.59
tst2012	24.05	0.44	48.62	0.31	54.72	0.43
De-En						
dev2010	28.14	0.60	52.83	0.40	50.37	0.57
tst2010	26.18	0.48	50.86	0.34	52.59	0.50
tst2011	30.28	0.51	55.00	0.32	47.86	0.47
tst2012	26.55	0.48	50.99	0.32	52.42	0.46
Nl-En						
dev2010	23.79	0.62	47.04	0.49	57.14	0.64
tst2010	31.23	0.48	54.62	0.32	47.90	0.45
tst2011	33.45	0.55	56.31	0.36	45.11	0.49
tst2012	29.89	0.46	53.16	0.31	47.60	0.42
Pl-En						
dev2010	20.56	0.58	44.74	0.46	62.47	0.67
tst2010	15.27	0.36	40.03	0.31	69.95	0.47
tst2011	18.68	0.42	43.64	0.32	65.42	0.53
tst2012	15.89	0.39	39.11	0.32	68.56	0.48
Ptb-En						
dev2010	33.57	0.64	56.06	0.41	45.53	0.57
tst2010	35.27	0.47	58.85	0.31	43.01	0.43
tst2011	38.56	0.54	61.26	0.32	39.87	0.45
tst2012	40.74	0.50	62.09	0.29	37.96	0.40
Ro-En						
dev2010	29.30	0.57	53.26	0.40	49.54	0.56
tst2010	28.18	0.47	52.32	0.33	51.13	0.46
tst2011	32.46	0.52	55.92	0.34	45.99	0.48
tst2012	29.08	0.48	52.73	0.33	50.32	0.45
Ru-En						
dev2010	17.37	0.50	41.63	0.40	66.96	0.60
tst2010	16.82	0.37	41.93	0.29	66.28	0.47
tst2011	19.11	0.42	43.82	0.32	62.63	0.49
tst2012	17.44	0.39	41.73	0.31	63.94	0.43
Sk-En						
dev2012	19.23	0.42	42.65	0.32	62.03	0.46
tst2012	21.79	0.58	45.01	0.41	58.28	0.55
Sl-En						
dev2012	15.90	0.45	40.16	0.36	67.23	0.53
tst2012	14.33	0.39	39.42	0.33	69.20	0.50
Tr-En						
dev2010	11.13	0.40	36.29	0.37	78.25	0.54
tst2010	12.13	0.32	37.87	0.27	75.56	0.45
tst2011	13.23	0.37	39.21	0.30	74.00	0.49
tst2012	12.45	0.33	38.76	0.29	73.63	0.43
Zh-En						
dev2010	9.62	0.39	33.97	0.36	82.47	1.01
tst2010	11.39	0.32	36.80	0.28	75.99	0.76
tst2011	14.13	0.39	39.62	0.32	65.02	0.42
tst2012	12.33	0.33	37.67	0.30	67.80	0.39

Table 5: Performance of baselines in terms of %BLEU, METEOR (mtr) and TER scores, with standard deviations (σ). Values were computed in case-punctuation sensitive mode.

tomatic transcription in CTM format. Multiple submissions were allowed, but one submission had to be marked as the primary run.

The quality of the submissions was then scored in terms of word error rate (WER). The results were scored case-insensitive, but were allowed to be submitted case-sensitive. Numbers, dates, etc. had to be transcribed in words as they are spoken, not in digits. Common acronyms, such as NATO and EU, had to be written as one word, without any special markers between the letters. This applies no matter if they are spoken as one word or spelled out as a letter sequence. All other letter spelling sequences had to be written as individual letters with spaces in between. Standard abbreviations, such as "etc." and "Mr." were accepted as specified by the GLM file in the scoring package that was provided to participants for development purposes. For words pronounced in their contracted form, it was permitted to use the orthography for the contracted form, as these were normalized into their canonical form according to the GLM file.

SLT/MT—The participants to the SLT and MT tracks had to provide the results of the translation of the test sets in NIST XML format. The output had to be true-cased and had to contain punctuation. Participants to the SLT track could either use the audio files directly, or use automatic transcriptions selected from the ASR submissions (Table 4).

The quality of the translations was measured automatically with BLEU [9] by scoring against the human translations created by the TED open translation project, and by human subjective evaluation (Section 5).

The evaluation specifications for the SLT/MT tracks were defined as case-sensitive with punctuation marks (*case+punc*). Tokenization scripts were applied automatically to all run submissions prior to evaluation.

Moreover, automatic evaluation scores were also calculated for case-insensitive (lower-case only) translation outputs with punctuation marks removed (*no_case+no_punc*). Besides BLEU, six additional automatic standard metrics (METEOR [10], WER [11], PER [12], TER [13], GTM [14], and NIST [15]) were calculated offline.

3. OLYMPICS Task

As a continuation of previous spoken dialog translation tasks [16, 17], this year's IWSLT featured a translation task in the Olympics domain. The OLYMPICS task is a small-vocabulary task focusing on human dialogs in travel situations where the utterances were annotated with dialog and speaker information that could be exploited by the participant to incorporate contextual information into the translation process.

3.1. Task Definition

The translation input condition of the OLYMPICS task consisted of correct recognition results, i.e., text input. Participants of the OLYMPICS task had to translate the Chinese sentences into English.

The monolingual and bilingual language resources that could be used to train the translation engines for the primary

runs were limited to the supplied corpora described in Section 3.2. These include all supplied development sets, i.e., the participants were free to use these data sets as they wish for tuning model parameters or as training bitext, etc. All other language resources, such as any additional dictionaries, word lists, or bitext corpora were treated as "additional language resources".

3.2. Supplied Data

The OLYMPICS task was carried out using parts of the Olympic Trilingual Corpus (HIT), a multilingual corpus that covers 5 domains (traveling, dining, sports, traffic and business) closely related to the Beijing 2008 Olympic Games [18]. It includes dialogs, example sentences, articles from the Internet and language teaching materials.

Moreover, the Basic Travel Expression Corpus (BTEC) [19], a multilingual speech corpus containing tourism-related sentences, was provided as an additional training corpus. The BTEC corpus consists of 20k training sentences and the evaluation data of previous IWSLT evaluation campaigns [17].

Both corpora are aligned at sentence level. Table 6 summarizes the characteristics of the Chinese (*zh*) and English (*en*) training (*train*), development (*dev*) and evaluation (*eval*) data sets. The first two columns specify the given data set and its type. The source language text ("*text*") and target language reference translation ("*ref*") resources also include annotated sample dialogs ("*dialog*") and their translation into the respective language ("*lang*"). The number of sentences are given in the "*sent*" column, and the "*avg.len*" column shows the average number of characters/words per training sentence for Chinese/English, respectively. The reported figures refer to tokenized texts.

The BTEC development data sets include up to 16 English reference translations for 3k Chinese inputs sentences. For the HIT data sets, only single reference translations were available.

For each sentence of the HIT corpus, context information on the *type of text* (dialog, samples, explanation), *scene* (airplane, airport, restaurant, water/winter sports, etc.), *topic* (asking about traffic conditions, bargaining over a price, front desk customer service, etc.), and the *speaker* (customer, clerk, passenger, receptionist, travel agent, etc.) was provided to the participants.

The dialogs of the two development and the evaluation data sets were randomly extracted from the HIT corpus after disregarding dialogs containing too short (less than 5 words) or too long (more than 18 words) sentences. The evaluation and development data sets included a total of 123 and 157 dialogs consisting on average of 8 and 13 utterances, respectively.

The supplied resources were released to the participants three months ahead of the official run submission period. The official run submission period was limited to one week.

Table 6: Supplied Data (OLYMPICS)

BTEC	data	lang	sent	avg.len	token	voc
train	(text)	Zh	19,972	11.8	234,998	2,483
	(text)	En	19,972	9.1	182,627	8,344
dev	(text)	Zh	2,977	9.4	27,888	1,515
	(ref)	En	38,521	8.1	312,119	5,927

HIT	data	lang	sent	avg.len	token	voc
train	(text)	Zh	52,603	13.2	694,100	4,280
	(text)	En	52,603	9.5	515,882	18,964
dev1	(dialog)	Zh	1,050	12.8	13,416	1,296
	(ref)	En	1,050	9.6	10,125	1.992
dev2	(dialog)	Zh	1,007	13.3	13,394	1,281
	(ref)	En	1,007	10.0	10,083	1,900
eval	(dialog)	Zh	998	14.0	14,042	1,310
	(ref)	En	998	10.6	10,601	2,023

3.3. Run Submissions

Participant registered for the OLYMPICS translation task had to submit at least one run. Run submission was carried out via email to the organizers with multiple runs permitted. However, the participant had to specify which runs should be treated as *primary* (evaluation using human assessments and automatic metrics) or *contrastive* (automatic evaluation only). Re-submitting runs was allowed as far as they were submitted prior to the submission deadline.

In total, 4 research groups participated in the OLYMPICS task and 4 primary and 4 contrastive runs were submitted.

3.4. Evaluation Specifications

The evaluation specification for the OLYMPICS task was defined as case-sensitive with punctuation marks (*case+punc*). The same tokenization script was applied automatically to all run submissions and reference data sets prior to evaluation. In addition, automatic evaluation scores were also calculated for case-insensitive (lower-case only) MT outputs with punctuation marks removed (*no_case+no_punc*).

All primary and contrastive run submissions were evaluated using the standard automatic evaluation metrics described in Section 2.6 for both evaluation specifications (see Appendix A).

In addition, human assessments of the overall translation quality of a single MT system were carried out with respect to the *adequacy* of the translation with and without taking into account the context of the respective dialog. The differences in translation quality between MT systems were evaluated using a paired comparison method that adopts a round-robin tournament structure to determine a complete system ranking, as described in Section 5.

4. Participants

A list of the participants of this year's evaluation is shown in Table 7. In total, 14 research teams from 11 countries took part in the IWSLT 2012 evaluation campaign. The number of primary and contrastive run submissions for each tasks

Table 7: List of Participants

FBK	Fondazione Bruno Kessler, Italy [20, 21]
HIT	Harbin Institute of Technology, China [22]
KIT	Karlsruhe Institute of Technology, Germany [23]
KIT-NAIST	KIT& NAIST collaboration [24, 25]
KYOTO-U	Kyoto University, Kurohashi-Kawahara Lab, Japan [26]
LIG	Laboratory of Informatics of Grenoble, France [27]
MITLL	Mass. Institute of Technology/Air Force Research Lab., USA [28]
NAIST	Nara Institute of Science and Technology, Japan [29]
NAIST-NICT	NAIST& NICT collaboration [30]
NICT	National Institute of Communications Technology, Japan [31, 32]
PJIIT	Polish-Japanese Institute of Information Technology, Poland [33]
POSTECH	Pohang University of of Science and Technology, Korea [34]
RACAI	Research Institute for AI of the Romanian Academy, Romania [35]
RWTH	Rheinisch-Westfälische Technische Hochschule Aachen, Germany [36]
TUBITAK	TUBITAK - Center of Research for Advanced Technologies, Turkey [37]
UEDIN	University of Edinburgh, UK [38]

	TED													OLY
	ASR	SLT	MT											MT
	En	$EnFr$	$EnFr$	$ArEn$	$DeEn$	$NlEn$	$PlEn$	$PtbEn$	$RoEn$	$RuEn$	$SkEn$	$TrEn$	$ZhEn$	$ZhEn$
FBK	X		X	X	X	X					X	X		
HIT														X
KIT	X	X	X											
KYOTO-U														X
LIG			X											
MITLL	X	X	X	X										
NAIST	X		X	X	X	X	X	X	X	X	X	X	X	
NICT	X									X				X
PJIIT							X							
POSTECH														X
RACAI									X					
RWTH	X	X	X	X	X						X		X	
TUBITAK				X								X		
UEDIN	X	X	X		X									
	7	4	7	5	4	2	2	1	2	2	3	3	2	4

are summarized in Table 8. In total, 48 primary runs and 54 contrastive runs were submitted by the participants.

Table 8: Run Submissions

Task	Primary (Contrastive) [Systems]
TED ASR$_{En}$	7 (8) [FBK,KIT,KIT-NAIST,MITLL,NICT,RWTH,UEDIN]
TED SLT$_{EnFr}$	4 (8) [KIT,MITLL,RWTH,UEDIN]
TED MT$_{EnFr}$	7 (13) [FBK,KIT,LIG,MITLL,NAIST,RWTH,UEDIN]
TED MT$_{ArEn}$	5 (5) [FBK,MITLL,NAIST,RWTH,TUBITAK]
TED MT$_{DeEn}$	4 (5) [FBK,NAIST,RWTH,UEDIN]
TED MT$_{NlEn}$	2 (2) [FBK,NAIST]
TED MT$_{PlEn}$	2 (2) [NAIST,PJIIT]
TED MT$_{PtbEn}$	1 (0) [NAIST]
TED MT$_{RoEn}$	2 (4) [NAIST,RACAI]
TED MT$_{RuEn}$	2 (1) [NAIST,NICT]
TED MT$_{SkEn}$	3 (0) [FBK,NAIST,RWTH]
TED MT$_{TrEn}$	3 (1) [FBK,NAIST,TUBITAK]
TED MT$_{ZhEn}$	2 (1) [NAIST,RWTH]
OLY MT$_{ZhEn}$	4 (4) [HIT,KYOTO-U,NAIST-NICT,POSTECH]

5. Human Evaluation

Subjective evaluation was carried out on all primary runs submitted by participants to the official tracks of the TED task, namely the SLT track (English-French) and the MT *official* track (English-French and Arabic-English) and to the OLYMPICS task (Chinese-English).

For each task, systems were evaluated using a subjective evaluation set composed of 400 sentences randomly taken from the test set used for automatic evaluation. Each evaluation set represents the various lengths of the sentences included in the corresponding test set, with the exception of sentences with less than 5 words, which were excluded from the subjective evaluation.

Two metrics were used for the IWSLT 2012 subjective evaluation, i.e. *System Ranking* evaluation and, only for the OLYMPICS task, *Adequacy* evaluation.

The goal of the *Ranking* evaluation is to produce a complete ordering of the systems participating in a given task [39]. In the ranking task, human judges are given two MT outputs of the same input sentence as well as a reference translation and they have to decide which of the two translation hypotheses is better, taking into account both the content and fluency of the translation. Judges are also given the possibility to assign a tie in case both translations are equally good or bad. The judgments collected through these pairwise

comparisons are then used to produce the final ranking.

Following the practice consolidated in the previous campaign, the ranking evaluation in IWSLT 2012 was carried out by relying on crowd-sourced data. All the pairwise comparisons to be evaluated were posted to Amazon's Mechanical Turk[6] (MTurk) through the CrowdFlower[7] interface. Data control mechanisms including *locale qualifications* and *gold units* (items with known labels which enable distinguishing between trusted and untrusted contributors) implemented in CrowdFlower were applied to ensure the quality of the collected data [40].

For each pairwise comparison we requested three redundant judgments from different MTurk contributors. This means that for each task we collected three times the number of the necessary judgments. Redundant judgment collection is a typical method to ensure the quality of crowd-sourced data. In fact, instead of relying on a single judgment, label aggregation is computed by applying majority voting. Moreover, agreement information can be collected to find and manage the most controversial annotations.

In our ranking task, there are three possible assessments: *(i)* output A is better than output B, *(ii)* output A is worse than output B, or *(iii)* both output A and B are equally good or bad (tie). Having three judgements from different contributors and three possible values, it was not possible to assign a majority vote for a number of comparisons. These *undecidable comparisons* were interpreted as a tie between the systems (neither of them won) and were used in the evaluation.

In order to measure the significance of result differences for each pairwise comparison, we applied the Approximate Randomization Test[8]. The results for all the tasks are available in Appendix B.

Besides system ranking, an additional evaluation metrics was used in the OLYMPICS task, where the overall translation quality of a single run submission was also evaluated according to the translation *adequacy*, i.e., how much of the information from the source sentence was expressed in the translation with and without taking into account the context of the respective dialog. Details on the adequacy evaluation are given in Section 5.2.2.

Finally, in order to investigate the degree of consistency between human evaluators, we calculated inter-annotator agreement[9] using the *Fleiss' kappa coefficient* κ [42, 43]. This coefficient measures the agreement between multiple raters (three in our evaluation) each of whom classifies N items into C mutually exclusive categories, taking into account the agreement occurring by chance. It is calculated as:

$$\kappa = \frac{P(a)-P(e)}{1-P(e)}$$

where $P\ a$ is the observed pairwise agreement between the raters and $P\ e$ is the estimated agreement due to chance, calculated empirically on the basis of the cumulative distribution of judgments by all raters. If the raters are in complete agreement then κ = 1. If there is no agreement among the raters (other than what would be expected by chance) then $\kappa \leq 0$. The interpretation of the κ values according to [44] is given in Table 9.

Table 9: Interpretation of the κ coefficient.

κ	**Interpretation**
< 0	No agreement
0.0 – 0.20	Slight agreement
0.21 – 0.40	Fair agreement
0.41 – 0.60	Moderate agreement
0.61 – 0.80	Substantial agreement
0.81 – 1.00	Almost perfect agreement

Within this common evaluation framework, different procedures were applied to the TED and OLYMPICS tasks.

5.1. TED Task

For the TED Task, subjective ranking was performed on the *Progress Test*, i.e. on the 2011 evaluation set[10], with the goal of measuring the progress of SLT and MT with respect to the top-ranked 2011 systems.

As a major novelty for this year, a change in the type of tournament used for the ranking evaluation was introduced. In IWSLT 2011, we adopted a round robin tournament, which is the most accurate way to determine system ranking due to its completeness (each system competes against every other system). The drawback of round robin is that completeness comes at a high cost, due to the large number of comparisons to be carried out. Thus, our goal for this year's evaluation was to adopt a tournament structure comparable with round robin in terms of reliability, but requiring less comparisons in favor of cost effectiveness.

Existing studies about the efficacy of sport tournament structures [45] demonstrated that knockout tournaments are comparable to round robin, if double elimination procedures are used and the allocation of players to the tournament structure is accurately assigned a-priori according to some criterion (seeding). The most promising structure, given its ability of ranking all players and the relatively few comparisons required, is the *Double Seeded Knockout with Consolation* (DSKOC) tournament.

In the DSKOC scheme proposed in [45], each player must loose twice before elimination from the tournament. The loss of one game does not therefore preclude that player from winning that tournament, provided that all future contests are won. Consolation play-offs are allowed at each stage of the tournament in order to place all players, and the a-priori seeding protocol is: P_1 - P_8, P_5 - P_4, P_3 - P_6, P_7 - P_2.

[6] http://www.mturk.com

[7] http://www.crowdflower.com

[8] To calculate Approximate Randomization we used the package available at: http://www.nlpado.de/∼sebastian/software/sigf.shtml [41]

[9] Agreement scores are presented in Section 5.1, Section 5.2, and in Appendix B.

[10] The reference translations for the 2011 test set were never released to the participants.

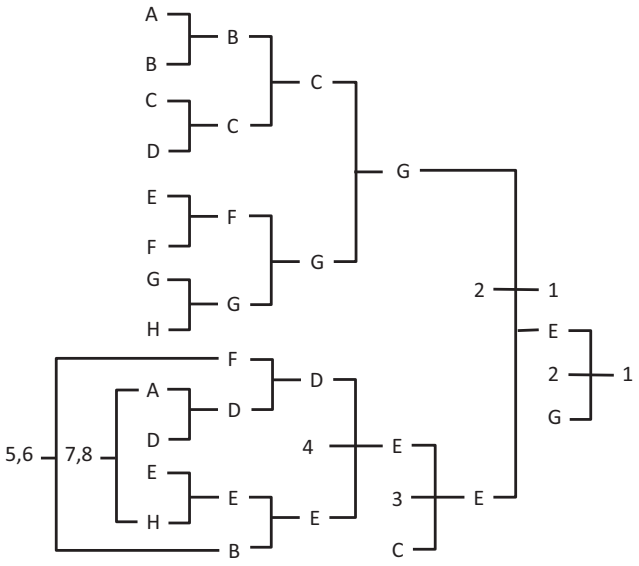


Figure 1: Example of double knockout tournament with consolation at each stage of the competition

Figure 1 shows an example of a DSKOC tournament structure.

The DSKOC scheme was tested on the IWSLT 2011 data. For all IWSLT 2011 tasks, the system ranking obtained with DSKOC was the same as the one obtained with the round robin scheme. Therefore, the DSKOC tournament was adopted with an 8-player scheme for the human evaluation of the IWSLT 2012 TED tasks. For the a-priori seeding, we used the *BLEU* scores obtained by the systems on the full 2011 test set.

Our evaluation scenario raises two issues that differentiate it from real sport tournaments, namely:

1. *Tied matches*. In case of tied outcome - i.e. equal number of evaluation sentences for which one system was judged better than the other - majority voting was not applied. Instead, we took into account all the judgments of each match and calculated which system obtained the highest number of "win" judgments[11].

2. *Systems competing more than once against each other*. The idea of giving a chance of recovery from an aberrant result, which is at the basis of the double elimination scheme in real sport tournaments, is not viable in our scenario were crowd-sourced judgments are collected only once. Thus, if two systems have to be compared twice, a second evaluation is not run and the results of the first comparison are used.

Our aim for IWSLT 2012 was not only to evaluate all the primary runs submitted for IWSLT 2012, but also to assess their progress with respect to the best 2011 systems. Given that an 8-player tournament was adopted, different system selection criteria were applied, depending on the participants in each track.

- **SLT$_{\mathbf{EnFr}}$**: all four 2012 runs were evaluated together with the four best runs of 2011[12].

- **MT$_{\mathbf{ArEn}}$**: all five 2012 runs were evaluated together with a baseline created by the organizers and the top two runs of 2011. In this track, only the top two 2011 systems were selected. This is due to the fact that the evaluation of last year showed a large gap between the two top-ranked systems and the last two systems, which obtained poor results both in terms of automatic metrics and subjective ranking.

- **MT$_{\mathbf{EnFr}}$**: as eight primary runs were submitted this year, two subsequent tournaments were carried out. In the first tournament, only the bottom four runs of 2012 were ranked. The top four runs of 2012 were ranked jointly with the top four 2011 runs in the second tournament.

A summary of the TED *Ranking* task is given in Table 10. Concerning the number of different matches, not all the comparisons required in a standard scenario[13] were crowdsourced in this evaluation, because *(i)* already evaluated matches were not repeated, and *(ii)* the results for the matches involving 2011 systems were taken from the IWSLT 2011 evaluation data. As far as inter-annotator agreement is concerned, all the three tracks are in the range of "Fair agreement". These results are comparable with those obtained last year and confirm the general trend among the tracks, where SLT$_{EnFr}$ shows the lowest agreement rate and MT$_{EnFr}$ the highest one.

Table 10: Summary of the TED *Ranking* task

Task	# ranked systems	# different HtH matches	I.A.A. (κ)
SLT$_{EnFr}$	8	15 (3 from 2011)	0.2263
MT$_{ArEn}$	8	14 (0 from 2011)	0.2496
MT$_{EnFr}$	12	21 (3 from 2011)	0.2861

For each TED track, Appendix B provides the system rankings according to the BLEU scores and the human evaluation results, as well as the complete structure of the tournaments and detailed tables about pairwise head-to-head comparisons.

5.2. OLYMPICS Task

The human evaluation of the new IWSLT 2012 translation task on the Olympics domain was performed with respect to system ranking and dialog adequacy. Both methodologies are described below.

[11] In other words, ties were resolved considering all the 1,200 judgments collected for the 400 evaluation sentences, instead of using the 400 labels resulting from majority voting.

[12] All the best 2011 systems where chosen according to the results of last year's human evaluation.

[13] In an 8-player DSKOC structure, 16 or 17 matches are necessary to complete the tournament.

5.2.1. System ranking

Following last year's evaluation methodology, system ranking for the OLYMPICS task was achieved through a paired comparison method that adopts a round-robin tournament structure. Round-robin is the most complete way to determine system ranking as it ensures a full coverage of paired comparisons between systems. We first prepared all the paired comparisons necessary for a complete round-robin over the selected 400 evaluation sentences (m=#sentences). Each system was evaluated against each of the other systems for each evaluation sentence. Considering all systems (n=#systems), there are $n\ n-\ /$ pairwise comparisons for each evaluation sentence, and thus $m*n\ n-\ /$ comparisons for the whole evaluation set. The complete ranking of the four system submissions to the task (n=4) using 400 evaluation sentences (m=400) required 2,400 comparisons.

Table 11: *Paired Comparison* Evaluation.

# systems	# comparisons per system	# comparison in total	# collected judgments	I.A.A. κ
4	1,200	2,400	7,200	0.3653

A summary of the OLY_{ChEn} paired comparison task is given in Table 11. As far as inter-annotator agreement is concerned, the results obtained compare well with the overall results obtained last year, falling in the class of "Fair agreement". The complete ranking of the systems and the results of all the pairwise comparisons are given in Appendix B.4.

5.2.2. Dialog Adequacy

In addition to the system ranking based on paired comparison, human assessments of the overall translation quality of a single MT system were carried out with respect to the *Adequacy* of the translation for all OLYMPICS task run submissions. For *Adequacy*, the evaluator was presented with the source language input as well as a reference translation and had to judge how much of the information from the original sentence was expressed in the translation [46]. The *Adequacy* judgments consisted of one of the grades listed in Table 12.

Table 12: Dialog Adequacy

Adequacy / Dialog	
5	All Information
4	Most Information
3	Much Information
2	Little Information
1	None

In addition to the above standard metrics, a modified version of the *adequacy* metrics (*dialog*) that takes into account information beyond the current input sentence was applied to the translation results of the OLYMPICS task in order to judge a given MT output in the context of the respective dialog. For the *dialog* assessment, the evaluators were presented with the history of previously uttered sentences, the input sentence and the reference translation. The evaluator had to read the dialog history first and then had to judge how much of the information from the reference translation is expressed in the translation in the context of the given dialog history by assigning one of the *dialog* grades listed in Table 12. In cases where parts of the information were omitted in the system output, but they could be understood in the context of the given dialog, such omission would not result in a lower *dialog* score. For the final adequacy metric scores, each system score was calculated as the *median* of the assigned grades. The adequacy evaluation was carried out by an expert grader trained on the given tasks.

The *adequacy* evaluation results of all run submissions are summarized in Appendix B.4. The *dialog* assessment was carried out one week after the *adequacy* evaluation was finished. In order to reduce evaluation costs, only the best performing system (HIT) according to the *adequacy* metric was selected for the subjective evaluation using the *dialog* metric. We measured the *intra-grader* consistency[14] and obtained a κ coefficient of 0.51 (*moderate* agreement) and 0.74 (*substantial* agreement) for the *adequacy* and *dialog* assessment, respectively.

6. Main Findings

In this section, we point out the methods and solutions that, according to the participants' descriptions, contributed most to the performance of their systems. Our intent is to provide some useful suggestions for setting up strong baselines for each track for the benefit of future participants or any interested researcher. The complete list of the system description papers that we consulted is included in the references and can be found in Table 7.

6.1. TED Task

In the following, we briefly comment on the general outcomes of each track and point out relevant features of the systems that participated this year. Notice that our selection cannot be considered exhaustive nor objective.

6.1.1. ASR Track

Seven teams participated this year in the ASR track. A comparison of the 2011 and 2012 results on the progress test set is given in Appendix A.2. We indeed observe a significant drop in WER[15] between the two best performing systems, from 13.5% to 10.9%. Remarkable progress is observed for all teams that participated in both editions.

All the ASR system developed this year have complex architectures performing multiple adaptation and system combination steps. Some of their relevant aspects are briefly highlighted:

[14] The proportion of times that the same judge assigns the same grade when assessing the same system output twice.

[15] Notice that these figures differ from those reported in [47] as the references were afterwards manually improved.

Acoustic training data: The NICT system was trained only on TED recordings, roughly 170h of speech, which means much less data was used than for other systems.
Acoustic front-end: The best performing systems employed multiple acoustic front-ends, including MLPs (KIT, RWTH) and deep NN features (UEDIN), to lower feature dimensionality.
Acoustic models: The top performing systems employed AMs trained on different acoustic features and with different methods, combining SAT and discriminative methods.
Language models: The NICT and MITLL engines include a RNN LM for n-best re-scoring. All participants used n-gram LMs adapted via data selection and interpolation, both before and after decoding. FBK reports comparable results when adaptation is done after decoding.

6.1.2. SLT Track

Four teams participated in this track. Subjective rankings were carried out on the progress test by considering all 2012 primary SLT runs and the four best SLT runs of 2011. Detailed automatic scores and subjective ranking results are reported in Appendices A and B, respectively. The reported BLEU rankings on the current and progress tests result are consistent and statistically significant. According to the subjective ranking, the top three 2012 systems are better than the top 2011 run. Notice, however, that the subjective ranking of the 2012 runs differs from the corresponding BLEU ranking.

Participants in the SLT track used their own ASR system output which was post-processed in order to match the standard MT input conditions (punctuation and casing). MT was performed on the single best ASR output using the same engine as the French-English MT track, or after minor changes.

6.1.3. MT Track

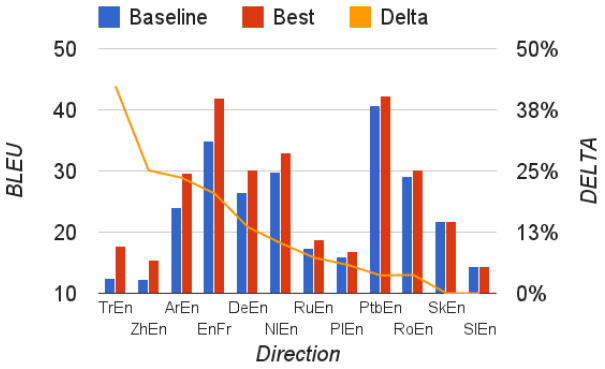
The official English-French and Arabic-English tracks had 7 participants, 5 respectively. For English-French, the BLEU rankings on the current and progress tests differ slightly. Subjective ranking on the progress test was carried out with two subsequent tournaments: one to select the top four runs of 2012, and another to determined their ranking jointly with the top four systems in the 2011 runs. The final outcome tells that the best two 2012 runs improved over the best 2011 run, and that the top three 2012 runs had identical BLEU ranks.

For Arabic-English, BLEU rankings on the current and progress tests are also slightly different. Subjective ranking was performed on the progress test by also including the best two 2011 runs. The best 2012 run ranks above the best 2011 run. The best two 2012 teams also improved their own 2011 runs. The subjective and BLEU rankings are again in perfect agreement.

A comparison between the baseline and the best performing systems is given in Figure 2.

The work carried out by the participants of TED MT tasks focused on the following aspects:

Figure 2: TED MT track: best runs vs. baselines



Data selection and adaptation: Basically all participants exploited data selection on the available out-of-domain resources to reduce the size and improve the accuracy of translation and language models. Out-domain models are combined by linear interpolation, log-linear interpolation, union (KIT,UEDIN), or fill-up (FBK). UEDIN also performed translation model adaptation with sparse lexical features.
Language model: Some well performing MT systems used class-based (KIT,RWTH) or hybrid (FBK) LMs to model the style of talks. KIT reports slight improvements with a continuous space LM (Restricted Bolzmannn Machine) applied during decoding.
Translation model: RWTH employed an improved phrase-extraction method that drastically reduces the size of the phrase-table. RWTH also reports gains with HPBT on Chinese-English, and FBK on Turkish-English. On the other side, UEDIN reports better results with PBT on German-English and French-English.
Reordering: For distant translation directions, RWTH and KIT applied POS based re-ordering rules, while FBK applied a hierarchical orientation model and early distortion cost estimates.
System combination: RWTH reports significant gains through confusion-network-based system combination.
To conclude, a few remarks concerning language specific issues. **Arabic and Chinese**: RWTH reports improvements by combining MT systems using multiple word segmentation models. For Chinese, RWTH also employs MT decoders processing text in reverted word order. **Turkish**: FBK reports relevant gains by using morphological segmentation and HPBT models. **Polish**: PJIIT reported negative results by applying morpho-syntactic factored models.

6.2. OLYMPICS Task

Four teams participated in the OLYMPICS task using quite different MT architectures including phrase-based SMT (HIT, NICT), syntax-based SMT (POSTECH), and syntax-based EBMT (KYOTO-U) approaches. The difficulty of this year's dialog translation tasks lay in the handling of out-of-vocabulary words and the sentence structure differences

(non-parallel sentence) of the supplied language resources, leading to lower evaluation scores for the structured-based MT approaches.

The work carried out by the participants of the OLYMPICS task focused on the following aspects:

Data preprocessing: The pre-processing of the Chinese language resources was carried out using the Stanford word segmenter [3] with the PKU model (HIT, NAIST-NICT) and in-house segmenters (KYOTO-U, POSTECH). For English, all participants only applied simple tokenization scripts. In addition, KYOTO-U applied sub-sentence splitting and non-parallel sentences filtering to improve the bilingual sentence alignment quality of the supplied corpus.

Additional language resources: KYOTO-U investigated the effects of using of external resources such as Wikipedia in order to reduce the out-of-vocabulary problem. Unfortunately, none of the participants used the dialog and speaker information annotated in the supplied corpus.

Translation model: HIT focused on model combination of phrase tables generated by *GIZA++* and *Pialign*.

Decoding: NICT extended the Minimum Bayes Risk decoding approach by considering maximum a-posteriori translation similarities and by taking advantage of the nearest neighbors of the source sentence. POSTECH focused on a forest-to-string machine translation approach based on binarized dependency forests. KYOTO-U carried out a tree-based decoding approach that uses an example-based MT (EBMT) system and integrates a Bayesian subtree alignment model based on dependency trees.

Clear and consistent rankings were obtained for human assessment using both paired comparison and adequacy metrics. Differences between all systems were statistically significant. Moreover, a comparison of the *adequacy* and *dialog* score differences of this year's and previous dialog translation tasks [16, 17] indicate that *dialog* metrics more closely reflect the reluctance of humans to accept machine translated output when taking into account the context of the conversation across different dialog types and domains.

7. Conclusions

We presented the organization and outcomes of the 2012 IWSLT Evaluation Campaign. This year the evaluation introduced several novelties: a small vocabulary translation tasks (OLYMPICS), unofficial TED talk MT tasks from 10 different languages into English, the use of a progress test set to compare this year's systems with the best runs of last year, and finally the adoption of new a tournament scheme to run the subjective evaluation on the official tracks. 16 teams participated in the evaluation, submitting a total of 48 primary runs. According to the automatic and subjective rankings of the official tracks on the progress test, performance was improved over the best results of last year. For the unofficial track, results by Moses baseline systems were made available for all 10 language pairs. For most of the tasks, participants were able to perform significantly better than the baseline. The plan for 2013 is to include additional unofficial language pairs and to adopt as progress test the 2012 test set, which for this reason will not be publicly released.

8. Acknowledgements

Research Group 3-01' received financial support by the '*Concept for the Future*' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative. The work leading to these results has received funding from the European Union under grant agreement no 287658 — Bridges Across the Language Divide (EU-BRIDGE). The subjective evaluation of the Evaluation Campaign was financed by a grant of the European Association for Machine Translation.

9. References

[1] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of the Annual Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012. [Online]. Available: http://hltshare.fbk.eu/EAMT2012/html/Papers/59.pdf

[2] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks," in *HLT-NAACL 2004: Short Papers*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 149–152.

[3] H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning, "A conditional random field word segmenter," in *Fourth SIGHAN Workshop on Chinese Language Processing*, 2005.

[4] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, Phuket, Thailand, September 2005, pp. 79–86.

[5] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic, 2007, pp. 177–180. [Online]. Available: http://aclweb.org/anthology-new/P/P07/P07-2045.pdf

[6] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech and Language*, vol. 4, no. 13, pp. 359–393, 1999.

[7] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models," in *Proceedings of Interspeech*, Melbourne, Australia, 2008, pp. 1618–1621.

[8] J. Clark, C. Dyer, A. Lavie, and N. Smith, "Better hypothesis testing for statistical machine translation: Controlling for optimizer instability," in *Proceedings of the Association for Computational Lingustics*, ser. ACL 2011. Portland, Oregon, USA: Association for Computational Linguistics, 2011, available at http://www.cs.cmu.edu/ jhclark/pubs/significance.pdf.

[9] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, USA, 2002, pp. 311–318.

[10] A. Lavie and A. Agarwal, "METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments," in *Proceedings of the Second Workshop on Statistical Machine Translation (WMT)*, Prague, Czech Republic, 2007, pp. 228–231.

[11] S. Niessen, F. J. Och, G. Leusch, and H. Ney, "An Evaluation Tool for Machine Translation: Fast Evaluation for Machine Translation Research," in *Proceedings of the Second International Conference on Language Resources & Evaluation (LREC)*, Athens, Greece, 2000, pp. 39–45.

[12] F. J. Och, "Minimum Error Rate Training in SMT," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, 2003, pp. 160–167.

[13] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of the The Seventh Conference of the Association for Machine Translation in the Americas (AMTA)*, Cambridge, USA, 2006, pp. 223–231.

[14] J. P. Turian, L. Shen, and I. D. Melamed, "Evaluation of Machine Translation and its Evaluation," in *Proceedings of the MT Summmit IX*, New Orleans, USA, 2003, pp. 386–393.

[15] G. Doddington, "Automatic Evaluation of Machine Translation Quality using N-gram Co-Occurrence Statistics," in *Proceedings of the Second International Conference on Human Language Technology (HLT)*, San Diego, USA, 2002, pp. 257–258.

[16] M. Paul, "Overview of the IWSLT 2009 Evaluation Campaign," in *Proceedings of the sixth International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, 2010, pp. 1–18.

[17] M. Paul, M. Federico, and S. Stüker, "Overview of the IWSLT 2010 Evaluation Campaign," in *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, Paris, France, 2010, pp. 3–27.

[18] M. Yang, H. Jiang, T. Zhao, and S. Li, "Construct Trilingual Parallel Corpus on Demand," *Chinese Spoken Language Processing, Lecture Notes in Computer Science*, vol. 4274, pp. 760–767, 2006.

[19] G. Kikui, S. Yamamoto, T. Takezawa, and E. Sumita, "Comparative study on corpora for speech translation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14(5), pp. 1674–1682, 2006.

[20] D. Falavigna, R. Gretter, F. Brugnara, and D. Giuliani, "FBK @ IWSLT 2012 - ASR track," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[21] N. Ruiz, A. Bisazza, R. Cattoni, and M. Federico, "FBKs Machine Translation Systems for IWSLT 2012s TED Lectures," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[22] X. Zhu, Y. Cui, C. Zhu, T. Zhao, and H. Cao, "The HIT-LTRC Machine Translation System for IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[23] M. Mediani, Y. Zhang, T.-L. Ha, J. Niehues, E. Cho, T. Herrmann, R. Kärgely, and A. Waibel, "The KIT Translation systems for IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[24] C. Saam, C. Mohr, K. Kilgour, M. Heck, M. Sperber, K. Kubo, S. Stüker, S. Sakti, G. Neubig, T. Toda, S. Nakamura, and A. Waibel, "The 2012 KIT and KIT-NAIST English ASR Systems for the IWSLT Evaluation," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[25] M. Heck, K. Kubo, M. Sperber, S. Sakti, S. Stüker, C. Saam, K. Kilgour, C. Mohr, G. Neubig, T. Toda, S. Nakamura, and A. Waibel, "The KIT-NAIST (Contrastive) English ASR System for IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[26] C. Chu, T. Nakazawa, and S. Kurohashi, "EBMT System of Kyoto University in OLYMPICS Task at IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[27] L. Besancier, B. Lecouteux, M. Azouzi, and L. N. Quang, "The LIG English to French Machine Translation System for IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[28] J. Drexler, W. Shen, T. Gleason, T. Anderson, R. Slyh, B. Ore, and E. Hansen, "The MIT-LL/AFRL IWSLT-2012 MT System," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[29] G. Neubig, K. Duh, M. Ogushi, T. Kano, T. Kiso, S. Sakti, T. Toda, and S. Nakamura, "The NAIST Machine Translation System for IWSLT2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[30] H. Shimizu, M. Utiyama, E. Sumita, and S. Nakamura, "Minimum Bayes-Risk Decoding Extended with Two Methods: NAIST-NICT at IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[31] H. Yamamoto, Y. Wu, C.-L. Huang, X. Lu, P. Dixon, S. Matsuda, C. Hori, and H. Kashioka, "The NICT ASR System for IWSLT2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[32] A. Finch, O. Htun, and E. Sumita, "The NICT Translation System for IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[33] K. Marasek, "TED English-to-Polish translation system for the IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[34] H. Na and J.-H. Lee, "Forest-to-String Translation using Binarized Dependency Forest for IWSLT 2012 OLYMPICS Task," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[35] S. D. Dumitrescu, R. Ion, D. Stefanescu, T. Boros, and D. Tufis, "Romanian to English Automatic MT Experiments at IWSLT12," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[36] S. Peitz, S. Mansour, M. Freitag, M. Feng, M. Huck, J. Wuebker, M. Nuhn, M. Nußbaum-Thom, and H. Ney, "The RWTH Aachen Speech Recognition and Machine Translation System for IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[37] C. Mermer, "The TUBITAK Statistical Machine Translation System for IWSLT 2012," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[38] E. Hasler, P. Bell, A. Ghoshal, B. Haddow, P. Koehn, F. McInnes, S. Renals, and P. Swietojanski, "The UEDIN Systems for the IWSLT 2012 Evaluation," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[39] M. Federico, S. Stüker, L. Bentivogli, M. Paul, M. Cettolo, T. Herrmann, J. Niehues, and G. Moretti, "The IWSLT 2011 Evaluation Campaign on Automatic Talk Translation," in *Proceedings of the eighth International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey, 2012, pp. 3543–3550.

[40] L. Bentivogli, M. Federico, G. Moretti, and M. Paul, "Getting Expert Quality from the Crowd for Machine Translation Evaluation," in *Proceedings of the MT Summmit XIII*, Xiamen, China, 2011, pp. 521–528.

[41] S. Padó, *User's guide to `sigf`: Significance testing by approximate randomisation*, 2006.

[42] S. Siegel and N. J. Castellan, *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, 1988.

[43] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, vol. 76(5), 1971.

[44] J. Landis and G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33 (1), pp. 159–174, 1977.

[45] T. McGarry and R. Schutz, "Efficacy of traditional sport tournament structures," *The Journal of the Operational Research Society*, vol. 48(1), pp. 65–74, 1997.

[46] J. S. White, T. O'Connell, and F. O'Mara, "The ARPA MT evaluation methodologies: evolution, lessons, and future approaches," in *Proc of the AMTA*, 1994, pp. 193–205.

[47] M. Federico, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2011 Evaluation Campaign," in *Proceedings of the eighth International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, USA, 2011, pp. 11–27.

[48] Y. Zhang, S. Vogel, and A. Waibel, "Interpreting Bleu/NIST Scores: How Much Improvement do We Need to Have a Better System?" in *Proceedings of the Second International Conference on Language Resources & Evaluation (LREC)*, 2004, pp. 2051–2054.

Appendix A. Automatic Evaluation

"case+punc" evaluation : case-sensitive, with punctuations tokenized
"no_case+no_punc" evaluation : case-insensitive, with punctuations removed

A.1. Official Testset (*tst2012*)

· All the sentence IDs in the IWSLT 2012 testset were used to calculate the automatic scores for each run submission.
· ASR and MT systems are ordered according to the *WER* and *BLEU* metrics, respectively.
· For each task, the best score of each metric is marked with **boldface**.
· Besides the NIST metrics, all automatic evaluation metric scores are given as percent figures (%).
· Besides the ASR scores, the mean scores of 2000 iterations were calculated for each MT output according to the *bootStrap* method [48].
· Omitted lines between scores indicate non-significant differences in performance between the MT engines.

TED : ASR English (ASR$_{En}$)

System	WER	(Count)
NICT	**12.1**	**(2318)**
KIT-NAIST	12.4	(2392)
KIT	12.7	(2435)
MITLL	13.3	(2565)
RWTH	13.6	(2621)
UEDIN	14.4	(2775)
FBK	16.8	(3227)

TED : SLT English-French (SLT$_{EnFr}$)

"case+punc" evaluation							System	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
29.78	**59.35**	**53.56**	**44.94**	**50.89**	**60.17**	**6.730**	KIT	**31.09**	**58.35**	**53.40**	**45.15**	**51.86**	**59.73**	**7.031**
29.09	58.83	54.38	45.29	51.83	59.67	6.646	UEDIN	30.70	58.08	53.96	45.38	52.59	59.39	6.946
28.51	57.50	54.93	46.11	52.56	59.18	6.611	RWTH	29.96	56.95	54.37	46.13	53.07	58.90	6.901
24.67	55.59	61.05	50.93	58.44	55.86	5.908	MITLL	25.52	54.58	61.59	51.75	60.16	55.12	6.100

TED : MT English-French (MT$_{EnFr}$)

"case+punc" evaluation							System	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
40.65	**69.21**	42.02	34.91	39.96	68.95	7.969	UEDIN	39.22	**66.32**	44.73	37.09	43.32	67.02	8.031
40.44	68.74	**40.82**	**34.62**	**38.82**	**69.32**	**8.102**	KIT	**39.23**	65.94	**43.33**	**36.78**	**42.01**	**67.44**	**8.187**
39.45	68.01	42.49	35.82	40.60	68.30	7.916	NAIST	38.06	65.16	45.35	38.15	44.13	66.29	7.967
39.40	68.37	41.61	35.23	39.53	69.03	8.034	RWTH	38.16	65.46	44.22	37.57	42.98	67.04	8.099
37.58	67.23	43.00	35.96	41.00	68.04	7.856	LIG	36.04	64.27	45.72	38.31	44.44	65.98	7.892
37.27	66.76	44.15	36.91	42.27	67.16	7.712	FBK	35.73	63.78	47.05	39.40	45.77	64.93	7.740
32.93	64.34	50.09	41.49	47.77	64.02	6.980	MITLL	31.57	61.24	53.49	44.32	51.99	61.68	6.989

TED : MT Arabic-English (MT$_{ArEn}$)

"case+punc" evaluation							System	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
29.32	**65.71**	**50.86**	**41.79**	**48.18**	**63.23**	**7.046**	RWTH	**28.24**	**63.13**	**53.67**	**43.99**	**51.99**	**61.43**	**7.156**
27.87	63.85	54.45	44.57	51.63	61.03	6.656	FBK	26.40	61.03	57.94	47.28	55.98	58.79	6.686
25.33	61.14	56.57	46.70	54.01	59.06	6.356	NAIST	23.77	58.03	60.12	47.37	58.32	56.46	6.360
25.30	62.33	54.20	44.75	51.53	60.17	6.519	TUBITAK	23.90	59.38	57.53	48.64	55.77	57.89	6.568
19.32	61.59	61.29	51.85	53.61	53.37	5.390	MITLL	22.95	58.51	60.07	49.62	58.16	57.07	6.370

TED : MT German-English (MT$_{DeEn}$)

"case+punc" evaluation							System	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
29.84	**66.28**	**52.78**	**41.71**	**49.05**	**63.74**	**7.053**	RWTH	**28.85**	63.73	**54.90**	**43.25**	**52.10**	**62.20**	**7.269**
28.80	66.23	53.85	42.21	50.01	63.38	6.930	UEDIN	28.45	**64.00**	55.75	43.57	52.74	61.86	7.153
28.18	65.41	55.48	43.60	51.67	62.72	6.771	FBK	27.76	62.88	57.37	44.84	54.41	61.08	7.003
27.97	64.66	55.14	43.56	51.53	62.30	6.754	NAIST	26.95	62.00	57.54	45.31	54.66	60.36	6.934

TED : MT Dutch-English (MT$_{NlEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
32.69	**67.59**	**50.12**	**39.45**	**46.15**	**65.51**	**7.463**	FBK	**31.96**	**65.19**	**51.76**	**40.55**	**49.12**	**64.47**	**7.714**
30.97	66.14	51.80	40.94	47.68	64.09	7.238	NAIST	30.29	63.74	53.64	42.10	50.84	63.06	7.471

TED : MT Polish-English (MT$_{PlEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
16.66	**49.90**	**70.49**	**58.21**	**66.88**	**49.55**	**5.062**	NAIST	**15.33**	**46.27**	**73.38**	**60.60**	**71.04**	**47.08**	**5.151**
15.32	47.94	71.85	59.61	67.97	48.32	4.844	PJIIT	14.28	44.08	73.88	61.18	71.53	46.14	4.983

TED : MT Portuguese(Brazilian)-English (MT$_{PtbEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
41.67	**75.91**	**39.84**	**32.60**	**37.82**	**72.05**	**8.318**	NAIST	**40.01**	**73.45**	**42.77**	**34.89**	**41.29**	**70.02**	**8.399**

TED : MT Romanian-English (MT$_{RoEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
29.64	**65.19**	**52.41**	**43.06**	**49.90**	**62.91**	**6.931**	NAIST	**27.59**	**61.93**	**56.13**	**45.93**	**54.27**	**60.27**	**6.951**
27.00	64.46	56.30	46.20	51.09	60.03	6.514	RACAI	26.92	61.36	56.95	46.50	55.02	59.85	6.894

TED : MT Russian-English (MT$_{RuEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
18.31	**52.37**	**65.74**	**54.53**	**62.52**	**51.75**	**5.332**	NAIST	**16.97**	**48.67**	**68.59**	**57.06**	**66.57**	**49.22**	**5.385**
10.24	40.31	70.60	60.93	67.76	47.06	2.979	NICT	08.89	35.74	74.43	65.70	72.67	42.71	2.251

TED : MT Slovak-English (MT$_{SkEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
21.50	52.85	**62.26**	**54.34**	**59.38**	**54.11**	**5.545**	FBK	20.82	50.11	**64.41**	**56.29**	**62.48**	51.78	**5.686**
20.55	**53.91**	66.76	58.42	60.68	50.93	5.168	NAIST	**21.43**	**51.51**	65.89	56.89	63.85	**52.12**	5.685
16.24	53.63	68.31	61.41	59.84	47.42	4.691	RWTH	19.71	50.08	65.77	57.65	63.97	51.29	5.593

TED : MT Turkish-English (MT$_{TrEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
17.16	**53.51**	**74.32**	**52.32**	**66.65**	**54.61**	**5.551**	FBK	**16.06**	**50.37**	**77.81**	**54.53**	**70.86**	**52.43**	**5.691**
14.87	50.47	77.47	55.41	69.79	51.86	5.148	NAIST	13.66	47.16	81.37	57.78	74.37	49.44	5.256
12.86	47.36	80.04	58.58	72.78	48.90	4.745	TUBITAK	11.96	43.79	83.23	60.69	76.89	46.45	4.876

TED : MT Chinese-English (MT$_{ZhEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
15.08	**49.76**	**69.52**	**56.64**	**65.05**	**49.73**	**4.931**	RWTH	**13.95**	**45.97**	**73.08**	**59.58**	**69.84**	**47.18**	**4.904**
12.04	45.62	71.78	59.10	67.82	46.76	4.364	NAIST	10.91	41.47	75.59	62.49	72.91	43.74	4.222

OLYMPICS : MT Chinese-English (MT$_{ZhEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
19.17	**53.79**	**66.88**	**56.34**	**61.36**	**51.51**	**4.777**	HIT	**18.85**	**48.90**	**72.21**	**59.26**	**68.85**	**49.85**	**5.197**
16.95	50.21	69.82	59.18	65.42	49.79	4.531	NICT	16.37	45.55	75.85	63.28	72.65	46.55	4.749
12.79	46.34	75.46	63.92	71.10	45.94	3.994	KYOTO-U	12.38	41.44	82.83	68.54	79.74	43.06	4.177
12.16	38.90	84.14	71.98	79.68	43.67	3.631	POSTECH	10.89	32.38	92.71	78.64	89.66	39.22	3.650

A.2. Progress Testset (*tst2011*)

· All the sentence IDs in the IWSLT 2011 testset were used to calculate the automatic scores for each run submission.
· ASR and MT systems are ordered according to the *WER* and *BLEU* metrics, respectively.
· For each task, the best score of each metric is marked with **boldface**.
· Besides the NIST metrics, all automatic evaluation metric scores are given as percent figures (%).
· Besides the ASR scores, the mean scores of 2000 iterations were calculated for each MT output according to the *bootStrap* method [48].
· Omitted lines between scores indicate non-significant differences in performance between the MT engines.

TED : ASR English (ASR$_{En}$)

System	WER	(Count)
NICT	**10.9**	(1401)
MITLL	11.1	(1432)
KIT	12.0	(1552)
KIT-NAIST	12.0	(1553)
UEDIN	12.4	(1599)
RWTH	13.4	(1731)
FBK	15.4	(1991)

IWSLT 2011	WER	(Count)
MITLL	13.5	(1741)
KIT	15.0	(1938)
LIUM	15.4	(1992)
FBK	16.2	(2091)
NICT	25.6	(3301)

TED : SLT English-French (SLT$_{EnFr}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
28.85	**58.25**	**54.63**	**46.32**	**52.07**	**58.96**	**6.360**	KIT	**29.60**	**56.87**	**55.10**	**47.10**	**53.67**	**58.22**	**6.619**
27.83	56.37	55.87	47.43	53.38	58.15	6.298	RWTH	28.62	55.24	56.15	48.17	54.74	57.35	6.524
26.53	56.19	56.57	48.00	54.06	57.27	6.130	UEDIN	27.65	55.07	56.76	48.55	55.36	56.54	6.377
24.28	54.75	61.40	51.49	58.75	55.59	5.711	MITLL	24.86	53.71	62.31	52.55	60.69	54.69	5.873

TED : MT English-French (MT$_{EnFr}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
39.00	**67.73**	43.79	36.97	41.56	67.48	7.483	UEDIN	**37.86**	**64.64**	46.19	39.20	44.90	65.43	7.583
38.64	67.11	**42.98**	**36.75**	**40.88**	**67.69**	**7.607**	RWTH	37.37	63.90	**45.47**	39.11	44.38	65.59	7.681
38.49	67.12	43.08	36.86	41.00	67.59	7.587	KIT	37.35	64.09	45.53	**39.10**	**44.27**	**65.67**	**7.691**
37.90	66.62	43.90	37.58	41.79	66.88	7.442	NAIST	36.63	63.53	46.87	39.93	45.59	64.80	7.514
37.43	66.10	44.78	37.94	42.80	66.53	7.375	FBK	35.86	62.89	47.88	40.62	46.54	64.15	7.419
36.87	66.08	44.13	37.48	42.04	66.87	7.437	LIG	35.66	62.78	47.09	39.98	45.79	64.60	7.492
31.43	62.92	52.07	43.45	49.67	62.60	6.535	MITLL	30.09	59.49	55.78	46.42	54.19	60.14	6.568

TED : MT Arabic-English (MT$_{ArEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
27.29	**62.11**	**56.96**	**47.23**	**54.08**	**59.40**	**6.409**	RWTH	**26.25**	**59.76**	**59.00**	**48.76**	**57.33**	**58.16**	**6.519**
25.47	59.61	60.38	50.20	57.73	57.56	6.029	FBK	24.03	57.03	63.06	52.38	61.44	55.76	6.058
23.85	58.45	59.96	49.65	57.09	56.84	5.990	TUBITAK	22.43	55.68	62.96	52.14	61.10	54.78	6.006
23.66	58.52	61.79	51.39	58.85	56.46	5.826	NAIST	22.20	55.58	64.94	54.15	63.26	54.13	5.814
18.00	58.18	66.37	56.41	59.20	50.96	4.949	MITLL	21.38	55.14	65.05	53.25	63.04	54.44	5.830

TED : MT German-English (MT$_{DeEn}$)

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
34.02	**70.46**	**48.05**	**37.99**	**44.50**	**67.03**	**7.426**	RWTH	**32.98**	**68.00**	**50.25**	**39.68**	**47.70**	**65.53**	**7.587**
32.42	70.32	49.91	38.28	45.77	66.99	7.311	UEDIN	31.68	67.94	52.17	39.99	48.94	65.42	7.450
32.38	69.87	50.30	39.06	46.56	66.68	7.243	FBK	31.77	67.56	52.28	40.53	49.32	65.14	7.421
31.53	69.21	50.87	39.34	46.83	66.10	7.193	NAIST	30.82	66.69	53.00	41.06	49.94	64.43	7.355

TED : MT Dutch-English (MT_{NlEn})

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
36.11	**71.40**	**47.94**	**37.51**	**43.91**	**67.81**	**7.623**	FBK	**35.30**	**69.30**	**49.70**	**38.56**	**47.06**	**66.95**	**7.842**
34.63	70.48	49.20	38.55	44.99	66.64	7.436	NAIST	33.82	68.21	51.24	39.72	48.49	65.77	7.632

TED : MT Polish-English (MT_{PlEn})

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
20.27	**55.81**	**66.07**	**53.92**	**62.49**	**54.13**	**5.484**	NAIST	**19.27**	**52.31**	**68.92**	**55.94**	**66.54**	**51.97**	**5.587**
18.65	53.61	68.11	55.42	64.19	53.10	5.279	PJIIT	18.00	50.30	69.91	56.86	67.45	51.12	5.469

TED : MT Portuguese(Brazilian)-English (MT_{PtbEn})

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
39.72	**75.06**	**41.67**	**34.11**	**39.45**	**71.04**	**7.990**	NAIST	**37.96**	**72.58**	**44.60**	**36.40**	**42.97**	**69.05**	**8.007**

TED : MT Romanian-English (MT_{RoEn})

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
33.62	**69.57**	**47.48**	**38.53**	**44.79**	**66.71**	**7.402**	NAIST	**31.84**	**66.62**	**50.62**	**40.92**	**48.79**	**64.58**	**7.447**
29.93	68.44	52.13	42.06	46.71	63.45	6.881	RACAI	30.10	65.57	52.58	42.05	50.53	63.61	7.266

TED : MT Russian-English (MT_{RuEn})

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
20.17	**55.09**	**64.35**	**52.91**	**61.14**	**53.76**	**5.436**	NAIST	**18.54**	**51.36**	**67.46**	**55.40**	**65.26**	**51.06**	**5.479**
11.52	42.37	68.93	58.62	66.03	49.02	3.473	NICT	09.97	38.04	72.56	63.22	70.83	44.80	2.791

TED : MT Turkish-English (MT_{TrEn})

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
17.23	**52.85**	**75.46**	**53.62**	**67.71**	**54.39**	**5.411**	FBK	**16.02**	**49.73**	**78.79**	**55.64**	**71.92**	**52.32**	**5.522**
15.04	50.02	79.38	57.42	71.74	51.55	4.965	NAIST	13.95	46.86	83.08	59.39	76.18	49.34	5.060
13.30	47.66	81.47	58.86	73.70	49.64	4.709	TUBITAK	12.34	44.19	84.41	60.48	77.63	47.59	4.847

TED : MT Chinese-English (MT_{ZhEn})

"case+punc" evaluation							**System**	*"no_case+no_punc" evaluation*						
BLEU	METEOR	WER	PER	TER	GTM	NIST		BLEU	METEOR	WER	PER	TER	GTM	NIST
17.20	**52.21**	**67.25**	**54.70**	**62.86**	**51.92**	**5.189**	RWTH	**15.67**	**48.36**	**70.65**	**57.43**	**67.48**	**49.41**	**5.128**
13.74	48.01	69.51	57.22	65.77	49.17	4.628	NAIST	12.12	43.84	73.27	60.58	70.71	45.95	4.463

Appendix B. Human Evaluation

B.1. TED SLT English-French Task - Progress Testset (*tst2011*)

System Ranking

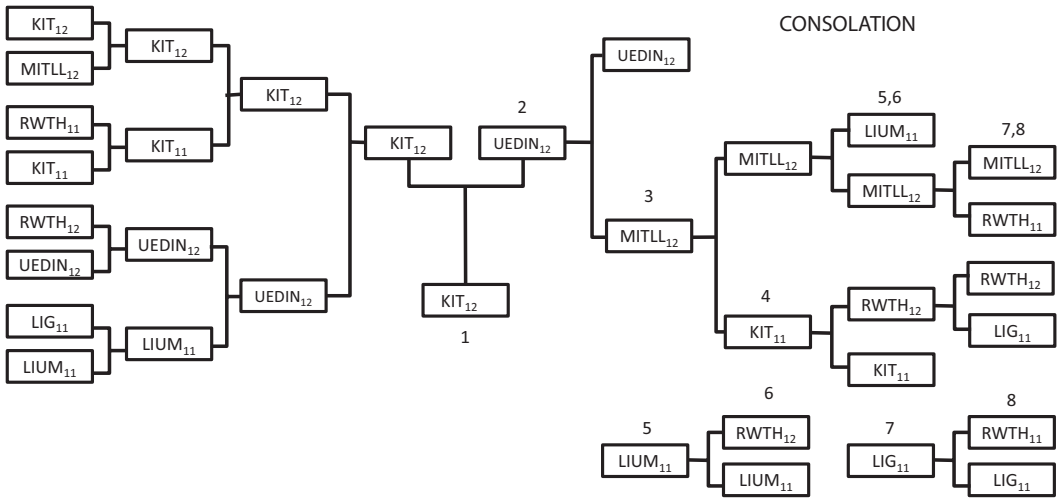
BLEU Ranking
(used for tournament seeding)

Ranking	System	BLEU score
1	KIT$_{12}$	28.86
2	LIUM$_{11}$	28.23
3	RWTH$_{12}$	27.85
4	KIT$_{11}$	26.78
5	RWTH$_{11}$	26.76
6	UEDIN$_{12}$	26.54
7	LIG$_{11}$	24.85
8	MITLL$_{12}$	24.27

Human Ranking
(resulting from tournament)

Ranking	System
1	KIT$_{12}$
2	UEDIN$_{12}$
3	MITLL$_{12}$
4	KIT$_{11}$
5	LIUM$_{11}$
6	RWTH$_{12}$
7	LIG$_{11}$
8	RWTH$_{11}$

Double Seeded Knockout with Consolation Tournament



Head to Head Matches Evaluation

· Head to Head matches: Wins indicate the percentage of times that one system was judged to be better than the other. The winner of the two systems is indicated in bold. The difference between 100 and the sum of the systems'wins corresponds to the percentage of ties.
· Statistical significance: † indicates statistical significance at $p \leq 0.10$, ‡ indicates statistical significance at $p \leq 0.05$, and ⋆ indicates statistical significance at $p \leq 0.01$, according to the Approximate Randomization Test based on 10,000 iterations.
· Inter Annotator Agreement: calculated using *Fleiss' kappa coefficient.*

HtH Matches	% Wins	I.A.A.
KIT$_{11}$- KIT$_{12}$	KIT$_{11}$: 23.75 KIT$_{12}$: **41.75***	0.1916
KIT$_{11}$- MITLL$_{12}$	KIT$_{11}$:28.50 MITLL$_{12}$: **33.50**	0.1716
LIG$_{11}$- RWTH$_{12}$	LIG$_{11}$: 31.25 RWTH$_{12}$: **31.75**	0.1993
LIUM$_{11}$- UEDIN$_{12}$	LIUM$_{11}$: 38.00 UEDIN$_{12}$: **38.00** *(a)*	0.1887
RWTH$_{11}$- MITLL$_{12}$	RWTH$_{11}$: 28.25 MITLL$_{12}$: **30.50**	0.1415

HtH Matches	% Wins	I.A.A.
MITLL$_{12}$- LIUM$_{11}$	MITLL$_{12}$: **39.75** LIUM$_{11}$: 37.50	0.2025
MITLL$_{12}$- KIT$_{12}$	MITLL$_{12}$: 18.00 KIT$_{12}$: **25.50**‡	0.3730
RWTH$_{12}$- KIT$_{11}$	RWTH$_{12}$: 37.50 KIT$_{11}$: **38.00**	0.2413
RWTH$_{12}$- LIUM$_{11}$	RWTH$_{12}$: 27.00 LIUM$_{11}$: **36.00**‡	0.2245
UEDIN$_{12}$- KIT$_{12}$	UEDIN$_{12}$: 37.25 KIT$_{12}$: **41.75**	0.2760

HtH Matches	% Wins	I.A.A.
UEDIN$_{12}$- MITLL$_{12}$	UEDIN$_{12}$: **40.75** MITLL$_{12}$: 34.50	0.2618
UEDIN$_{12}$- RWTH$_{12}$	UEDIN$_{12}$: **19.25** RWTH$_{12}$: 16.00	0.4009
RWTH$_{11}$- KIT$_{11}$	RWTH$_{11}$: 24.00 KIT$_{11}$: **27.75**	0.1784
LIG$_{11}$- LIUM$_{11}$	LIG$_{11}$: 21.55 LIUM$_{11}$: **30.08**‡	0.1743
RWTH$_{11}$- LIG$_{11}$	RWTH$_{11}$: 26.88 LIG$_{11}$: 29.65	0.1697

(a) Total number of wins considering all the judgments by the three annotators: UEDIN$_{12}$= 475; LIUM$_{11}$= 461.

B.2. TED MT Arabic-English Task - Progress Testset (*tst2011*)

System Ranking

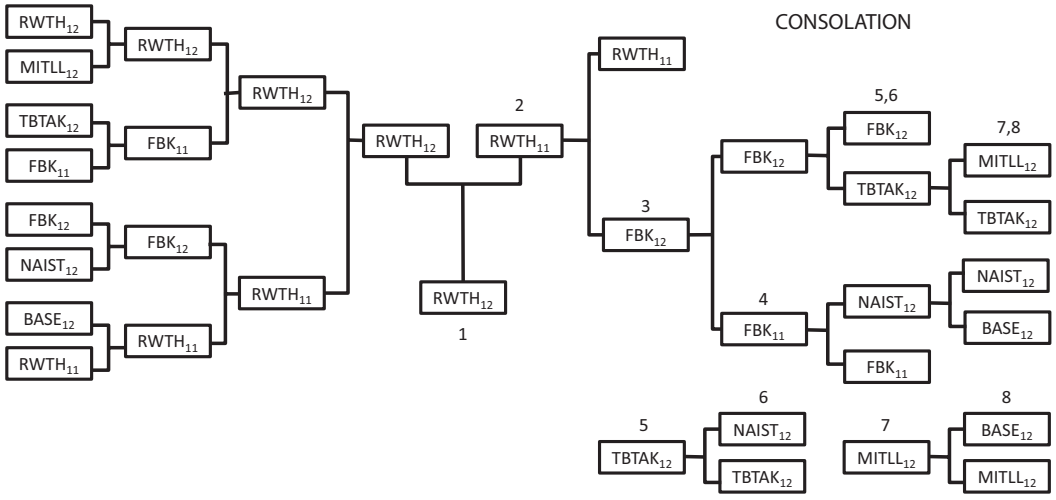
BLEU Ranking
(used for tournament seeding)

Ranking	System	BLEU score
1	RWTH$_{12}$	27.28
2	RWTH$_{11}$	26.32
3	FBK$_{12}$	25.46
4	FBK$_{11}$	24.31
5	TUBITAK$_{12}$	23.85
6	NAIST$_{12}$	23.65
7	BASELINE$_{12}$	22.08
8	MITLL$_{12}$	17.99

Human Ranking
(resulting from tournament)

Ranking	System
1	RWTH$_{12}$
2	RWTH$_{11}$
3	FBK$_{12}$
4	FBK$_{11}$
5	TUBITAK$_{12}$
6	NAIST$_{12}$
7	MITLL$_{12}$
8	BASELINE$_{12}$

Double Seeded Knockout with Consolation Tournament



Head to Head Matches Evaluation

· Head to Head matches: Wins indicate the percentage of times that one system was judged to be better than the other. The winner of the two systems is indicated in bold. The difference between 100 and the sum of the systems'wins corresponds to the percentage of ties.
· Statistical significance: † indicates statistical significance at $p \leq 0.10$, ‡ indicates statistical significance at $p \leq 0.05$, and ⋆ indicates statistical significance at $p \leq 0.01$, according to the Approximate Randomization Test based on 10,000 iterations.
· Inter Annotator Agreement: calculated using *Fleiss' kappa coefficient.*

HtH Matches	% Wins	I.A.A.
FBK$_{11}$- FBK$_{12}$	FBK$_{11}$: 23.75 FBK$_{12}$: **24.75**	0.2766
MITLL$_{12}$- RWTH$_{12}$	MITLL$_{12}$: 12.50 RWTH$_{12}$: **59.00***	0.2834
FBK$_{11}$- NAIST$_{12}$	FBK$_{11}$: **37.50*** NAIST$_{12}$: 18.25	0.2693
FBK$_{11}$- RWTH$_{12}$	FBK$_{11}$: 21.50 RWTH$_{12}$: **40.75***	0.2417
FBK$_{11}$- TUBITAK$_{12}$	FBK$_{11}$: **41.00*** TUBITAK$_{12}$: 26.50	0.1971

HtH Matches	% Wins	I.A.A.
NAIST$_{12}$- FBK$_{12}$	NAIST$_{12}$: 20.50 FBK$_{12}$: **47.25***	0.2352
NAIST$_{12}$- TUBITAK$_{12}$	NAIST$_{12}$: 24.00 TUBITAK$_{12}$: **24.00** [(a)]	0.2545
TUBITAK$_{12}$- FBK$_{12}$	TUBITAK$_{12}$: 18.25 FBK$_{12}$: **37.25***	0.2937
TUBITAK$_{12}$- MITLL$_{12}$	TUBITAK$_{12}$: **39.75*** MITLL$_{12}$: 19.75	0.2030
RWTH$_{11}$- FBK$_{12}$	RWTH$_{11}$: **38.50*** FBK$_{12}$: 25.25	0.2297

HtH Matches	% Wins	I.A.A.
RWTH$_{11}$- RWTH$_{12}$	RWTH$_{11}$: 20.25 RWTH$_{12}$: **27.25**†	0.3236
RWTH$_{11}$- BASELINE$_{12}$	RWTH$_{11}$: **58.75*** BASELINE$_{12}$: 10.25	0.2654
BASELINE$_{12}$- MITLL$_{12}$	BASELINE$_{12}$: 16.50 MITLL$_{12}$**25.25***	0.1933
BASELINE$_{12}$- NAIST$_{12}$	BASELINE$_{12}$:17.75 NAIST$_{12}$: **37.25***	0.2284

(a) Total number of wins considering all the judgments by the three annotators: TUBITAK$_{12}$= 358; NAIST$_{12}$= 327.

B.3.1 TED MT English-French - Progress Testset (*tst2011*)

· First tournament: all 2012 systems to determine the top four ones.

System Ranking

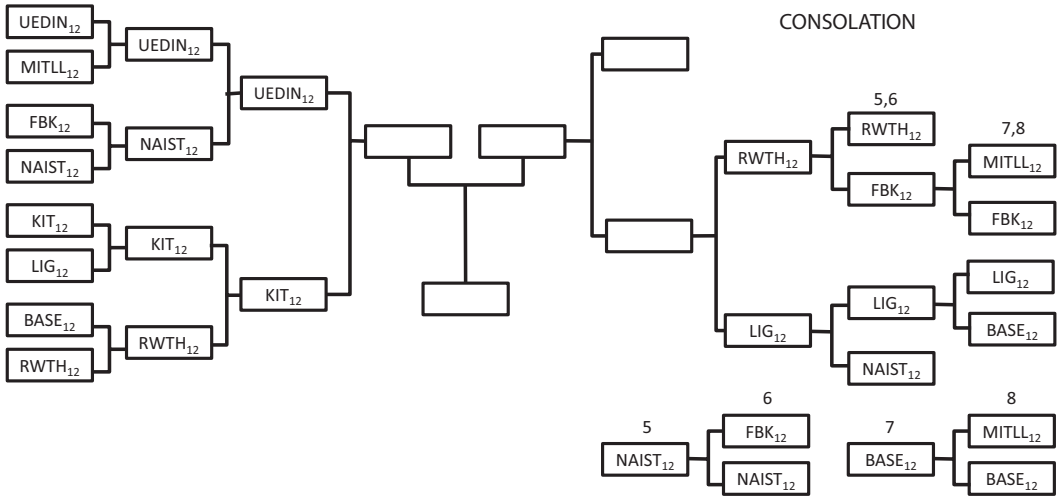
BLEU Ranking
(used for tournament seeding)

Ranking	System	BLEU score
1	UEDIN$_{12}$	39.01
2	RWTH$_{12}$	38.66
3	KIT$_{12}$	38.49
4	NAIST$_{12}$	37.90
5	FBK$_{12}$	37.43
6	LIG$_{12}$	36.88
7	BASELINE$_{12}$	33.90
8	MITLL$_{12}$	31.44

Human Ranking
(resulting from tournament)

Ranking	System
	KIT$_{12}$
	LIG$_{12}$
	RWTH$_{12}$
	UEDIN$_{12}$
5	NAIST$_{12}$
6	FBK$_{12}$
7	BASELINE$_{12}$
8	MITLL$_{12}$

Double Seeded Knockout with Consolation Tournament



Head to Head Matches Evaluation

· Head to Head matches: Wins indicate the percentage of times that one system was judged to be better than the other. The winner of the two systems is indicated in bold. The difference between 100 and the sum of the systems'wins corresponds to the percentage of ties.
· Statistical significance: † indicates statistical significance at $p \leq 0.10$, ‡ indicates statistical significance at $p \leq 0.05$, and ⋆ indicates statistical significance at $p \leq 0.01$, according to the Approximate Randomization Test based on 10,000 iterations.
· Inter Annotator Agreement: calculated using *Fleiss' kappa coefficient.*

HtH Matches	% Wins	I.A.A.
BASELINE$_{12}$- LIG$_{12}$	BASELINE$_{12}$: 24.75 LIG$_{12}$: **45.75***	0.1665
BASELINE$_{12}$- MITLL$_{12}$	BASELINE$_{12}$: **39.75** MITLL$_{12}$: 32.75	0.1963
FBK$_{12}$- MITLL$_{12}$	FBK$_{12}$: **43.50**‡ MITLL$_{12}$: 32.75	0.1508
FBK$_{12}$- RWTH$_{12}$	FBK$_{12}$: 27.25 RWTH$_{12}$: **36.75**‡	0.2500

HtH Matches	% Wins	I.A.A.
LIG$_{12}$- KIT$_{12}$	LIG$_{12}$: 26.00 KIT$_{12}$: **33.50**†	0.2921
MITLL$_{12}$- UEDIN$_{12}$	MITLL$_{12}$: 16.50 UEDIN$_{12}$: **47.50***	0.2367
NAIST$_{12}$- UEDIN$_{12}$	NAIST$_{12}$: 20.50 UEDIN$_{12}$: **33.00***	0.4014
NAIST$_{12}$- FBK$_{12}$	NAIST$_{12}$: **34.75**‡ FBK$_{12}$: 25.25	0.3085

HtH Matches	% Wins	I.A.A.
NAIST$_{12}$- LIG$_{12}$	NAIST$_{12}$: 32.00 LIG$_{12}$**34.50**	0.2622
RWTH$_{12}$- BASELINE$_{12}$	RWTH$_{12}$: **34.25*** BASELINE$_{12}$: 22.25	0.2298
RWTH$_{12}$- KIT$_{12}$	RWTH$_{12}$: 32.50 KIT$_{12}$: **33.50**	0.3218

B.3.2 TED MT English-French Progressive Task - Progress Testset (*tst2011*)

· Second tournament: the four top-ranked 2012 systems the four top-ranked 2011 systems

System Ranking

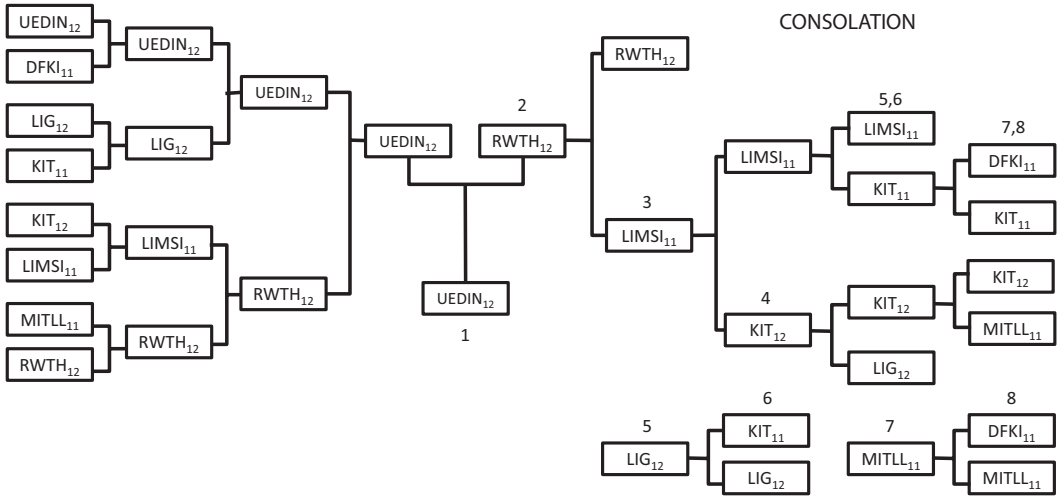
BLEU Ranking
(used for tournament seeding)

Ranking	System	BLEU score
1	UEDIN$_{12}$	39.01
2	RWTH$_{12}$	38.66
3	KIT$_{12}$	38.49
4	KIT$_{11}$	37.65
5	LIG$_{12}$	36.88
6	LIMSI$_{11}$	36.49
7	MITLL$_{11}$	35.28
8	DFKI$_{11}$	34.39

Human Ranking
(resulting from tournament)

Ranking	System
1	UEDIN$_{12}$
2	RWTH$_{12}$
3	LIMSI$_{11}$
4	KIT$_{12}$
5	LIG$_{12}$
6	KIT$_{11}$
7	MITLL$_{11}$
8	DFKI$_{11}$

Double Seeded Knockout with Consolation Tournament



Head to Head Matches Evaluation

· Head to Head matches: Wins indicate the percentage of times that one system was judged to be better than the other. The winner of the two systems is indicated in bold. The difference between 100 and the sum of the systems'wins corresponds to the percentage of ties.
· Statistical significance: † indicates statistical significance at $p \leq 0.10$, ‡ indicates statistical significance at $p \leq 0.05$, and ⋆ indicates statistical significance at $p \leq 0.01$, according to the Approximate Randomization Test based on 10,000 iterations.
· Inter Annotator Agreement: calculated using *Fleiss' kappa coefficient.*

HtH Matches	% Wins	I.A.A.
DFKI$_{11}$- UEDIN$_{12}$	DFKI$_{11}$: 22.75 UEDIN$_{12}$: **46.00***	0.2681
LIG$_{12}$- UEDIN$_{12}$	LIG$_{12}$: 23.00 UEDIN$_{12}$: **39.50***	0.2871
RWTH$_{12}$- LIMSI$_{11}$	RWTH$_{12}$: **35.25** LIMSI$_{11}$: 34.25	0.2625
RWTH$_{12}$- MITLL$_{11}$	RWTH$_{12}$: **39.25** MITLL$_{11}$: 33.75	0.2794

HtH Matches	% Wins	I.A.A.
KIT$_{11}$- LIG$_{12}$	KIT$_{11}$: 35.00 LIG$_{12}$: **37.75**	0.3218
LIMSI$_{11}$- KIT$_{12}$	LIMSI$_{11}$: **42.75** KIT$_{12}$: 38.50	0.2779
MITLL$_{11}$- KIT$_{12}$	MITLL$_{11}$: 28.75 KIT$_{12}$: **41.75***	0.2347
RWTH$_{12}$- UEDIN$_{12}$	RWTH$_{12}$: 23.50 UEDIN$_{12}$: **32.00**‡	0.3296

HtH Matches	% Wins	I.A.A.
DFKI$_{11}$- MITLL$_{11}$	DFKI$_{11}$: 40.00 MITLL$_{11}$: **42.50**	0.3777
KIT$_{11}$- LIMSI$_{11}$	KIT$_{11}$: 41.25 LIMSI$_{11}$: **43.50**	0.4154
DFKI$_{11}$- KIT$_{11}$	DFKI$_{11}$: 42.25 KIT$_{11}$: **43.00**	0.4235

B.4. OLYMPICS MT Chinese-English Task

System Ranking

· A subset of 400 test sentences was used to carry out the subjective ranking evaluation.
· The *"All systems"* scores indicate the average number of times that a system was judged better then ($>$) or better/equal to ($\geq$) any other system.
· The *"Head to head"* scores indicate the number of pairwise head-to-head comparisons won by a system.

System	ALL SYSTEMS $>$ others	ALL SYSTEMS $\geq$ others
HIT	**0.3808**	**0.8642**
NAIST-NICT	0.3025	0.8242
KYOTO-U	0.2150	0.7242
POSTECH	0.0850	0.6042

System	HEAD-TO-HEAD # wins
HIT	3 / 3
NAIST-NICT	2 / 3
KYOTO-U	1 / 3
POSTECH	0 / 3

Head to Head Matches Evaluation

· Head to Head matches: Wins indicate the percentage of times that one system was judged to be better than the other. The winner of the two systems is indicated in bold. The difference between 100 and the sum of the systems'wins corresponds to the percentage of ties.
· Statistical significance: † indicates statistical significance at $p \leq 0.10$, ‡ indicates statistical significance at $p \leq 0.05$, and $\star$ indicates statistical significance at $p \leq 0.01$, according to the Approximate Randomization Test based on 10,000 iterations.
· Inter Annotator Agreement: calculated using *Fleiss' kappa coefficient.*

HtH Matches	% Wins	I.A.A.
HIT- POSTECH	HIT: **47.75*** POSTECH: 6.25	0.3881
NAIST-NICT- KYOTO-U	NAIST-NICT: **32.50*** KYOTO-U: 17.25	0.3251
NAIST-NICT- HIT	NAIST-NICT: 17.75 HIT: **29.50***	0.3484

HtH Matches	% Wins	I.A.A.
KYOTO-U- HIT	KYOTO-U: 16.75 HIT: **37.00***	0.3819
KYOTO-U- POSTECH	KYOTO-U: **30.50*** POSTECH: 13.25	0.3722
NAIST-NICT- POSTECH	NAIST-NICT: **40.50*** POSTECH: 6.00	0.3616

Dialog Adequacy

(best = 5.0, . . ., worst = 1.0)

The following tables show how much of the information from the input sentence was expressed in the translation with (*adequacy*) and without (*dialog*) taking into account the context of the respective dialog.

OLYMPICS	**MT**	**Adequacy**	**Dialog**
MT$_{ZhEn}$	HIT	**3.17**	**3.42**
	NAIST-NICT	3.00	
	KYOTO-U	2.90	
	POSTECH	2.49	

The NICT ASR System for IWSLT2012

Hitoshi Yamamoto, Youzheng Wu, Chien-Lin Huang, Xugang Lu, Paul R. Dixon, Shigeki Matsuda, Chiori Hori, Hideki Kashioka

Spoken Language Communication Laboratory,
National Institute of Information and Communication Technology,
Kyoto, Japan

hitoshi.yamamoto@nict.go.jp

Abstract

This paper describes our automatic speech recognition (ASR) system for the IWSLT 2012 evaluation campaign. The target data of the campaign is selected from the TED talks, a collection of public speeches on a variety of topics spoken in English. Our ASR system is based on weighted finite-state transducers and exploits an combination of acoustic models for spontaneous speech, language models based on n-gram and factored recurrent neural network trained with effectively selected corpora, and unsupervised topic adaptation framework utilizing ASR results. Accordingly, the system achieved 10.6% and 12.0% word error rate for the tst2011 and tst2012 evaluation set, respectively.

1. Introduction

This paper describes our automatic speech recognition (ASR) system for the IWSLT 2012 evaluation campaign.

The target speech data of the ASR track of the campaign is selected from TED talks, a collection of short presentations to an audience spoken in English. These talks are generally in spontaneous speaking style, which touch on a variety of topics related to Technology, Entertainment and Design (TED). Main challenges of the track are clean transcription of spontaneous speech, detection and removal of non-words, and talk style and topic adaptation [1].

An overview of our ASR system is depicted in Figure 1. The core decoder of the system is based on weighted finite-state transducers (WFSTs). It exploits two types of state-of-the-art acoustic models (AMs) of spontaneous speech which are integrated in lattice level. Here, n-gram language models (LMs) are trained with in-domain and effectively selected out-of-domain corpora. Then, it employs recurrent neural network (RNN) based LMs newly extended to incorporate additional linguistic features. Finally, it utilizes ASR results to adapt LMs to talk style and topic.

This paper is organized as follows. Section 2 explains the training data and procedure of AMs in the system. Section 3 presents an overview of the data and technique used to build and adapt our LMs. Section 4 describes decoding strategy and experimental results.

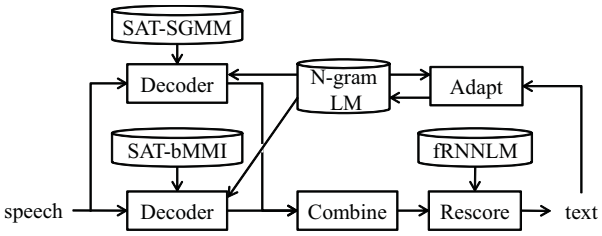


Figure 1: *Overview of the NICT ASR system for IWSLT2012.*

2. Acoustic Modeling

2.1. Training Corpus

To train AMs suitable for TED talks, we crawled movies and subtitles of talks published prior to 2011 from the TED website[1]. The collected 777 talks contain 204 hours audio and $1.8M$ words, excluding 19 talks of the development set (dev2010, tst2010).

For each talk, the subtitle is aligned to the audio of the movie because it doesn't contain accurate time stamps of speech segments for training phoneme-level acoustic models. We utilize SailAlign [2] to extract text-aligned speech segments from the audio data. As shown in Figure 2, it iterates two steps, (a) text-based alignment of ASR results and transcriptions and (b) ASR model adaptation using text-aligned speech segments. Here it runs with its basic setting, using HTK and AM trained on WSJ. After two iterations, 170 hours of text-aligned speech segments (with $1.6M$ words) are defined as AM training corpus.

2.2. Training Procedure

The acoustic feature vector has 40 dimensions. We first extract 13 static MFCCs including zeroth order for each frame (25ms width and 10ms shift) and normalize them with cepstrum mean normalization for each talk. Then, for each frame, we concatenate MFCCs of 9 adjacent frames (4 on each side of the current frame) and apply transformation matrix based on linear discriminant analysis (LDA) and maxi-

[1] http://www.ted.com/

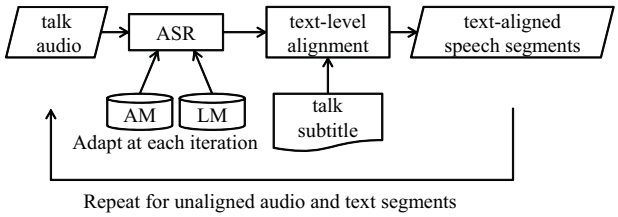


Figure 2: *Adaptive and iterative scheme of SailAlign [2].*

mum likelihood linear transformation (MLLT) to reduce its dimension to 40. In addition, we apply feature space MLLR for speaker adaptive training for each talk, assuming that one talk includes one speaker.

The acoustic models are cross-word triphone HMMs of which units are derived from 39 phonemes. Each phoneme is classified by its position in word (4 classes: begin, end, singleton and the others) and each vowel is further distinguished by its accent mark (3 classes: first, second and the others).

Three types of acoustic models are developed with the Kaldi speech recognition toolkit [3] revision 941. We first train HMMs with GMM output probability. This model totally include 6.7K states and 80K Gaussians trained with ML estimation (SAT-ML). Then we increase the number of Gaussian of it to $240K$ (other parts are not changed) and train them with boosted MMI criterion (SAT-bMMI). We also build HMMs with subspace GMM output probability. This model consists of $9.1K$ states, which is transformed from the SAT-ML model (SAT-SGMM).

3. Language Modeling

3.1. Training Corpus

The IWSLT evaluation campaign defines a closed set of publicly available English texts as training data of LM. We use the in-domain corpus (transcription of TED talks) and parts of the out-of-domain corpora (English Gigaword Fifth Edition and News Commentary v7) and pre-process the data as follows: (1) converting non-standard words (such as CO2 or 95%) to their pronunciations (CO two, ninety five percent) using a non-standard-word expansion tool[2] [4], and (2) removing duplicated sentences. The statistics of the preprocessed corpora are shown in Table 1.

The lexicon consists of the CMU Pronouncing Dictionary[3] v.0.7a. In addition, we extract new words (not included in the CMU dictionary) from the preprocessed in-domain corpora and generate their pronunciations with a WFST-based grapheme-to-phoneme (G2P) technique [5]. The extended lexicon contains $156.3K$ pronunciation entries of $133.3K$ words which are used as the LM vocabulary with an OOV rate of 0.8% on the dev2010 data set.

[2] http://festvox.org/nsw

[3] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Table 1: *Statistics of English LM training corpora*

	Corpus	#sentences	#words
in-domain	TED Talks	$142K$	2,402K
out-of-domain	News Commentary	$212K$	4,566K
	English Gigaword	$123M$	2,722M

3.2. Domain adaptation

The large out-of-domain corpora likely includes sentences that are so unlike the domain of the TED talks. LM trained on these unlike sentences is probably harmful. Therefore, we adopt domain adaptation by selecting only a portion of the out-of-domain corpus instead of using the whole.

We employ cross-entropy difference metric for domain adaptation, which biases towards sentences that are both like the in-domain corpus and unlike the average of the out-of-domain corpus [6]. Each sentence s of the out-of-domain corpus is scored as follows,

$$H_I(s) - H_O(s), \tag{1}$$

where $H_I(s)$ and $H_O(s)$ represent cross-entropy scores according to LM_I trained on the in-domain corpus, and LM_O trained on a subset sentences randomly selected from the out-of-domain corpus. Here, LM_I and LM_O are similar size. Then the lowest-scoring sentences are selected as a subset of out-of-domain corpus.

3.3. N-gram LM

For the in-domain and the selected out-of-domain corpora, modified Kneser-Ney smoothed n-gram LMs (n=3,4) are constructed using SRILM [7]. They are interpolated to form a baseline of n-gram LMs by optimizing the perplexity of the development data set. To apply the domain adaptation, we empirically select $1/4$ of the out-of-domain corpus with $30M$ sentences and $559M$ words using Eq. (1).

3.4. Factored RNNLM

Recently, recurrent neural network (RNN) based LMs [8] become an increasingly popular choice for LVCSR tasks due to consistent improvements. In our system, we employ a factored RNNLM that exploits additional linguistic information, including morphological, syntactic, or semantic. This novel approach was proposed in our previous studies [9].

In the official run, our factored RNNLM uses two types of features, word surface and part-of-speech tagged by GENIA Tagger[4]. Other types of linguistic features are investigated in [10]. We set the number of hidden neurons in the hidden layer and the number of classes in the output layer to 480 and 300.

Since it is very time consuming to train factored RNNLM on large data, we select a subset sentences of the out-of-

[4] http://www.nactem.ac.uk/tsujii/software.html

Table 2: *Word error rate (WER, %) of the development sets and test sets. The results of primary run in our submission are represented by italic characters.*

Step	dev2010	tst2010	tst2011	tst2012
1a. Boosted MMI	16.7	14.5	12.3	13.9
1b. Subspace GMM	17.3	14.9	12.9	14.2
2. System combination	16.4	13.8	12.0	13.3
3. Factored RNNLM	15.3	13.1	10.9	12.1
4. Topic adaptation	*15.0*	*12.8*	**10.6**	**12.0**
4a. Post-processing	14.8	12.6	*10.9*	*12.1*
4b. Our decoder	—	—	**10.6**	**12.0**

domain corpus with Eq. (1) and uses it together with the in-domain corpus for training. Finally, the training data of factored RNNLM contains 1,127K sentences with 30M words.

3.5. Topic adaptation

The TED talks in the IWSLT test sets touch on various topics without adhering to a single genre. To model each test set better, we utilize first-pass recognition hypothesis for topic adaptation of n-gram LMs. A problem here is that recognition hypothesis includes errors that limits the adaptation performance. To avoid negative impact of the errors in the first-pass result, we propose a similar metric to Eq. (1), which takes into account the recognition hypothesis and randomly selected sentences of out-of-domain corpus. Our adaption can be expressed as,

$$H_{ASR}(s) - H_O(s). \tag{2}$$

For each test set, we rank sentences of the out-of-domain according to Eq. (2), select $1/8$ of sentences with the lowest scores, build an adapted n-gram LM based on the selected sentences, interpolate the adapted LM with the in-domain LM by optimizing the perplexity of the development set. Here, the lexicon is extended to include new words appearing more than 10 times in the selected sentences.

4. Decoding system

4.1. Decoding system

The procedure of our ASR system depicted in Figure 1 is divided into four steps as follows:

1. Decode input speech using two sets of models,
2. Combine lattices output from the decoders,
3. Rescore n-best with factored RNNLM,
4. Adapt LMs and run through the steps above again.

First, we use WFST-based decoder to create lattice for input speech. In the submitted system, we employ decoder of the Kaldi toolkit for 3-gram decoding and 4-gram lattice rescoring. Here, two types of AMs described in Section 2.2, (a) SAT-bMMI and (b) SAT-SGMM, are employed individually, with n-gram LMs described in Section 3.3. This step produce two lattices l_a and l_b corresponding to the two AMs.

Then, the two lattices are combined using WFST compose operation as follows:

$$l_c = \mathrm{compose}(\mathrm{scale}(w, l_a),\ \mathrm{scale}(1-w, l_b)), \tag{3}$$

where scale is used to scale transition costs of WFST with the given weight w (set to 0.5) and $\mathrm{compose}$ is an operation to compute the composition of the two input WFSTs. When the resulting lattice l_c is empty, l_a is output instead of it. Note that the $\mathrm{project}$ operation is applied to l_b before the $\mathrm{compose}$ to map its output symbols on transitions to input side.

In the third step, factored RNNLM based rescoring is applied to n-best list extracted from the lattice l_c (n=100). The LM score of input i-th sentence s_i in the n-best is calculated as an interpolation of two kinds of LMs,

$$P(s_i) = \gamma \times P_{fRNN}(s_i) + (1-\gamma) \times P_{4g}(s_i), \tag{4}$$

where γ is a weighting factor (set to 0.5), $P_{fRNN}()$ and $P_{4g}()$ stand for scores based on factored RNN and 4-gram LMs, respectively. Then the 1-best sentence is obtained from the n-best scored by Eq. (4).

In the final step, n-gram LMs and lexicon are adapted to each test set, using the topic adaptation technique described in Section 3.5. Using 1-best results of the previous step, training data is newly selected from out-of-domain corpora with Eq. (2). Then the system run through the steps 1 to 3 again as a second pass decoding with the adapted LMs. Note that the AMs and factored RNNLM are not updated here.

4.2. Evaluation Results

Table 2 shows performance of our ASR system on transcribing the development sets, dev2010 and tst2010, and the test sets, tst2011 and tst2012. Word error rates (WERs) were decreased by combining two lattices derived from different types of AMs (Step 2). With respect to LMs, rescoring using factored RNNLM significantly contributed to achieve better performance (Step 3) and topic adaptation based on dynamic data selection also showed improvement (Step 4). These results would appear that each of technique employed in our system has a particular ability to improve ASR performance, although there are some exceptional cases in talk-level as shown in Figure 3.

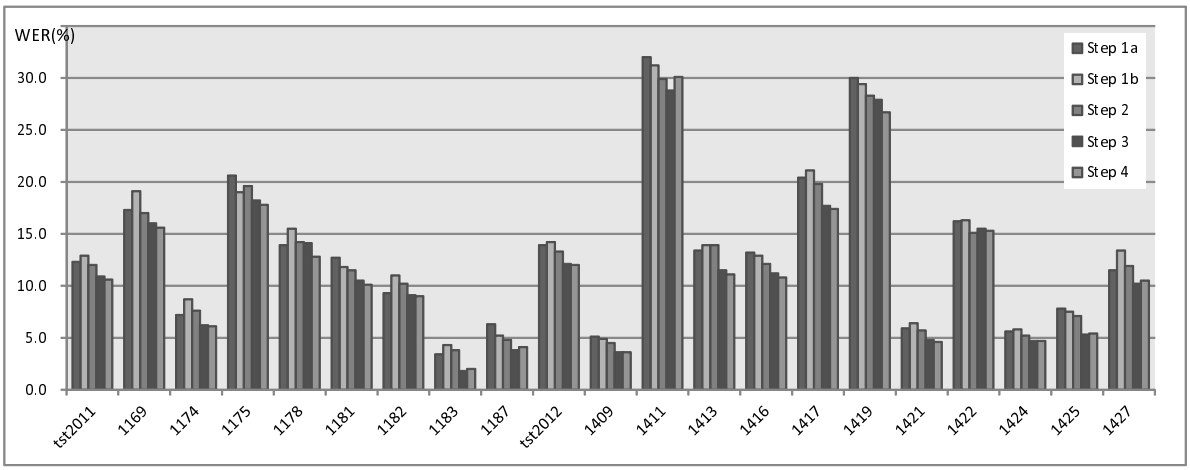


Figure 3: *Talk-level WERs of the tst2011 and tst2012.*

Note that the ASR results of the Step 4 are post-processed in our test submission (Step 4a). This step shrinks repetitions of one word or two words in word sequence. Though it helps to decrease WER of the development sets, it results in higher WER for the test sets.

Table 2 also shows the performance of our system when it utilizes our own WFST-based decoder (a variant of [11]) which can compose LMs on-the-fly during decoding time (Step 4b). The decoding process in Step 1 runs on-the-fly 4-gram decoding instead of the 4-gram rescoring after the 3-gram decoding, and also allowed for a more efficient graph building scheme. It achieved a reduction in computing time and memory usage when composing the WFSTs and running the decoder. Compared to the submitted system, it used 3% time and 26% memory in composing and 48% time and 46% memory in decoding.

5. Summary

In this paper, we describe our ASR system for the IWSLT 2012 evaluation campaign. The WFST-based system including system combination in terms of state-of-the-art AMs, factored RNNLM based rescoring, and unsupervised topic adaptation with dynamic data selection indicated an improvement in WER on transcribing the TED talks.

6. Acknowledgements

The authors would like to thank Mr. K. Abe for discussions on developing the ASR system and Dr. J. R. Novak for providing G2P toolkit.

7. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul and S. Stüker, "Overview of the IWSLT 2012 Evaluation Campaign," in *Proc. of IWSLT*, 2012.

[2] A. Katsamanis, *et al.*, "SailAlign: Robust long speech-text alignment," in *Proc. of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, 2011.

[3] D. Povey, *et al.*, "The Kaldi Speech Recognition Toolkit," in *Proc. of Workshop on Automatic Speech Recognition and Understanding*, 2011.

[4] R. Sproat, *et al.*, "Normalization of non-standard words," *Computer Speech and Language*, Vol. 15, pp. 287–333, 2001.

[5] J. R. Novak, *et al.*, "Improving WFST-based G2P Conversion with Alignment Constraints and RNNLM N-best Rescoring," in *Proc. of Interspeech*, 2012.

[6] R. Moore and W. Levis, "Intelligent selection of language model training data," in *Proc. of ACL*, 2010.

[7] A. Stolcke, *et al.*, "SRILM at Sixteen: Update and Outlook," in *Proc. of Workshop on Automatic Speech Recognition and Understanding*, 2011.

[8] T. Mikolov, *et al.*, "Recurrent neural network based language model," in *Proc. of Interspeech*, 2010.

[9] Y. Wu, *et al.*, "Factored Language Model based on Recurrent Neural Network," in *Proc. of COLING*, 2012.

[10] Y. Wu, *et al.*, "Factored Recurrent Neural Network Language Model in TED Lecture Transcription," in *Proc. of IWSLT*, 2012.

[11] P. R. Dixon, *et al.*, "A Comparison of Dynamic WFST Decoding Approaches," in *Proc. of ICASSP*, 2012.

The KIT Translation Systems for IWSLT 2012

Mohammed Mediani[⋆]*, Yuqi Zhang*[⋆]*, Thanh-Le Ha*[⋆]*, Jan Niehues*[⋆]*, Eunah Cho*[⋆]*, Teresa Herrmann*[⋆]*, Rainer Kärgel*[†] *and Alexander Waibel*[⋆]

Institute of Anthropomatics
KIT - Karlsruhe Institute of Technology

[⋆] firstname.lastname@kit.edu
[†] rainer.kaergel@student.kit.edu

Abstract

In this paper, we present the KIT systems participating in the English-French TED Translation tasks in the framework of the IWSLT 2012 machine translation evaluation. We also present several additional experiments on the English-German, English-Chinese and English-Arabic translation pairs.

Our system is a phrase-based statistical machine translation system, extended with many additional models which were proven to enhance the translation quality. For instance, it uses the part-of-speech (POS)-based reordering, translation and language model adaptation, bilingual language model, word-cluster language model, discriminative word lexica (DWL), and continuous space language model.

In addition to this, the system incorporates special steps in the preprocessing and in the post-processing step. In the preprocessing the noisy corpora are filtered by removing the noisy sentence pairs, whereas in the postprocessing the agreement between a noun and its surrounding words in the French translation is corrected based on POS tags with morphological information.

Our system deals with speech transcription input by removing case information and punctuation except periods from the text translation model.

1. Introduction

In the IWSLT 2012 Evaluation campaign [1], we participated in the tasks for text and speech translation for the English-French language pair. The TED tasks consist of automatic translation of both the manual transcripts and transcripts generated by automatic speech recognizers for talks held at the TED conferences [1].

The TED talks are given in English in a large number of different domains. Some of these talks are manually transcribed and translated by volunteers over the globe [2]. Given these manual transcripts and a large amount of out-of-domain data (mainly news), our ambition is to perform optimal translation on the untranslated lectures which are more likely from different domains. Furthermore, we strive for performing as well as possible on the automatically transcribed lectures.

The contribution of this work is twofold: on the one hand, it demonstrates how the complementary manipulation of in-domain and out-of-domain data is gainful in building more accurate translation models. It will be shown that while the large amount of out-of-domain data ensures wider coverage, the limited in-domain data indeed helps to model better the style and the genre. On the other hand, we show that using a text translation system with a proper processing of punctuation can handle the translation of automatic transcriptions to some extent.

Compared to our last year's system, three new components are introduced: adaptation of the candidate selection in the translation model (Section 5), continuous space language model (Section 8), and part-of-speech (POS)-based agreement correction (Section 9).

The next section briefly describes our baseline, while Sections 3 through 9 present the different components and extentions used by our phrase-based translation system. These include the special preprocessing of the spoken language translation (SLT) system, POS-based reordering, translation and language model adaptation, the cluster language model, the descriminative word lexica (DWL), the continuous space language model, and the POS-based agreement correction. After that, the results of the different experiments (official and additional language pair systems) are presented and finally a conclusion ends the paper.

2. Baseline System

For the corresponding tasks, the provided parallel data consist of the EPPS, NC, UN, TED and Giga corpora, whereas the monolingual data consist of the monolingual version of the News Commentary and the News Shuffled corpora. In addition, the use of the Google Books Ngrams[2] was allowed. We did not use the UN data and Google Books Ngrams this year. The reason was that in several previous experiments (not reported in this paper), they consistently had a negative impact on the performance.

[1] http://www.ted.com

[2] http://ngrams.googlelabs.com/datasets

A common preprocessing is applied to the raw data before performing any model training. This includes removing long sentences and sentences with length difference exceeding a certain threshold. In addition, special symbols, dates and numbers are normalized. The first letter of every sentence is smart-cased. Furthermore, an SVM classifier was used to filter out the noisy sentences pairs in the Giga English-French corpus as described in [3].

The baseline system was trained on the EPPS, TED, and NC corpora. In addition to the French side of these corpora, we used the provided monolingual data and the French side of the parallel Giga corpus, for language model training. Systems were tuned and tested against the provided Dev 2010 and Test 2010 sets.

All language models used are 4-gram language models with modified Kneser-Ney smoothing, trained with the SRILM toolkit [4]. The word alignment of the parallel corpora was generated using the GIZA++ Toolkit [5] for both directions. Afterwards, the alignments were combined using the grow-diag-final-and heuristic. The phrases were extracted using the Moses toolkit [6] and then scored by our in-house parallel phrase scorer [7]. Phrase pair probabilities are computed using modified Kneser-Ney smoothing as in [8]. Word reordering is addressed using the POS-based reordering model and is described in detail in Section 4. The POS tags for the reordering model are obtained using the TreeTagger [9]. Tuning is performed using Minimum Error Rate Training (MERT) against the BLEU score as described in [10]. All translations are generated using our in-house phrase-based decoder [11].

3. Preprocessing for Speech Translation

The system translating automatic transcripts needs some special preprocessing on the data, since generally there is no or not reliable case information and punctuation in the automatically generated transcripts. We have tried two ways to deal with the difference on casing and punctuation between a machine translation (MT) system and a SLT system. In addition, we also optimize the system with different development data: simulated ASR output and original automatic speech recognition (ASR) output.

In order to make the system translate the automatically generated transcripts, the first method we have used is to lowercase the source side of the training corpora and remove the punctuation except periods from the source language. On these modified source sentences and untouched target sentences, all models are re-trained, including alignments, phrase tables, reordering rules, bilingual language model and DWL model. Therefore, we can avoid having to build a whole MT system for the SLT task. In order to simplify the procedure, we tried a second method where we directly modify the source phrases in the phrase tables. We lowercase the source phrases and remove the punctuation except periods from the source phrases. Though there could be duplicated phrase pairs with different scores in the phrase table due to this modification, during the decoding the phrase with the best scores will be selected according to the weights.

Two ways to optimize the system are possible. The first one is to use the manual transcripts but it requires lower casing and removal of punctuation marks. The other one is to use the ASR single-best output released by the SLT task. The advantage of optimizing with the manual transcripts is that the system will be adjusted with higher quality sentences. On the other side, optimization using ASR output makes the system more consistent with the evaluation test data. We have tested both methods in our experiments.

4. Word Reordering Model

Our word reordering model relies on POS tags as introduced by [12]. Rule extraction is based on two types of input: the Giza alignment of the parallel corpus and its corresponding POS tags generated by the TreeTagger for the source side.

For each sequence of POS tags, where a reordering between source and target sentences is detected, a rule is generated. Its head consists of sequential source tags and its body is the permuted POS tags of the head which match the order of the corresponding aligned target words. After that, the rules are scored according to their occurrence and pruned according to a given threshold.

In our system, the reordering is performed as a preprocessing step. Rules are applied to the test set and possible reorderings are encoded in a word lattice, where the edges are weighted according to the rule's probability.

Finally, the decoding is performed on the resulted word lattice. During decoding, the distance-based phrase reordering could also be applied additionally.

5. Adaptation

To achieve the best performance on the target domain, we performed adaptation for translation models as well as language models.

5.1. Translation Model Adaptation

In a phrase-based translation system, building the translation consists of two steps. First, we select a set of candidate translations from the phrase table (candidate selection). In our system, we normally take the top 10 translations for every source phrase according to initially predefined weights. In the second step, the best translation is built from these candidates using the scores from the translation model (phrase scoring) as well as other models.

In some of our systems we also adapted the first step, while the second step was adapted in all of our systems by using additional scores for the phrase table.

To adapt the translation model towards the target domain, first, a large translation model is trained on all the available data. Then, a separate in-domain model is trained on the in-domain data only, reusing the alignment from the large model. The alignment is trained on the large data, because it

seems to be more important for the alignment to be trained on bigger corpora than being based on only in-domain data.

When we do not adapt the candidate selection, the best translations from the general phrase table is used and only the scores from the in-domain phrase table are taken into account. In the other case, we take the union of the phrase pairs collected from both phrase tables. We will refer to this adaptation method as **CSUnion** in the description of the results.

The scores of the translation model are adapted to the target domain by combining the in-domain and out-of-domain scores in a log-linear combination. The adapted translation model uses the four scores (phrase-pair probabilities and lexical scores for both directions) from the general model as well as the two probabilities of both directions from the small in-domain model. If the phrase pair does not occur in the in-domain part, a default score is used instead of a relative frequency. In our case, we use the lowest probability that occurs in the phrase table.

5.2. Language Model Adaptation

For the language model, it is also important to perform an adaptation towards the target domain. There are several word sequences, which are quite uncommon in general, but may be used often in the target domain.

As it is done for the translation model, the adaptation of the language model is also achieved by a log-linear combination of different models. This also fits well into the global log-linear model used in the translation system. Therefore, we train a separate language model using only the in-domain data from the TED corpus. Then it is used as an additional language model during decoding. Optimal weights are set during tuning by MERT.

6. Cluster Language Model

In addition to the word-based language model, we also use a cluster language model in the log-linear combination. The motivation is to make use of larger context information, since there is less data sparsity when we substitute words by word classes.

First, we cluster the words in the corpus using the MKCLS algorithm [13] given a number of classes. Second, we replace the words in the corpus by their cluster IDs. Finally, we train an n-gram language model on this corpus consisting of cluster IDs.

Because the TED corpus is small and important for this translation task and it exactly matches the target genre, we trained the cluster language model only on TED corpus in our experiments. The TED corpus is characterized by a huge variety of topics, but the style of the different talks of the corpus is quite similar. When translating a new talk from the same domain, we may not find a good translation in the TED corpus for many topic specific words. What TED corpus can help with, however, is to generate sentences in the same style. During decoding the cluster-based language model works as an additional model in the log-linear combination.

7. Discriminative Word Lexica

Mauser et al. [14] have shown that the use of DWL can improve the translation quality. For every target word, they trained a maximum entropy model to determine whether this target word should be in the translated sentence or not using one feature per one source word.

One specialty of this task is that we have a lot of parallel data we can train our models on, but only a quite small portion of these data, the TED corpus, is very important to the translation quality. Since building the classifiers on the whole corpus is quite time consuming, we try to train them on the TED corpus only.

When applying DWL in our experiments, we would like to have the same conditions for the training and test case. For this we would need to change the score of the feature only if a new word is added to the hypothesis. If a word is added the second time, we do not want to change the feature value. In order to keep track of this, additional bookkeeping would be required. Also the other models in our translation system will prevent us from using a word too often.

Therefore, we ignore this problem and can calculate the score for every phrase pair before starting with the translation. This leads to the following definition of the model:

$$p(e|f) = \prod_{j=1}^{J} p(e_j|f) \tag{1}$$

In this definition, $p(e_j|f)$ is calculated using a maximum likelihood classifier.

Each classifier is trained independently on the parallel training data. All sentence pairs where the target word e_j occurs in the target sentence are used as positive examples. We could now use all other sentences as negative examples. But in many of these sentences, we would anyway not generate the target word, since there is no phrase pair that translates any of the source words into the target word.

Therefore, we build a target vocabulary for every training sentence. This vocabulary consists of all target side words of phrase pairs matching a source phrase in the source part of the training sentence. Then we use all sentence pairs where e_j is in the target vocabulary but not in the target sentences as negative examples. This has shown to have a postive influence on the translation quality [3] and also reduces training time.

8. Continuous Space Language Model

In recent years, different approaches to integrate a continuous space models have shown significant improvements in the translation quality of machine translation systems, e.g. [15]. Since the long training time is the main disadvantage of this model, we only trained it on the small, but very domain-relevant TED corpus.

In contrast to most other approaches, we did not use a feed-forward neural network, but used a Restricted Bolzmann Machine (RBM). The main advantage of this approach is that we can calculate the free energy of the model, which is proportional to the language model probability, very fast. Therefore, we are able to use the RBM-based language model during decoding and not only in the rescoring phase. The model is described in detail in [16].

The RBM used for the language model consists of two layers, which are fully connected. In the input layer, for every word position there are as many nodes as words in the vocabulary. Since we used an 8-gram language model, there are 8 word positions in the input layer. These nodes are connected to the 32 hidden units in the hidden layer.

During decoding, we calculate the free energy of the RBM for a given n-gram. The product of this values is then used as an additional feature in the log-linear model of the decoder.

9. Postprocessing for Agreement Correction

The agreement in gender and number is one of the challenging problems encountered when translating from English into a morphologically richer language such as French. Consequently, a special postprocessing was designed in order to remedy the case where disagreements between nouns and related surrounding words exist. This post-processing is based on the POS tags generated by LIA tagger[3]. In order to improve the agreement features, several post-processing heuristics are applied on a sentence basis, which include the correction of the grammatical number and gender of adjective, article, possessive determiner, forms of *quelque* and past participles based on their corresponding nouns.

In order to minimize spurious assignments when finding instances of these parts of speech related to a specific noun, strict heuristics are used: Adjectives must appear straight before or after the noun. Articles, possessive determiners and forms of *quelque* have to directly precede nouns or have at most one adjective in between. Past participles must stand after (possibly reflexive) inflected forms of *être* that immediately follow nouns.

10. Results

In this section, we present a summary of our experiments for all tasks we have carried out for the IWSLT 2012 evaluation. It includes the official systems for the MT and SLT translation tasks and additional systems for other language pairs: English-German, English-Chinese and English-Arabic translations. All the reported scores are the case-sensitive BLEU, and calculated based on the provided Dev and Test sets.

[3] http://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/bechet/download_fred.html

10.1. MT Task

Table 1 summarises how our MT system evolved. The baseline translation model was trained on EPPS, TED, NC, and Giga corpora. This big model was adapted with a smaller one trained on TED data only as described in Section 5. The language model is a log-linear combination of three language models trained on different data sets: the French side of the EPPS, TED, and NC corpora, the provided monolingual news data (Monolingual EPPS, NC and News Shuffled), and a smaller in-domain language model trained on TED data. The reordering in this system was handeled as a preprocessing step using POS-based rules as described in Section 4. The result of this setting was 28.5 BLEU points on Dev and 31.73 on Test. The performance could be improved by around 0.4 on Dev and 0.2 on Test by using a bilingual language model (details about bilingual language model computation can be found in [17]). An additional 0.2 on both Dev and test could be gained by using a cluster language model where the clusters were trained on the in-domain TED data. After that, changing the adaptation strategy by the union selection discussed in Section 5 shows slight improvement of 0.1 on both Dev and Test. The effect of the DWL trained on only the TED corpus was rather dissimilar on Dev and Test. While it slightly improved the score on Dev (0.1) it has a much greater effect on Test (0.5). Further small improvement could be observed by using a continuous space language model: around 0.09 on both Dev and Test. Finally, by using the POS-based post-processing correction of the agreement on the target side the score on Test could be improved by an additional 0.06, resulting in 32.84 BLEU points on Test. We submitted the translations of Test2011 and Test2012 generated by this final system as primary; the translations generated by the second best system (same as the final but without agreement corrections) as contrastive.

System	Dev	Test
Baseline	28.50	31.73
+Bilingual LM	28.93	31.90
+Cluster LM	29.15	32.13
+CSUnion	29.27	32.21
+DWL	29.37	32.70
+RBM LM	29.46	32.78
+Agreement Correction	-	32.84

Table 1: Summary of experiments for the English-French MT task

10.2. SLT Task

The baseline system of the speech translation task used almost the same configuration as the one for the MT task, for which the POS-based reordering and the adaptation for both translation and language model with TED data were added to the baseline. The special processing we have done for SLT

task lie in the following aspects.

In order to simplify building the system, we did not retrain a new alignment for the SLT task, but modify the phrase tables from the MT task to make it suitable for the SLT task. Casing information and punctuation except periods has been removed from the source side of the phrase table. Then we feed this new phrase table with possibly duplicate phrase pairs into the SLT system and let the decoder select the best ones for a translation. For the purpose of comparison, we also rebuild a whole new SLT system, in which the alignment, the phrase table and all other models are newly generated with the training data without punctuation and casing information. However, the newly trained system is not better than the MT system with the modified phrase table. The experimental results are presented in Table 2. **large-retrain-PT** are with the newly trained phrase table on the same corpora. **large-modify-PT** is the system with the modified phrase table trained on bilingual corpora TED, NC, EPPS and Giga corpus. We can see that the completely retraining the system does not improve the result. It is very surprising that the retrained system hurts the result much. One possible explanation could be punctuations are very help to generate good alignments. In order to know the reasons more clearly, more experiments should be done in the future.

Another difference to the MT system is the the data used to build translation model does not include the Giga corpus. It includes only TED, NC and EPPS, since including the Giga corpus could not improve the translation results in the SLT task, as it does in the MT task. The intermediate experiments of comparing these two training data sets are shown in Table 2. **small-modify-PT** is the system trained only on TED, NC and EPPS. The systems trained on TED, NC, EPPS and Giga are called **large**.

System	Dev	Test(ASR)
large-retrain-PT	17.14	18.92
large-modify-PT	18.67	21.08
small-modify-PT	18.93	21.84

Table 2: Intermediate experiments with different phrase tables for the English-French SLT task

Our SLT system is optimized on the modified Dev text data by removing the punctuation except periods and lowercasing. And we have tested the system both on modified text test data which is with the same processing as the Dev text data and on the ASR output of the test data. Table 3 presents the results optimized on modified Text and ASR output, respectively. The two columns marked with **Test(ASR)** are comparable scores. There is no convinced evidence that on which condition the optimization is better. In the settings of "Baseline", "Adaptation" and "Bilingual LM" optimizing on ASR output gets better results. After applying all models, the system optimized on the modified text data wins about 0.5 BLEU points. Considering the final result after adding all models is better and the test data from modified Text if more reliable than the ASR output, we have chosen the system optimized on the modified text data as our primary system.

We present our system for the SLT task step by step in Table 3. The bilingual language model was trained on the EPPS corpus and all other available parallel data, whose punctuation marks on the source side are all removed. The cluster language model is trained on the TED corpus, where the words are classified into 50 classes. The DWL model is also trained on the TED corpus, but the punctuation and casing information have been removed from the source side of the training data.

Compared to the baseline the SLT system has improved about 1.1 BLEU on both text and ASR test data by adding all the models. The largest gain is about 0.5 by adding the cluster-based language model. The domain adaptation model has improved all scores on Dev, text Test and ASR Test. It especially improves the text Test by 0.5 BLEU. The bilingual language model does not seem to contribute much to the results, except a little improvement of 0.2 on the ASR test data. Then we add the DWL model which also improves the test data by about 0.2 BLEU points. Finally we have carried out the morphology agreement correction as described in Section 9, which improves around 0.1 on the test data.

This system was the system we used to translate the SLT evaluation set for our submission. We have submitted one primary system and three contrastive systems. The primary system is the translation of the ASR output *system1* with all models presented in Table 3. And the contrastive systems are the translations of the ASR outputs *system1 - system3* excluding the **Agreement Correction** model.

10.3. Additional Language Pairs

10.3.1. English-German

Several experiments were conducted for the English-German MT track on the TED corpus. They are summarized in Table 4. The baseline system is essentially a phrase-based translation system with some preprocessing steps on both source and target sides. Adapting huge parallel data from EPPS and NC to TED translation model helps us gain 0.71 BLEU scores on the test set. Short-range reordering based on POS information yields reasonable improvements on both development and test sets by about 0.5 BLEU points. In the language modeling aspect, different factors were experimented with, and 4-gram POS language model using RF-Tagger[4] slightly improves our system over the development set by 0.22 BLEU points but considerably shows its impact on test set with an improvement of 1 BLEU point. We approach our best system by adding a 9-gram cluster-based language model where the German side corpus is grouped into 50 classes, yielding 22.61 and 22.93 BLEU points on development and test sets, respectively.

[4] http://www.ims.uni-stuttgart.de/projekte/corplex/RFTagger/

	Optimization on Text			Optimization on ASR	
System	Dev (Text)	Test (Text)	Test (ASR)	Dev (ASR)	Test (ASR)
Baseline	25.37	27.57	21.68	19.11	21.86
+ Adaptation	25.64	28.08	21.90	19.31	22.04
+ Bilingual LM	25.07	28.08	22.07	19.14	22.28
+ Cluster LM	25.17	28.79	22.57	19.32	22.40
+ DWL	25.06	28.84	22.79	19.34	22.23
+ Agreement Correction	-	-	22.86	-	-

Table 3: Summary of experiments for the En-Fr SLT task

System	Dev	Test
Baseline	20.59	20.50
+ Adaptation	21.39	21.21
+ Reordering	21.97	21.74
+ POS LM	22.19	22.73
+ Cluster LM	22.61	22.93

Table 4: Experiments for the English-German on TED task

In this English-German translation system, we have also tried some other models such as using DWL, long-range reordering, bilingual language model as well as external monolingual language models but we do not gain noticeable improvements. Moreover, some experiments on tree-based reordering, which we believe helpful in this language pair, has been reserved for further considerations due to the limited time.

10.3.2. English-Chinese

With the bilingual data released by the TED Task of IWSLT 2012 we have developed an English-Chinese translation system. As it is an initial system for this new translation direction, we have made the main effort on data processing and preprocessing.

There are three corpora that could be used: the TED bilingual sentence-aligned corpus, the UN bilingual document-aligned corpus and the monolingual Google Ngrams corpus. In our system we have used the TED corpus to train the translation model and trained a language model on TED, UN and Google Ngrams. In addition we classify the Google Ngram corpus with its year information, such as google1980 is the ngrams from 1980-1989, and train a language model separately on each class. Our experience has shown that google1980 has contributed the most to the improvement, even more than the whole Google Ngram corpus.

In constrast to European languages, there are no spaces between Chinese words. Therefore, in the preprocessing of English-Chinese translation we need to decide on whether to segment Chinese into words, or to segment it into characters. We have tried both in our experiments. For the Chinese word segmentation we have made use of the Stanford Chinese word segmenter[5]. For the Chinese character segmentation we have simply inserted a space between neighbor Chinese characters. Then we have trained two systems: one based on Chinese words, the other based on Chinese characters. Table 5 shows the results from the two systems. Since the evaluation scores on Chinese words (**Test(Word)**) and on Chinese character (**Test(Cha.)**) are not comparable to each other, we segment the translation hypothesis on words into Chinese characters. Then the scores at the two columns **Test(Cha.)** are comparable. We can see that the system trained on characters is usually better than the system on words.

In Table 5 we present the steps which achieve improvement. The baseline system is trained only on the TED corpus (both for translation model and language model). By adding all possible language models and a reordering model, the BLEU score on test data has gained 0.2 points in total. Most improvements come from the larger language model. It seems that the current reordering model does not work quite well for the English-Chinese translation. Further analysis and work need to be done on the reordering model.

System	on characters		on words	
	Dev (Cha.)	Test (Cha.)	Test (Cha.)	Test (Word)
Baseline(4gram LM)	14.37	17.26	16.69	9.92
8gram LM	14.48	17.28	17.08	10.03
+ 4gram UN LM	14.61	17.38	16.80	9.99
+ POS Reordering	14.69	17.28	17.32	10.23
+ 5gram google1980	14.73	17.47	16.82	9.84

Table 5: Translation results for English-Chinese

The other models that we have tried, but have not given improvement to the system, include sentence-aligned extraction from the UN corpus and long-range reordering as described in [18].

[5] http://nlp.stanford.edu/software/segmenter.shtml

10.3.3. English-Arabic

The parallel data provided for this direction was from TED and UN. As for the English-Chinese direction (presented in Section 10.3.2), greater effort was devoted to the data preprocessing. The preprocessing for the English side is identical to the one used in the English-French system of the MT Task. Some of these preprocessing operations, such as long pair removal, were also applied to the Arabic side. In addition to that, the Arabic side was further orthographically transliterated using Buckwalter transliteration [19]. Tokenization and POS tagging were performed by the AMIRA toolkit [20]. The resulting translation is converted back to Arabic scripting before evaluation.

Table 6 presents some initial experiments for the English-Arabic pair. The baseline system uses only TED data for translation and language modeling. This gave a score of 13.12 on Dev and 8.05 on Test. This system was remarkably enhanced by introducing the short range reordering rules. The scores were improved by about 0.3 on Dev and 0.2 on Test. Adding monolingual data from the UN corpus had a great impact on the score on Dev (improved by 0.6), whereas it has a much lower effect on Test (improves by 0.1 only). In this last setting, three language models were log-linearly combined: one trained on TED data, one trained on UN data, and another one trained on both. Since the UN corpus was provided as raw data (no sentence alignment was performed before), we selected a sub-corpus of documents consisting of exactly the same number of sentences. This resulted in around 500K additional parallel sentences. The line **SubUN parallel** in Table 6 shows that these data had almost no effect on the system's performance. It increased the score on Dev by 0.02 and by 0.07 on Test. However, using the first translation model (trained on TED only) as indomain model to adapt the last setting shows slightly better improvements (around 0.1 on Dev and Test). Using a bilingual language model rather harmed the system on Dev by around -0.1 but improved the score on Test by 0.06. We choose to include this model because combined with the cluster language model it could improve our system by around 0.2 on Dev and Test wheras none of these models alone could outperform this score (some of these experiments are not reported here).

System	Dev	Test
Baseline	13.12	8.05
+ POS Reordering	13.46	8.23
+ Language models	14.08	8.32
+ SubUN parallel	14.10	8.39
+ TM Adapt	14.24	8.46
+ Bilingual LM	14.15	8.52
+ Cluster LM	14.28	8.63

Table 6: Experiments for the English-Arabic

11. Conclusions

In this paper, we presented the systems with which we participated in the TED tasks in both speech translation and text translation from English into French in the IWSLT 2012 Evaluation campaign. Our phrase-based machine translation system was extended with different models.

For the official language pair, even though we were authorized to use the UN parallel corpus and the monolingual Google Books Ngrams, these data had always a negative impact on our system's quality. More experiments should be carried out to extract some useful parts of these large data.

The successful application of different supplementary models trained exclusively on TED data (cluster language model, DWL, and continuous space language model) shows the usefulness and importance of in-domain data for such tasks, regardless of their small size.

The large amount of data used to train the different models integrated in our statistical system could not compensate for the ambiguity of translating into a morphologically richer language. Therefore, applying very simple and limited heuristics based on the target language grammar gave small but consistent improvments using the POS-based agreement correction.

We also presented experiments with several additional pairs. Namely, from English into one of the languages German, Chinese, or Arabic.

The use of additional bilingual corpora on adapting translation models as well as more complicated features from different language models led to expected performance in the English-German translation system. The effects of other techniques, e.g. long-range reordering or discriminative word alignment (DWA), were less obvious, mainly coming from the characteristics of the TED data.

In case of English-Chinese, we have found that the system based on Chinese characters works better than the system based on Chinese words. The BLEU score calculated on Chinese characters and Chinese words are also different: the BLEU score on character is about 17 while evaluation on the words the score is around 10. In addition we found that the current reordering model does not help much on this language pair. Further work needs to be done in this field in the future.

Due to the limited amount of data, the English-Arabic system performed relatively poorly. Furthermore, it showed eventual discrepency between Dev and Test data. Here again, as mentioned before, the UN data were not helpful.

12. Acknowledgements

This work was partly achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

13. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 Evaluation Campaign," in *Proc. of the International Workshop on Spoken Language Translation*, Hong Kong, HK, December 2012.

[2] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[3] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The KIT English-French Translation systems for IWSLT 2011," in *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT)*, 2011.

[4] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *International Conference on Spoken Language Processing*, Denver, Colorado, USA, 2002.

[5] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[6] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of ACL 2007, Demonstration Session*, Prague, Czech Republic, 2007.

[7] M. Mediani, J. Niehues, and A. Waibel, "Parallel Phrase Scoring for Extra-large Corpora," in *The Prague Bulletin of Mathematical Linguistics*, no. 98, 2012, pp. 87–98.

[8] G. F. Foster, R. Kuhn, and H. Johnson, "Phrasetable smoothing for statistical machine translation," in *EMNLP*, 2006, pp. 53–61.

[9] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *International Conference on New Methods in Language Processing*, Manchester, United Kingdom, 1994.

[10] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *Workshop on Data-drive Machine Translation and Beyond (WPT-05)*, Ann Arbor, Michigan, USA, 2005.

[11] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *International Conference on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[12] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Skövde, Sweden, 2007.

[13] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes." in *EACL'99*, 1999.

[14] A. Mauser, S. Hasan, and H. Ney, "Extending Statistical Machine Translation with Discriminative and Trigger-based Lexicon Models," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, ser. EMNLP '09, Singapore, 2009.

[15] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous Space Translation Models with Neural Networks," in *Proceedings of the 2012 Conference of the NAACL-HLT*, Montréal, Canada, June 2012.

[16] J. Niehues and A. Waibel, "Continuous Space Language Models using Restricted Boltzmann Machines," in *Proceedings of the nineth International Workshop on Spoken Language Translation (IWSLT)*, 2012.

[17] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, Edinburgh, UK, 2011.

[18] J. Niehues and M. Kolss, "A POS-Based Model for Long-Range Reorderings in SMT," in *Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece, 2009.

[19] N. Habash and F. Sadat, "Arabic Preprocessing Schemes for Statistical Machine Translation," in *Proceedings of the NAACL-HLT*, ser. NAACL-Short '06, Stroudsburg, PA, USA, 2006.

[20] M. Diab, "Second Generation Tools (AMIRA 2.0): Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking," in *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April 2009.

The UEDIN Systems for the IWSLT 2012 Evaluation

Eva Hasler, Peter Bell, Arnab Ghoshal, Barry Haddow, Philipp Koehn,
Fergus McInnes, Steve Renals, Pawel Swietojanski

School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, UK
{e.hasler,peter.bell,fergus.mcinnes,s.renals}@ed.ac.uk,
{aghoshal,pkoehn,bhaddow}@inf.ed.ac.uk,p.swietojanski@sms.ed.ac.uk

Abstract

This paper describes the University of Edinburgh (UEDIN) systems for the IWSLT 2012 Evaluation. We participated in the ASR (English), MT (English-French, German-English) and SLT (English-French) tracks.

1. Introduction

We report on experiments carried out for the development of automatic speech recognition (ASR), machine translation (MT) and spoken language translation (SLT) systems on the datasets of the International Workshop on Spoken Language Translation (IWSLT) 2012. Details about the evaluation campaign and the different evaluation tracks can be found in [1].

For the ASR track, we focused on the use of adaptive tandem features derived from deep neural networks, trained on both in-domain data from TED talks [2], and out-of-domain data from a corpus of meetings.

Our experiments for the MT track compare approaches to data filtering and phrase table adaptation and focus on adaptation by adding sparse lexicalised features. We explore different tuning setups on in-domain and mixed-domain systems.

For the SLT track, we carried out experiments with a punctuation insertion system as an intermediate step between speech recognition and machine translation, focussing on pre- and post-processing steps and comparing different tuning sets.

2. Automatic Speech Recognition (ASR)

In this section we describe the 2012 UEDIN system for the TED English transcription task. In summary, the system is an HMM-GMM system trained on TED talks available online, using tandem features derived from deep neural networks (DNNs). We were able to obtain benefits by including out-of-domain neural network features trained on a corpus of multi-party meetings. For recognition, a two-pass decoding architecture was used.

2.1. Acoustic modelling

Our core acoustic model training set was derived from 813 TED talks dating prior to the end of 2010. The recordings were automatically segmented, giving a total of 153 hours of speech. Each segment was matched to a portion of the manual transcriptions for the relevant talk using a lightly supervised technique described in [3]. For this purpose, we used existing acoustic models trained on multiparty meetings.

Three-state left-to-right HMMs were trained on features derived from the aligned TED data using a flat start initialisation. During the training process, a further re-alignment of the training segments and transcriptions was carried out, following which around 143 hours of speech remained for the final estimation of state-clustered cross-word triphone models. The resulting models contained approximately 3,000 tied states, with 16 Gaussians per state. Recognition was performed using HTK's HDecode. The first pass recognition transcription was used to estimate a set of CMLLR transforms [4] for each talk, using a regression class tree with 32 leaf-nodes, which were used to adapt the models for a second decoding pass.

The acoustic features used in the baseline system were 13-dimensional PLP features with first, second and third order differential coefficients, projected to 39 dimensions using an HLDA transform. To obtain acoustic features for the final system, we carried out experiments on the use of acoustic features derived from neural networks in the tandem framework [5]. Following our successful experience in [6], we investigated the use of features derived from networks trained on out-of-domain data using the Multi-layer Adaptive Networks (MLAN) architecture. In MLAN, tandem features are generated from in-domain data using neural network weights trained on out-of-domain data, and concatenated with in-domain PLP features and derivatives. A second, adaptive neural network is trained on these features. The final MLAN features used for HMM training and as input to the recogniser are obtained by concatenating posteriors from this second network with the original PLPs, projected with an HLDA transform. Figure 1 contrasts the MLAN process with the more standard use of out-of-domain posterior features. The procedure is described in more detail in [6].

In the experiments presented here, HMMs were trained

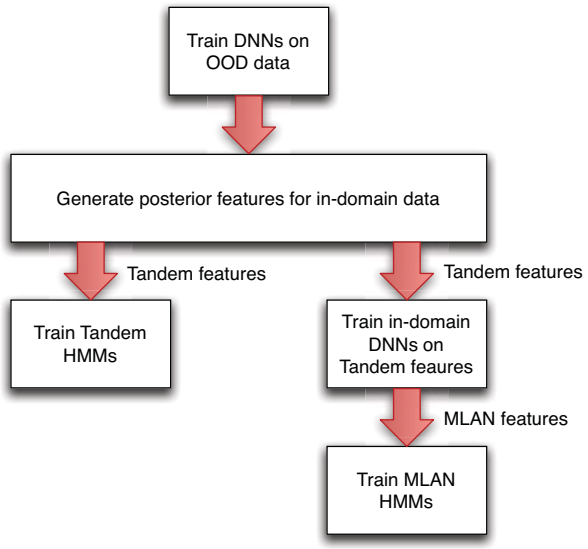


Figure 1: Multi-Level Adaptive Network (MLAN) architecture

on three sets of features:

- *In-domain* tandem features derived from four-layer deep neural networks (DNN) trained on the TED PLP features using monophone targets fixed by forced alignment with the baseline PLP models
- *Out-of-domain* features generated from Stacked Bottleneck networks trained on 120 hours of multi-party meetings from the AMI corpus using the setup described in [7]. Note that in general this domain is not well-matched to the TED domain[1]
- MLAN features obtained from four-layer DNNs trained on the AMI neural network features, concatenated with in-domain PLP features, again using monophone targets

The HMMs were trained using the tandem framework: the various neural network features were projected to 30 dimensions[2] and augmented with in-domain PLP features, projected from 52 to 39 dimensions with an HLDA transform, giving a total feature vector dimension of 69 in all three cases.

In the initial experiments, the HMMs were trained with maximum-likelihood training only. For the final system, we additionally employed speaker-adaptive training (SAT) [4] and MPE discriminative training [8]. When adaptation transforms were applied to the tandem features, the neural network and PLP features were adapted independently, using block diagonal (39x39 and 30x30) transforms.

[1]Standard HMMs trained on the AMI corpus, adapted using CMLLR to the test data, gave WER of 32.0% and 30.7% on the dev2010 and tst2010 sets respectively

[2]Except for the AMI bottleneck features, which were obtained from a 30-dimensional bottleneck with no further projection

Corpus	Word count
IWSLT12.TALK.train.en (in-domain)	2.4M
Europarl v7	54M
News commentary v7	4.4M
News crawl 2007	24.4M
News crawl 2008	23.1M
News crawl 2009	23.4M
News crawl 2010	23.9M
News crawl 2011	47.3M
Total	202.9M

Table 1: LM training data sizes.

2.2. Language modelling

The language models used for the ASR evaluation were obtained by interpolating individual modified Kneser-Ney discounted LMs trained on the small in-domain corpus of TED transcripts and the larger out-of-domain sources. The out-of-domain sources were europarl (v7), news commentary (v7) and news crawl data from 2007 to 2011. A random 1M sentence subset of each of news crawl 2007-2010 was used, instead of the entire available data, for quicker processing. The size of the resulting LM training data is shown in Table 1. The LMs were estimated using the SRILM toolkit [9]. The interpolated LMs had a perplexity of 160 (for 3-gram) and 159 (for 4-gram) on the combined dev2010 and tst2010 data. The optimal interpolation weights for both the 3-gram and 4-gram LMs were roughly 0.64 for the in-domain LM and between 0.02 and 0.06 for the different out-of-domain models. The vocabulary was fixed at 60,000 words.

We also carried out experiments using a language model built for the 2009 NIST Rich Transcription evaluation (RT09). This model was trained on a range of data sources, including corpora of conversational speech and meetings – see [7] for details. The vocabulary for this model was fixed at 50,000.

2.3. Results

We firstly carried out experiments on the dev2010 and tst2010 development data sets, using the NIST scoring toolkit to measure word error rate (WER). Our system models the initials in acronyms such as U.S., U.K. etc as individual words – for internal consistency, the development results here do not apply the automatic contraction of initials, which would result in an approximate 0.3% drop in WER below the figures shown. (Our final evaluation system, however, does include this correction).

Table 2 shows results of a two-pass speaker-adaptive system using the LM built for the IWSLT evaluation. All figures use a trigram LM except for the final row in the table. The results compare the use of different tandem features, and confirm our earlier findings that the MLAN technique is an effective method of domain adaptation, even when the domains are not particularly well matched. The use of SAT and

System	dev2010	tst2010
PLP + HLDA	26.7	24.9
TED tandem	21.3	20.3
AMI tandem	22.8	20.7
MLAN	20.5	18.7
+ SAT + MPE	18.5	16.4
+ 4gram LM	18.3	16.3

Table 2: Development set results (WER/%).

System	WER
MLAN	15.1
+ SAT + MPE	12.8
+ 4gram LM	12.4

Table 3: Results of MLAN systems on the *tst2011* test set

MPE training yields further improvements on the best feature set.

Somewhat unexpectedly, we found the RT09 LM to be more effective than the LM including in-domain data, with the best acoustic models achieving WER of 17.8% and 15.4% on dev2010 and tst2010 respectively. An interpolation of the two language models was found to yield even better performance, however, with WER of 17.1% and 14.7% respectively.

Finally, Table 3 shows results of selected acoustic models on the tst2011 test set, using our IWSLT language model. On the 2012 test data, the final system (MLAN + SAT + MPE + 4gram) achieved a WER of 14.4%.

3. Machine Translation (MT)

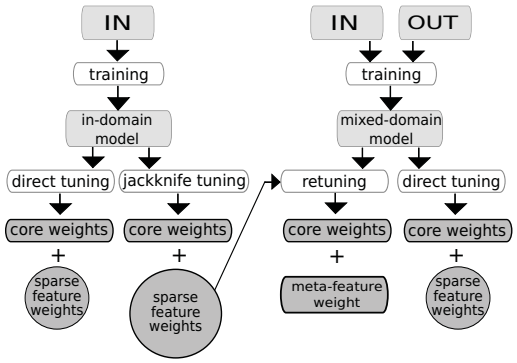
In this section we describe our machine translation systems for two language pairs of the MT track, English-French (en-fr) and German-English (de-en). We compare approaches to data filtering, phrase table adaptation and adaptation by adding sparse lexicalised features tuned on in-domain data, with different tuning setups.

3.1. Baseline SMT systems

Table 4 lists the available parallel and monolingual in-domain and out-of-domain training data. We built baseline systems with the Moses toolkit [10] on in-domain data (TED talks) as shown in tables 5 and 6 (labelled IN-PB and IN-HR) and further on in-domain data plus parallel out-of-domain data as shown in table 7 (labelled IN+OUT-PB). Parallel out-of-domain data consists of the Europarl, News Commentary and MultiUN corpora[3] for both language pairs and for en-fr also the French-English 10^9 corpus from WMT2012. The language models are 5-gram models with modified Kneser-Ney smoothing. Additional experiments were run with monolingual language model data from the Gigaword corpus (French Gigaword Second Edition, English Gigaword Fifth Edition) and News Crawl corpora from WMT2012, as marked in the results tables.

[3]For en-fr, this is the section from the year 2000 only, while for de-en it comprises the sections from 2000-2009.

Figure 2: *In-domain (IN) and mixed-domain (IN+OUT) models with three tuning schemes for tuning sparse feature weights: direct tuning, jackknife tuning and retuning.*



For the German-English systems we applied compound splitting [11] and syntactic pre-ordering [12] on the source side. As optimizers we used MERT as implemented in the current version of Moses and a modified version of the MIRA implementation in Moses as described in [13]. The language models were trained with the SRILM toolkit [9] and Kneser-Ney discounting. They were trained separately for each domain and subdomain (e.g. news data from different years) and linearly interpolated on the in-domain development set. Reported BLEU scores are case-insensitive and were computed using the mteval-v11b.pl script.

Hierarchical systems were only trained on in-domain data and lagged behind phrasebased performance by 0.7 BLEU for en-fr and 0.6 BLEU for de-en. Therefore, for all following systems we limited ourselves to phrasebased systems.

Table 4: *Word counts of in-domain and out-of-domain data.*

Parallel corpus	en-fr	de-en
TED (in-domain)	2.4M/2.5M	2.1M/2.2M
Europarl v7	50M/53M	45M/48M
News Commentary v7	3.0M/3.4M	3.5M/3.4M
MultiUN	316M/354M	5.5M/5.7M
10^9 corpus	576M/672M	n/a
Monolingual corpus	**fr**	**en**
TED (in-domain)	2.5M	2.4M
Europarl v7	55M	54M
News Commentary v7	4.2M	4.5M
News Crawl 2007-2011	512M	2.3G
Gigaword	820.6M	4.1G

3.2. Extensions

We experimented with several adaptation and tuning methods on top of our IN and IN+OUT baselines. One is the data selection method described in [14], using bilingual cross-entropy difference to select sentence pairs that are similar to the in-domain data and dissimilar to the out-of-domain data. We tried different filtering setups, selecting 10%, 20% and 50% of the parallel out-of-domain data. We also used the filtered target sides of the parallel data for building language models. Another approach is described in [15] (labelled *x+yE* there and *in+outE* here) and modifies the IN+OUT phrase tables by replacing all scores of phrase pairs found in the in-domain data by the values estimated on in-domain data only. The idea is to use the out-of-domain data only to provide additional phrases, i.e. to ignore counts from out-of-domain data whenever a phrase pair was seen in the in-domain data.

Table 5: *English-French in-domain (IN) systems trained with MERT (PB=phrasebased, HR=hierarchical), length ratio in brackets.*

System	test2010
IN-PB	**29.58 (0.966)**
IN-HR	28.94 (0.970)

Table 6: *German-English in-domain (IN) systems trained with MERT (PB=phrasebased, HR=hierarchical, PRE=preordering), length ratio in brackets.*

System	test2010
IN-PB (CS)	28.26 (0.999)
IN-PB (PRE)	28.04 (0.996)
IN-PB (CS + PRE)	**28.54 (0.995)**
IN-HR (CS + PRE)	27.88 (0.983)
IN-PB (CS + PRE)	
min=max=5	28.54 (0.995)
+ max=50	**28.57 (0.999)**
+ max=100	**28.60 (0.990)**
+ max=50, min=10	**28.65 (0.991)**

We tried several different approaches in order to specifically adapt the phrase pair choice to the style and vocabulary of TED talks. First, we added sparse word pair and phrase pair features on top of the in-domain translation systems and tuned them discriminatively with the MIRA algorithm. Word pair features are indicators of aligned pairs of a source and a target word, phrase pair features are indicators of a particular phrase pair used in a translation hypothesis and depend on the decoder segmentation of the source sentence. The values of these features in a translation hypothesis are counts of the number of times a word or phrase pair occurs in the current translation hypothesis. These sparse features are meant to capture preferred word and phrase choices in the in-domain data and therefore provide a bias for the translation model towards in-domain style and vocabulary. An example of a phrase pair feature is *pp_a,language∼une,langue=1*.

In the standard setup, sparse features were tuned on a small development set (dev2010), but we also used an alternative setup where they were tuned on the entire in-domain data, using 10 jackknife systems each trained on $\frac{9}{10}$ of the data and leaving out one fold for translation (the jackknife systems were run in parallel just like in normal parallelized discriminative tuning). We refer to the latter setup as *word pairs (JK)* and *phrase pairs (JK)*. For the systems built from in-domain and out-of-domain data (mixed-domain) we trained the sparse features on the development set as before. But since training with the jackknife setup would be rather time-consuming with the larger data sets, we reused the features trained on the in-domain data instead. In order to bring them on the right scale for the larger models, we ran a retuning step where jackknife-tuned features are treated as an additional component in the log-linear translation model. Running MERT on this extended model, we tuned a global meta-feature weight which is applied to all sparse features during decoding. Figure 2 gives an overview of all tuning setups involving sparse features on top of in-domain and mixed-domain models (direct tuning refers to sparse feature tuning on a development set). This is described in more detail in [13].

Table 7: *English-French and German-English mixed-domain (IN + OUT) systems trained with MERT, PB=phrasebased.*

System	test2010	
	en-fr	de-en
IN-PB	29.58	28.54
IN+OUT-PB	31.67	28.39
+ only in-domain LM	30.97	28.61
+ gigaword + newscrawl	31.96	30.26
IN-PB		
+ 10% OUT	32.30	**29.29**
+ 20% OUT	**32.45**	29.11
+ 50% OUT	32.32	28.68
best + gigaword + newscrawl	**32.93**	**31.06**
in+outE	32.19	29.59
+ only in-domain LM	30.89	29.36
+ gigaword + newscrawl	**32.72**	**31.30**

3.3. Results

In this section we compare results of the different data and tuning setups. Unless stated otherwise, the systems were tuned on the dev2010 set and evaluated on the test2010 set.

Table 5 shows the English-French systems and table 6 shows the German-English systems trained on in-domain (IN) data only. In both cases the phrase-based model outperformed the hierarchical model. For German-English, the best baseline system used both compound splitting and syntactic

Table 8: *German-English and English-French extensions of in-domain systems with sparse word pair and phrase pair features.*

System	test2010	
	en-fr	de-en
IN-PB, MERT	29.58	28.54
IN-PB, MIRA	30.28	28.31
+ word pairs	**30.36**	**28.45**
+ phrase pairs	**30.62**	**28.40**
+ word pairs (JK)	**30.80**	**28.78**
+ phrase pairs (JK)	**30.77**	**28.61**

pre-ordering. We tried different settings for the compound splitter, adjusting the minimum and maximum word counts. The min-counts avoids splitting into rare words, the max-count avoids splitting frequent words. The results indicate that changing the default values can yield a slight increase in performance.

Table 7 shows the mixed-domain systems (in-domain (IN) + out-of-domain data (OUT)) for both language pairs. The IN+OUT-PB baselines used the parallel data and the respective language model data. For en-fr, using additional out-of-domain data for the language model is better than using the in-domain LM alone (+0.7), but adding the newscrawl and gigaword data yields only a small further improvement (+0.3). For de-en, the IN+OUT-PB baseline is worse than the IN-PB baseline and improves when using only the in-domain LM. This indicates that the parallel OUT data is very dissimilar to the TED data for this language pair. However, adding newscrawl and gigaword data yields a larger improvement of 1.9 BLEU. The next block shows results of the data filtering approach and confirms the tendency from above. The de-en system profits from using only 10% of the OUT data (+0.9 BLEU) and adding more language model data yields an additional +1.8 BLEU. The en-fr system also benefits from using only part of the OUT data (+0.8 BLEU), in this case 20%, but only improves by 0.5 BLEU with additional LM data. The last block shows results of the *in+outE* approach, which uses the IN+OUT table but with scores from the IN table for all phrase pairs that were seen in the in-domain corpus. The results of this approach are comparable to the data selection method (a bit worse for en-fr and a bit better for de-en), but the advantage is that no data is thrown away and there is no need to tune a threshold for data selection.

Table 8 shows extensions of the in-domain systems for both language pairs. For en-fr, using MIRA to train the baseline system instead of MERT yields a gain of +0.7 BLEU and adding sparse word pair and phrase pair features adds a further 0.2 and 0.3 BLEU. We get the best performance by tuning the sparse features with the jackknife method, i.e. on all in-domain training data, yielding +1.2 over the MERT baseline. For de-en, the MIRA baseline is slightly worse than the MERT baseline, but adding sparse features on top of it has a similar positive effect. One thing to note is that the best weights during MIRA training were selected according to the test2010 set, so the results have to be considered optimistic when evaluating on test2010[4], while for evaluation on test2011 and test2012 we had distinct dev, devtest and test sets.

Table 9 shows combinations of the systems described in tables 7 and 8 for both language pairs. In the first block, we trained sparse features on a development set on top of the IN+OUT systems with data selection (10% for de-en and 20% for en-fr). In the second block, we applied a retuning step to integrate the sparse features trained on jackknife systems into the IN+OUT systems with data selection (see figure 2 for clarification). MERT results for test2010 are averaged over three runs, and the best of these three systems was used to translate test2011. For both language pairs we see improvements over the baselines with both methods of training sparse features (direct tuning and retuning) and we selected the best performing system on test2010 for submission (highlighted in grey). Evaluation on test2011 shows, however, that some of the contrastive systems (other systems from this table) perform better on this test set. The best performing systems on test2010 yield the following scores on test2011: for en-fr, 39.95 BLEU w/o additional LM data and 40.44 BLEU with additional newscrawl and gigaword data, and for de-en, 33.31 BLEU w/o additional LM data and 36.03 BLEU with additional gigaword and newscrawl data.

The systems used for our submissions did not include the additional monolingual data, which add an additional 0.5 BLEU for en-fr and 2.7 BLEU for de-en. As mentioned above, our en-fr system includes only one portion of the multiUN data (from the year 2000) instead of all data from years 2000-2009.

4. Spoken Language Translation (SLT)

Our SLT system takes the output of an ASR system, applies several transformational steps and then translates the output to French, using one of our English→French systems from section 3. We compare different preprocessing and tuning setups and show results on the outputs of four different ASR systems.

The transformations between ASR output and MT input are a pipeline consisting of three steps.

1. preprocessing of ASR output (number conversion)
2. punctuation insertion by translation from English w/o punctuation to English with punctuation
3. postprocessing (punctuation correction)

In the proprocessing step, we convert numbers that are represented in a systematically different way compared to the

[4]Though past experiments have suggested that choosing the weights on the development set instead does no make much difference.

Table 9: *German-English and English-French extensions of mixed-domain systems with sparse features. Grey cells mark systems used for submissions. Results of MERT-tuned systems for test2010 are averages over three runs of which the best was chosen for translating test2011.*

System	en-fr		de-en	
	test2010	test2011	test2010	test2011
IN-PB + 10%/20% OUT, MIRA	33.22	40.02	28.90	**34.03**
+ word pairs	**33.59**	39.95	28.93	33.88
+ phrase pairs	33.44	40.02	29.13	33.99
IN-PB + 10%/20% OUT, MERT	32.32	39.36	29.13	33.29
+ retune(word pairs JK)	32.90	**40.31**	**29.58**	33.31
+ retune(phrase pairs JK)	32.69	39.32	29.38	33.23
Submission system (grey) + gigaword + newscrawl	**33.98**	**40.44**	**31.28**	**36.03**

MT input data (details below). The punctuation insertion system is a standard MT translation system and is similar to the FullPunct-PPMT setup described in [16]. It was trained with the Moses toolkit [10] on 141M parallel sentences from the TED corpus, where the source side consists of transcribed speech and the target side consists of the source side of the parallel MT data. Source and target TED talks were first mapped according to talkids and then sentence-aligned. All speaker information was removed from the data.

Table 10 shows several variants of the punctuation insertion system. The evaluation metric is BLEU with respect to the MT source texts, because the punctuation insertion systems tries to 'translate' ASR outputs into MT inputs. Baseline1 refers to the training data of 141M parallel sentences, baseline2 used this data plus a duplicate of it where all but the sentence-final punctuation was removed. The idea was to avoid excessive insertion of punctuation by providing the system with both alternatives (the same phrases with and without punctuation), but this did not yield better results when combined with the original casing (w/o truecasing). To avoid introducing noise during decoding, we restricted the system to monotone decoding. Truecasing is usually useful to reduce data sparseness, but for punctuation insertion it turned out to be better to keep the original case information in order to avoid inserting sentence-initial punctuation. We also tried removing all quotes from the training data since predicting opening and closing quotes is more difficult than predicting other kinds of punctuation, but this did not yield improvements. In a first step we only converted year numbers with regular expressions, for example

- *nineteen thirty two* $\rightarrow$ *1932*
- *two thousand and nine* $\rightarrow$ *2009*
- *nineteen nineties* $\rightarrow$ *1990s*

Even though there is no strict convention of number representation in MT data, we also tried converting more types of numbers like

- *one hundred seventy four* $\rightarrow$ *174*
- *a hundred and twenty* $\rightarrow$ *120*
- *twenty sixth* $\rightarrow$ *26th*

which yielded some additional improvements. Postprocessing of punctuation insertions removes punctuation from the beginning of the sentence (where it is sometimes erroneously inserted), inserts final periods when there is no sentence-final punctuation and tries to make quotation marks more consistent (by removing single quotation marks or inserting additional ones).

Table 10: *Variants of punctuation insertion systems (evaluation set: test2010).*

Punctuation Insertion System	BLEU(MT source)
baseline 1	83.92
+ monotone decoding	84.01
+ w/o truecasing	84.49
+ w/o quotes	84.02
+ more number conversion	**84.80**
baseline 2	83.99
+ monotone decoding	84.04
+ w/o truecasing	83.76

We experimented with different tuning sets for the punctuation insertion system. The source side is one of devtest2010 ASR transcript, a concatenation of the dev2010 and test2010 ASR transcripts and a concatenation of the dev2010 and test2010 ASR outputs (all number-converted). The target side is the English side of the MT dev2010 set. Table 11 at the top shows the BLEU score with respect to the MT source of the raw ASR 2010 transcript and with number conversion. Next is the performance of the system that was tuned on dev2010 ASR transcripts. The number-converted ASR transcript improves by over 13 BLEU points when running it through the punctuation insertion system. As expected, there is a large gap between the quality of ASR

Table 12: *ASR outputs (English) → French. The punctuation insertion system used for test2010 was trained on ASR transcripts, the system used for test2011/test2012 on ASR outputs.*

SLT pipeline + MT System	BLEU(MT source)	BLEU(MT target)	Oracle
test2010 ASR transcript	85.17	30.54	33.98
test2010 ASR output UEDIN	61.82	22.89	33.98
test2011 ASR output system0	**67.40**	27.37	40.44
test2011 ASR output system1	65.73	**27.47**	40.44
test2011 ASR output system2	65.82	**27.48**	40.44
test2011 ASR output UEDIN	63.35	26.83	40.44
test2012 ASR output system0	70.73	n/a	n/a
test2012 ASR output system1	67.90	n/a	n/a
test2012 ASR output system2	66.82	n/a	n/a
test2012 ASR output UEDIN	63.74	n/a	n/a

Table 11: *Punctuation insertion + postprocessing with varying tuning and evaluation sets.*

Baselines w/o punctuation insertion	BLEU(MT source)
test2010 ASR transcript	70.79
+ number conversion	71.37
Punctuation Insertion System	**BLEU(MT source)**
Tune: dev2010 ASR transcript	
test2010 ASR transcript	84.80
+ postpr.	85.17
test2010 ASR output	61.65
+ postpr.	61.82
test2011 ASR output	62.04
+ postpr.	62.39
Tune: dev2010+tst2010 ASR transcripts	
test2011 ASR output + postpr.	**63.03**
Tune: dev2010+tst2010 ASR outputs	
test2011 ASR output + postpr.	**63.35**

transcripts vs. ASR outputs, but for all data sets the postprocessing step improves the quality. Thus, we can see that each step in the SLT pipeline improves the quality of the final output. The next two blocks show the quality of the test2011 system when the punctuation insertion system is tuned on a combination of the dev2010 and test2010 sets, both ASR transcripts and ASR outputs. Using more tuning data gains another 0.6 BLEU points and using real ASR outputs a further 0.3 BLEU improvement.

Table 12 shows the results of the complete SLT pipeline for test2010 and test2011 (the MT references for test2012 were not available at the time of writing). Before the translation step there is a large gap of more than 23 BLEU points between the ASR transcript and output, which mirrors the recognition errors. This results in a gap of more than 7 BLEU points after translation to French. The translation of the test2010 ASR transcript is 3.5 BLEU points below the translation of the real MT source set which is shown as the oracle (translation with perfect inputs). The MT sytem used for translation of the ASR output was the highlighted en-fr system from table 9, but here we are showing the results of translation systems with additional newscrawl and giga data (the difference was below 0.2 BLEU for the test2011 sets). Translating the test2010 set to English yields a BLEU score of 22.89. This could be improved by using ASR output of the dev2010 for tuning the punctuation system. For the test2011 set, there is gap of 4 BLEU points between the processed ASR outputs of the UEDIN system and the highest-ranking system (system0), measured against the MT source file. The BLEU score difference of the translations is only about 0.5 though, with system0 yielding a translation BLEU score of 27.37. Even though system0 yields the best BLEU score on the MT input file (67.40), system1 and system2 yield the best translation scores of the four systems, with 27.47 and 27.48 BLEU.

5. Conclusion

We presented our results for the ASR, MT and SLT tasks of the IWSLT 2012 Evaluation.

Our best ASR system for the TED task achieved scores of 12.4% on the 2011 test data set and 14.4% on the 2012 set. We found that the MLAN scheme for incorporating out-of-domain information using neural network features was effective in reducing WER compared to our standard tandem system.

Our largest MT systems yield BLEU scores of 40.44 for English-French and 36.03 for German-English on test2011. The data selection and phrase table adaptation methods showed comparable improvements over the mixed-domain baselines and we saw gains by adding sparse lexicalised features tuned on in-domain data. However, the relative results of our primary and constrastive systems varied quite a bit between the test2010 and test2011 data sets, so we cannot yet draw a final conclusion about an optimal setup.

Our SLT system yields BLEU scores between 26.83 and 27.48 on test2011, depending on the quality of the ASR outputs. Pre- and postprocessing of punctuation insertion turned out to be useful and we got slightly better results when tuning

the system on ASR outputs rather than ASR transcripts.

6. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. of the International Workshop on Spoken Language Translation*, Hong Kong, HK, December 2012.

[2] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[3] A. Stan, P. Bell, and S. King, "A grapheme-based method for automatic alignment of speech and text data," in *Proc. IEEE Workshop on Spoken Language Technology*, Miama, Florida, USA, Dec. 2012.

[4] M. Gales, "Maximum likelihood linear transforms for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 75-98, 1998.

[5] H. Hermanksy, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000, pp. 1635–1630.

[6] P. Bell, M. Gales, P. Lanchantin, X. Liu, Y. Long, S. Renals, P. Swietojanski, and P. Woodland, "Transcription of multi-genre media archives using out-of-domain data," in *Proc. IEEE Workshop on Spoken Language Technology*, Miama, Florida, USA, Dec. 2012.

[7] T. Hain, L. Burget, J. Dines, P. Garner, F. Grezl, A. Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "Transcribing meetings with the AMIDA systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 486–498, 2012.

[8] D. Povey and P. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP*, vol. I, 2002, pp. 105–108.

[9] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *Proc. ICSLP*, vol. 2, 2002, pp. 901–904.

[10] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *ACL 2007: proceedings of demo and poster sessions*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 177–180.

[11] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *In Proceedings of EACL*, 2003, pp. 187–193.

[12] M. Collins, P. Koehn, and I. Kučerová, "Clause restructuring for statistical machine translation," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ser. ACL '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 531–540.

[13] E. Hasler, B. Haddow, and P. Koehn, "Sparse lexicalised features and topic adaptation for SMT," in *Proceedings of the International Workshop on Spoken Language Translation*, Hong Kong, HK, December 2012.

[14] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 355–362.

[15] B. Haddow and P. Koehn, "Analysing the effect of Out-of-Domain data on SMT systems," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 422–432.

[16] J. Wuebker, M. Huck, S. Mansour, M. Freitag, M. Feng, S. Peitz, C. Schmidt, and H. Ney, "The RWTH Aachen machine translation system for IWSLT 2011," in *International Workshop on Spoken Language Translation*, San Francisco, California, USA, Dec. 2011, pp. 106–113.

The NAIST Machine Translation System for IWSLT2012

Graham Neubig, Kevin Duh, Masaya Ogushi, Takamoto Kano
Tetsuo Kiso, Sakriani Sakti, Tomoki Toda, Satoshi Nakamura

Graduate School of Information Science
Nara Institute of Science and Technology, Japan

Abstract

This paper describes the NAIST statistical machine translation system for the IWSLT2012 Evaluation Campaign. We participated in all TED Talk tasks, for a total of 11 language-pairs. For all tasks, we use the Moses phrase-based decoder and its experiment management system as a common base for building translation systems. The focus of our work is on performing a comprehensive comparison of a multitude of existing techniques for the TED task, exploring issues such as out-of-domain data filtering, minimum Bayes risk decoding, MERT vs. PRO tuning, word alignment combination, and morphology.

1. Introduction

This paper describes the NAIST participation in the IWSLT 2012 evaluation campaign [1]. We participated in all 11 TED tasks, dividing our efforts in half between the official English-French track and the 10 other unofficial Foreign-English tracks. For all tracks we used the Moses decoder [2] and its experiment management system to run a large number of experiments with different settings over many language pairs.

For the English-French system we experimented with a number of techniques, settling on a combination that provided significant accuracy improvements without introducing unnecessary complexity into the system. In the end, we chose a four-pronged approach consisting of using the web data with filtering to remove noisy sentences, phrase table smoothing, language model interpolation, and minimum Bayes risk decoding. This led to a score of 31.81 BLEU on the tst2010 data set, a significant increase over 29.75 BLEU of a comparable system without these improvements. In Section 2 we describe each of the methods in more detail and examine their contribution to the accuracy of the system. For reference purposes, in Section 3, we also present additional experiments that gave negative results, which were not included in our official submission.

For the 10 translation tasks into English, we focused on techniques that could be used widely across all languages. In particular, we experimented with unsupervised approaches to handling source-side morphology, minimum Bayes risk decoding, and large language models. In the end, most of our systems used a combination of unsupervised morphology processing and large language models, which resulted in an average gain of 1.18 BLEU points over all languages. Section 4 describes these results in further detail.

Decoding	dev2010	tst2010
Baseline	26.02	29.75
NAIST Submission	27.05	**31.81**

Table 1: The scores for systems with and without the proposed improvements.

2. English-French System

The NAIST English-French translation system for IWSLT 2012 was based on phrase-based statistical machine translation [3] using the Moses decoder [2] and its corresponding training regimen. Overall, we made four enhancements over the standard Moses setup to improve the translation accuracy:

Large-scale Data with Filtering: In order to use the large, but noisy parallel training data in the English-French Giga Corpus, we implemented a technique to filter out noisy translated text.

Phrase Table Smoothing: We performed phrase table smoothing to improve the probability estimates of low-frequency phrases.

Language Model Interpolation: In order to adapt to the domain of the task, we interpolated language models trained using text from several domains.

Minimum Bayes-Risk Decoding: We used lattice-based minimum Bayes risk decoding to select hypotheses that are supported by other hypotheses in the n-best list, and calibrated the probability distribution to further improve performance.

We demonstrate our results (in BLEU score) before and after these techniques are added in Table 1. It can be seen that the combination of these 4 improvements leads to a 2.06 point gain in BLEU score on tst2010 over the baseline system. We will explain each of the techniques in detail as follows.

Corpus	English	French
TED	2.36M	2.47M
News Commentary (NC)	2.99M	3.45M
EuroParl (EP)	50.3M	52.5M
United Nations (UN)	302M	338M
WMT2012 Giga	575M	672M
Giga (+Filtering)	485M	565M

Table 2: The number of words in each corpus.

Giga Data	dev2010	tst2010
None	26.61	31.52
Unfiltered	27.03	**31.90**
Filtered	27.05	31.81

Table 3: Accuracy given various styles of using the Giga data.

2.1. Data

The first step of building our system was preparing the data. Table 2 shows the size and genre of each of the corpora available for the task. From these corpora, we used TED, NC, EuroParl, UN, and Giga for training the language model, and TED, NC, EuroParl, and filtered Giga (explained below) for training the translation model.[1] Tuning was performed on dev2010, and testing was performed on tst2010.

In particular, the English-French Giga-word corpus is from the web and thus covers a wide variety of diverse topics, making it a strong ally for the construction of a general domain machine translation system. However, as the sentences were automatically extracted, they contain a significant number of errors where the content of the parallel sentences actually do not match, or only match partially. In order to filter out some of this noise, we re-implemented a variant of the sentence filtering method of [4].

The method works by using a clean corpus to train a classifier that can detect mis-aligned sentences. Because the clean corpus only contains correctly aligned sentences, we create pseudo-negative examples by traversing the corpus and randomly swapping two consecutive sentences with some set probability. These swapped sentences are labeled as "negative," and the remainder of the unswapped samples are labeled as positive.

In this application, the feature set chosen for the classifier must satisfy two desiderata. First, as with all machine learning applications, the features must be sufficient to discriminate between the classes that we are interested in: properly or improperly aligned sentences. Second, as our training data (a clean corpus) and testing data (a noisy corpus) will necessarily be drawn from different domains, we would like to use a small, highly generalizable feature set that will work on both domains. In order to achieve both of these objectives, we take hints from [4] and [5] to define the following features, where f_1^J and e_1^I are the source and target sentences, and J and I are their respective lengths:

Length Ratio features capture the fact that properly aligned sentences should be approximately the same length. Two continuous features $\max(J,I)/\min(J,I)$, J/I, and three indicator features $J > I$, $I > J$, $I = J$.

Model One Probability features capture the fact that an unsupervised alignment model (in this case, the efficiently calculable IBM Model One [6]) should assign higher probability to well-aligned sentences. In this category, we use two continuous features $\log P_{M1}(e_1^I|f_1^J)$ and $\log P_{M1}(f_1^J|e_1^I)$.

Alignment features use Viterbi word alignments and capture certain patterns that should occur in properly aligned sentences. Word alignments are calculated using IBM Model One, and symmetrized using the "intersection" criterion [7]. If the number of aligned words is K, our features include aligned word ratio $K/\min(I,J)$, total number of aligned words K, number of alignments that are monotonic, monotonic alignment ratio, and the average length of gaps between words (similar to "distortion" used in phrase-based MT [3]).

Same Word features count the number of times that a word of length n is exactly equal to a word in the opposite sentence. This is useful for noticing when proper names, numbers, or words with a shared linguistic origin occur in both sentences. In our system we use separate features for $n = 1$, $n = 2$, $n = 3$, and $n \geq 4$.

To train the non-parallel sentence identifier, we use data from the TED, NC, and EuroParl corpora swapping sentences with a probability of 0.3 to create pseudo-negative examples. We use this as training data for a support vector machine (SVM) classifier, which we train using LIBLINEAR [8]. In order to get an estimate of the accuracy of sentence filtering, we perform 8-fold cross validation on the training data, and achieve a classification accuracy of 98.0%.[2]

Next, we run the trained classifier on the entirety of the Giga corpus and remove the examples labeled as non-parallel. As a result of filtering with the classifier, a total of 485M English and 565M French words remained, a total of 84.3% of the original corpus.

Finally, using no Giga data, the unfiltered Giga data, and the filtered Giga data (in addition to all other data sets), we measured the final accuracy of the translation system. The

[1] We also attempted to use the UN corpus for training the translation model, but found that it provided no gain, likely because of the specialized writing style of UN documents.

[2] Of course, as we are using pseudo-negative examples in the Europarl corpus instead of real negative examples from the Giga corpus, these accuracy features are only approximate.

Smoothing	dev2010	tst2010
None	26.75	31.19
Good-Turing	27.05	**31.81**

Table 4: BLEU results using translation model smoothing.

LM	dev2010	tst2010
TED Only	24.80	29.44
Without Interp.	26.30	31.15
With Interp.	27.05	**31.81**

Table 5: Results training the language model on only TED data, and when other data is used without and with language model interpolation.

results are shown in Table 3. As a result, we can see that using the data from the Giga corpus has a positive effect on the results, but filtering does not have a clear significant effect on the results.

2.2. Phrase Table Smoothing

We also performed experiments that used smoothing of the statistics used in calculating translation model probabilities [9]. The motivation behind this method is that the statistics used to train the phrase table are generally sparse, and tend to over-estimate the probabilities of rare events. In the submitted system we used Good-Turing smoothing for the phrase table probabilities.

Results comparing a system with smoothing and without smoothing can be found in Figure 4. It can be seen that Good-Turing smoothing of the phrase table improves results by a significant amount.

2.3. Language Model Interpolation

One of the characteristics of the IWSLT TED task is that, as shown in Table 2, we have several heterogeneous corpora. In addition, the in-domain TED data is relatively small, so it can be expected that we will benefit from using data outside of the TED domain. In order to effectively utilize out-of-domain data in language modeling, we build one language model for each domain and interpolate the language models to minimize perplexity on the TED dev2010 set using the method described by [10] and implemented in the SRILM toolkit [11].

To measure the effectiveness of this technique, we also measure the accuracy without any data other than TED, and when the data from all domains was simply concatenated together for LM learning. The results can be found in Table 5. We can see that adding the larger non-TED data to the language model is essential, and using linear interpolation to adjust the language model weights can also provide large further gains.

2.4. Minimum Bayes Risk Decoding

Finally, we experimented with improved decoding strategies for translation, particularly using minimum Bayes risk decoding (MBR, [12]). In normal translation, the decoder attempts to simply find the answer with the highest probability among the translation candidates

$$\hat{E} = \text{argmax}_E P(E|F) \tag{1}$$

in a process called Viterbi decoding. As an alternative to this, MBR attempts to find the hypothesis that minimizes risk

$$\hat{E} = \text{argmin}_E \sum_{E' \in \mathcal{E}} P(E'|F) L(E', E) \tag{2}$$

considering the posterior probability $P(E'|F)$ of hypotheses E' in the space of all possible hypotheses $\mathcal{E}$, as well as a loss $L(E', E)$ which determines how bad a translation E is if the true translation is E'. In this work (as with most others on MBR in MT) we use one minus sentence-wise BLEU+1 score [13] as our loss function

$$L(E', E) = 1 - \text{BLEU+1}(E', E). \tag{3}$$

In initial research on MBR, the space of possible hypotheses $\mathcal{E}$ was defined as the n-best list output by the decoder. This was further expanded by [14], who defined MBR over lattices. We tested both of these approaches (as implemented in the Moses decoder).

Finally, one fine point about MBR is that it requires a good estimate of the probability $P(E'|F)$ of hypotheses. In the discriminative training framework of [15], which is used in most modern SMT systems, scores of machine translation hypotheses are generally defined as a log-linear combination of feature functions such as language model or translation model probabilities

$$P(E'|F) = \frac{1}{Z} e^{\sum_i w_i \phi_i(E', F)} \tag{4}$$

where ϕ_i indicates feature functions such as the language model, translation model, and reordering model log probabilities, w_i is the weight measuring the relative importance of this feature, and Z is a partition function that ensures that the probabilities add to 1.

Choosing the weights w_i for each feature such that the answer with highest probability

$$\hat{E} = \text{argmax}_E P(E|F) \tag{5}$$

is the best possible translation is a process called "tuning," and essential to modern SMT systems. However, in most tuning methods, including the standard minimum error rate training [16] that was used in the proposed system, while the relative weight of each feature w_i is adjusted, the overall sum of the weights $\sum_i w_i$ is generally set fixed at 1. While this is not a problem when finding the highest probability hypothesis in 5, it will affect the probability estimates $P(E'|F)$, with

Decoding	dev2010	tst2010
Viterbi	27.59	31.01
MBR ($\lambda = 1$)	27.29	31.24
Lattice MBR ($\lambda = 1$)	26.70	31.25
Lattice MBR ($\lambda = 5$)	27.05	**31.81**

Table 6: BLEU Results using Minimum Bayes Risk decoding.

larger s assigning a larger probability to the most probable hypothesis, and a smaller s spreading the probability mass more evenly across all hypotheses.

In order to improve the calibration of our probability estimates, and thus improve the performance of MBR, we introduce an addition scaling factor λ into the calculation of our probability

$$P(E'|F) = \frac{1}{Z} e^{\lambda \sum_i w_i \phi_i(E',F)}. \quad (6)$$

Using this lambda, we tried every value in 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, and 10.0, and finally chose $\lambda = 5.0$, which gave the best performance on tst2010.

The final results of our system with Viterbi decoding (no MBR), regular MBR over n-best lists, and lattice MBR with the scaling factors of 1 and 5, are shown in Table 6. It can be seen that both MBR and lattice-based MBR give small improvements over the baseline without tuning λ, while tuning λ gives a large improvement.[3] The reason why MBR reduces the accuracy on dev2010 is because dev2010 was used in tuning the parameters during MERT, so the one-best answers tend to be better on average than they would be on a held-out test set.

3. Additional Results on English-French

This section presents additional results obtained on the English-French track. The results here, for the most part, did not obtain worthwhile BLEU improvements in preliminary experiments, so we did not include them in the official system as described in Section 2. Although the systems reported in this section use the same dev and test set as that of Section 2, the training conditions and system configurations have slight differences, so the results should not be directly compared. We include these (negative) results for reference purposes, in order to aid understanding of the English-French TED task.

3.1. Exploiting Out-of-domain Data

We experimented with the simplest approach to exploiting out-of-domain bitext in translation models: data concatenation. This can be seen as adaptation at the earliest stage of the translation pipeline, and has achieved competitive results on TED En-Fr [17]. Three conditions were tried: (1) TED-only data, (2) TED + News (NC), (3) TED + NC + EuroParl (EP). Results are shown in Table 7.

First, we observe that adding data gives slight improvements (29.32 to 29.57). To analyze the potential for improvement, we also measured BLEU using "CheatLM" decoding [18]. "CheatLM" is an analysis technique for TM adaptation where the language model is trained on the reference; this gives a optimistic estimate on what can be achieved by the translation model, if other components are tuned almost perfectly. Here we see that TED+NC+EP (59.93 BLEU) can achieve large improvements over TEDonly (55.10 BLEU), indicating the potential value of out-of-domain bitext. However, note that the corresponding OOV rate reduction is relatively small (1.2% to 0.52%). We hypothesize that out-of-domain probably is not helping because of improved word coverage, but rather because of improved word alignment estimation. In any case, the improvements are slight so we do not attempt to draw any further conclusions.

Data	standard	CheatLM	force	OOV
TEDonly	29.32	55.10	16%	1.2%
TED+NC	29.43	58.64	17%	0.85%
TED+NC+EP	**29.57**	**59.93**	21%	0.52%

Table 7: Translation Model Adaption by simple out-of-domain data concatenation. The "standard" and "CheatLM" columns show the BLEU scores on tst2012, using standard Moses decoding and "CheatLM" decoding. The column "force" shows the percentage of tst2010 sentences that can be translated into the reference using forced decoding. OOV indicates the token out-of-vocabulary rate.

3.2. Word Alignment & Phrase Table Combination

We investigated different alignment tools and ways to combine them, as shown in Table 8. Observations are as follows:

- GIZA++ and BerkeleyAligner achieve similar BLEU on this task.
- Concatenating GIZA++ and BerkeleyAligner word alignment results, prior to phrase extraction, achieves a small boost (29.57 to 29.89 BLEU).
- We also experimented with pilaign [19], a Bayesian phrasal alignment toolkit. This tool directly extracts phrases without resorting to the preliminary step of word alignments, and achieves extremely compact phrase table sizes (0.8M entries) without significantly sacrificing BLEU (29.24).
- Combining the GIZA++ and pialign phrase tables by Moses' multiple decoding paths feature did not improve results. Overall, we did not find much differ-

[3] It should be noted that due to constraints in the available data for these MBR experiments we are both tuning on testing on tst2010, but the tuning of λ also demonstrated gains in accuracy on the official blind test on tst2011 and tst2012 (37.33→37.90 and 38.92→39.47 respectively).

ence among these various approaches so we used the standard GIZA++ tool chain in the official submission.

Tool	BLEU	TableSize
1: GIZA++	29.57	109
2: BerkeleyAligner	29.39	170
3: pialign	29.24	0.8
1+2: ConcatAlign (GIZA,Berkeley)	**29.89**	200
1+3: TwoTable (GIZA,pialign)	29.56	201

Table 8: BLEU scores on tst2010 of various combinations of alignment and phrase training tools. TableSize shows the phrase-table size of corresponding method (in millions of entries). GIZA++ and BerkeleyAligner are trained the the TED+NC+EP bitext; pialign is trained only on TED, due to time constraints in our preliminary experiments.

3.3. Lexical Reordering Models

Several reordering models available in the Moses decoder were tried. In general, we found the full "msd-bidir-fe" option to perform best, despite the small number of word order differences between English and French. Results are shown in Table 9.

Reordering model	BLEU
msd-bidir-fe	**29.57**
msd-bidir-f	29.43
monotonicity-bidir-fe	29.29
msd-backward-fe	29.22
distance	28.99
msd-bidir-fe-collapse	28.86

Table 9: Comparison of Reordering models on tst2010.

3.4. MERT vs. PRO tuning

We compared two tuning methods: MERT and PRO [20]. We used the implementations distributed with Moses. For both MERT and PRO, we set the size of k-best list to $k = 100$, used 14 standard features, and removed duplicates in k-best lists when merging previously generated k-best lists. We ran MERT in multi-threaded setting until convergence. Since the number of random restarts in MERT greatly affects on the translation accuracy [21], we tried various number of random restarts for 1, 10, 20, and 50.[4] For PRO, we used MegaM[5] as a binary classifier with the default setting. We ran PRO for 25 iterations. We tried two kinds of PRO: [20] interpolated the weights with previously learned weights to improve the stability (henceforth "PRO-interpolated")[6], and the version that do not use such a interpolation (henceforth "PRO-basic").

[4]Currently, Moses's default setting is 20.

[5]http://www.cs.utah.edu/~hal/megam/

[6]We set the same interpolation coefficient value of 0.1 as [20] noted.

# of random restarts	Iteration	Dev BLEU	Time (m) Wall	Time (m) CPU
1	11	28.18	0.59	0.82
10	11	28.17	2.21	17.22
20	12	28.29	4.91	57.88
50	12	28.31	9.72	171.91

Table 10: The effect of the number of random restarts in MERT on BLEU score and multi-threaded time. "Iteration" denotes the number of iterations which MERT needs to be converged. "Time" denote the average time of weight optimization for each iteration, averaged over all iterations.

Method	Dev BLEU
MERT	**28.29**
PRO-basic	26.99
PRO-interpolated	27.11

Table 11: Comparison with MERT and PRO. For MERT, the number of random restarts was set to 20.

We first investigate the effect of the number of random restarts in MERT on BLEU score and run-time for each iteration. Table 10 shows the result. As the number of random restarts increases, BLEU score improves. However, the run-time increases as well. We used 20 random restarts to compare to PRO.

Table 11 shows the results of MERT and PRO. As can be seen in Figure 11, MERT exceeds PRO-basic by 1.3 points and PRO-interpolated by 1.18 points. As a result, we used MERT for tuning in Sections 2 and 4.

4. Systems for Translation into English

We participated in the translation of all 10 additional language-pairs of the TED Talk track. The source languages are Arabic (ar), German (de), Dutch (nl), Polish (pl), Brazilian-Portuguese (pt), Romanian (ro), Russian (ru), Slovak (sk), Turkish (tr), and Chinese (zh). The target language for all tasks is English (en).

Since all tasks translate into the same language, we are able to share the language model as well as many of the configurations for the Experimental Management System (EMS). This setup provides an invaluable chance to compare the same techniques across structurally-different languages, and is the focus of our work. Rather than optimizing for specific languages, we concentrate on building common systems under the same EMS framework and on comparing the performance of existing techniques cross-lingually.

It is interesting to note that the 10 language-pairs cover a diverse range of linguistic phenomenon. In terms of historical relationships, the Italic family (pt,ro) and Germanic family (de, nl) are expected to be closer to the target language of English. The Slavic family (pl,ru,sk), Arabic, and Turkish

languages exhibit rich morphology (fusional, non-catenative, or agglutinative). Additionally, the Germanic family may show word order differences (V2 and SOV) and Chinese requires word segmentation.

4.1. Experiments

Table 12 summarizes all the results (BLEU scores) for translation into English. In all language pairs, the baseline consists of a standard phrase-based Moses system (GIZA++ alignment, grow-diag-final-and heuristic, lexical ordering, 4-gram language model) trained on the TED Talks portion of the training data. MERT tuning is performed on the "dev2010" portion of the data and Table 12 shows test results on "tst2010."[7] While it is not possible to directly compare BLEU across languages, we do observe that the Italic and Germanic languages fare better on this TED task (> 25 BLEU), while Chinese, Turkish, and the Slavic languages perform poorly at $10 - 17$ BLEU.

We then proceeded to improve on these baseline results. First, adding additional out-of-domain data (nc=News Commentary, ep=Europarl, un=UN Multitext) to the language model increased results uniformly for all language pairs (line (b) of Table 12). We used an interpolated language model, trained in the same fashion as in our English-French system.

Next, we tried two strategies for handling rich morphology in the input. The "CompoundSplit" program in the Moses package was developed for languages with extensive noun compounding, e.g. German, and breaks apart words if sub-parts are seen in the training data over a certain frequency [22]. The alternate "Morfessor" program [23] is an unsupervised morphological analyzer based on the Minimum Description Length principle – it tries to find the the smallest set of morphemes that parsimoniously cover the training set. Morfessor is expected to segment more aggressively than CompoundSplit, especially because it can find both bound and free morphemes. However, we empirically found that Morfessor segments too aggressively for unknown words (i.e. each character becomes a morpheme), so we do not segment OOV words in dev/test.[8] The results in line (c) of Table 12 shows that German benefit most from CompoundSplit, while Arabic, Russian, and Turkish benefit from Morfessor. The remaining languages perform approximately equal or slightly better with these morphology enhancements, so in further experiments we keep the morphology pre-processing (de & ro uses CompoundSplit; others use Morfessor).

In line (d) of Table 12, we further added the Giga corpus to the interpolated language model. For some languages, this gave a large improvement (ar, de, pl, sk), while for other languages the results remain similar. Some of these results represent our official submission. In line (e), adding Lattice MBR decoding uniformly degraded results, so we chose not to include it. This is in contrast with our English-French results. We suspect that in this case uniformity of the training data and lack of diversity in the n-best list may have damaged MBR; the resulting translations appear similar in structure, but many have extraneous articles and determiners, which hurts BLEU. It should also be noted that unlike English-French, we did not calibrate the probability distribution by adjusting λ, which might also had a significant effect on the results. Finally in line (f), we added additional out-of-domain bitext for Translation Model training. This only helped slightly for pl and tr, while degrading other language pairs: we conclude that more advanced TM adaptation methods is necessary, and simply concatenating the bitext does not help.

Finally, we note that our submitted systems for each language achieve a 0.7-2.5 BLEU improvement over the respective baselines. We also achieve slight improvements in METEOR, despite not tuning for it. While the feature that helped most depends on language, we observe that morphological pre-processing and larger language models are generally worthwhile efforts.

[7] For Slovak, which lacked an official dev/test split, we split the development data, with the first half for tuning and the second half for testing. All source languages, except for Slovak, have comparable amounts of in-domain data (130k-145k sentence pairs).

[8] In other words, we keep OOV words as is and propagate it to the output. This implies that we lose the opportunity to translate OOV words whose component morphemes are seen in the training data. However, we think this conservative option is safer in the presence of potential over-segmentation.

5. Conclusion

This paper described our experiments with a number of existing machine translation techniques for the IWSLT 2012 TED task. Some of these techniques, such as minimum Bayes risk decoding with calibrated probabilities, language model interpolation, unsupervised morphology processing, translation model smoothing, and the use of large data proved to be effective. We also found that a number of techniques, including tuning using PRO, alignment combination, and data filtering had less of a positive effect.

6. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, HK, December 2012.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2007, pp. 177–180.

[3] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the Human Language Technology Conference (HLT-NAACL)*, 2003, pp. 48–54.

[4] D. S. Munteanu and D. Marcu, "Improving machine translation performance by exploiting non-parallel corpora," *Computational Linguistics*, vol. 31, no. 4, pp. 477–504, 2005.

SYSTEM	ar	de	nl	pl	pt	ro	ru	sk	tr
(a): baseline	21.6	26.8	30.6	15.5	35.6	28.7	16.8	16.8	12.5
(b): (a)+LM:nc,ep,un	21.9	26.9	31.4	15.6	36.1	29.2	17.3	17.7	12.6
(c): (b)+morphology									
compoundsplit	22.5	27.4	31.2	15.6	36.2	29.1	17.0	17.7	12.9
morfessor	23.4	26.8	31.6	15.6	36.3	28.8	17.6	17.7	13.6
(d): (c)+LM:giga	**24.1**	**28.0**	**31.4**	16.2	**36.2**	**29.4**	**17.5**	**18.4**	13.8
(e): (d)+lattice MBR	23.4	27.1	30.7	15.4	34.7	27.6	16.4	17.8	13.7
(f): (d)+TM (outdomain)	21.7	26.2	29.5	**16.4**	35.3	29.3	16.5	-	**13.9**
Δbleu: (d) or (f) - (a)	2.5	1.2	0.8	0.9	0.7	0.7	0.8	1.6	1.4
Δmeteor : (d) or (f) - (a)	1.7	0.7	0.2	0.7	0.2	0.3	0.8	0.6	1.5

Table 12: BLEU Results for Translations into English. Roughly, each row builds on top of the previous row. Boldface indicates official submission. For zh-en (not shown in table as the segmentation methods are different from other language pairs), the BLEU results are: 10.8 for character-based translation and 11.6 for word-based translation (Stanford word segmenter, PKU standard), using +LM:nc,ep,un but not +LM:giga nor +TM(outdomain), which degraded results.

[5] M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel, "The KIT English-French translation systems for IWSLT 2011," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2011.

[6] P. F. Brown, V. J. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, pp. 263–312, 1993.

[7] F. J. Och, C. Tillmann, and H. Ney, "Improved alignment models for statistical machine translation," *Proceedings of the 4th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 20–28, 1999.

[8] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, 2008.

[9] G. Foster, R. Kuhn, and H. Johnson, "Phrasetable smoothing for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006, pp. 53–61.

[10] F. Jelinek and R. L. Mercer, "Interpolated estimation of markov source parameters from sparse data," pp. 381–397, 1980.

[11] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of the 7th International Conference on Speech and Language Processing (ICSLP)*, 2002.

[12] S. Kumar and W. Byrne, "Minimum bayes-risk decoding for statistical machine translation," in *Proceedings of the Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics (NAACL) Meeting (HLT/NAACL)*, 2004.

[13] C.-Y. Lin and F. J. Och, "Orange: a method for evaluating automatic evaluation metrics for machine translation," in *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, 2004, pp. 501–507.

[14] R. Tromble, S. Kumar, F. Och, and W. Macherey, "Lattice Minimum Bayes-Risk decoding for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008, pp. 620–629.

[15] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002.

[16] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2003.

[17] K. Duh, K. Sudoh, and H. Tsukada, "Analysis of translation model adaptation for statistical machine translation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) - Technical Papers Track*, 2010.

[18] S. Matsoukas, personal communication, 2010.

[19] G. Neubig, T. Watanabe, S. Mori, and T. Kawahara, "Machine translation without words through substring alignment," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, Jeju, Korea, July 2012, pp. 165–174.

[20] M. Hopkins and J. May, "Tuning as ranking," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

[21] R. C. Moore and C. Quirk, "Random restarts in minimum error rate training for statistical machine translation," in *Proceedings of the 22th International Conference on Computational Linguistics (COLING)*, 2008, pp. 585–592.

[22] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proceedings of the 10th European Chapter of the Association for Computational Linguistics (EACL)*, 2003.

[23] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, 2002, pp. 21–30.

FBK's Machine Translation Systems for IWSLT 2012's TED Lectures

N. Ruiz, A. Bisazza, R. Cattoni, M. Federico

Fondazione Bruno Kessler-IRST
Via Sommarive 18, 38123 Povo (TN), Italy
nicruiz@fbk.eu

Abstract

This paper reports on FBK's Machine Translation (MT) submissions at the IWSLT 2012 Evaluation on the TED talk translation tasks. We participated in the English-French and the Arabic-, Dutch-, German-, and Turkish-English translation tasks. Several improvements are reported over our last year baselines. In addition to using fill-up combinations of phrase-tables for domain adaptation, we explore the use of corpora filtering based on cross-entropy to produce concise and accurate translation and language models. We describe challenges encountered in under-resourced languages (Turkish) and language-specific preprocessing needs.

1. Introduction

FBK's machine translation activities in the IWSLT 2012 Evaluation Campaign [1] focused on the speech recognition and translation of TED Talks[1], a collection of public speeches on a variety of topics and with transcriptions available in multiple languages. In this paper, we discuss our involvement in the official Arabic-English and English-French Machine Translation tasks, as well as the auxillary German-English, Dutch-English, and Turkish-English Machine Translation tasks.

We begin with an overview of the research procedure in common with all of language pair experiments in Section 2: namely, data filtering, phrase and reordering table fill-up, and mixture language modeling. In Section 4 we discuss our Arabic-English and Turkish-English MT systems. In Section 3 we discuss our English-French submissions. In Section 6 we discuss our German- and Dutch-English systems. Finally, in Section 8 we summarize our findings.

2. TED Machine Translation Overview

For all systems except for our Turkish-English system, we set up a standard phrase-based system using the Moses toolkit [2]. We construct a statistical log-linear model including a filled-up phrase translation and hierarchical reordering models [3, 4, 5], a primary mixture target language model (LM), as well as distortion, word, and phrase penalties. The distortion limit is set to the default value of 6, except for Arabic- and Turkish-English (see respective sections). As proposed by [6], statistically improbable phrase pairs are removed from our phrase tables.

For each target language, we train 5-gram mixture language models from the available corpora, as described in Section 2.3. The language models are trained with IRSTLM [7] with improved Kneser-Ney smoothing and no pruning. Additional experiments on hybrid word/class language models are performed in the Arabic-English task. The weights of the log-linear combination are optimized via minimum error rate training (MERT) [8].

In the following sections, we discuss the data selection, phrase and reordering table fill-up, and mixture language modeling used by each of our systems. We follow the discussion with our language-specific submissions.

2.1. Data selection

Each out-of-domain corpus was domain-adapted by filtering aggressively using a cross-entropy difference scoring technique described by [9] on the target side and optimizing the perplexity against the (target language) TED training data by incrementally adding sentences.

The idea of data selection is to find the subset of sentences within an out-of-domain corpus that better fits with a given in-domain corpus. Each sentence of the out-of-domain corpus is evaluated by comparing its likelihood (in terms of cross-entropy) to appear in the out-of-domain corpus against its likelihood to compare in the in-domain corpus. In order to decide how many sentences to keep, we build an out-of-domain language model incrementally and measure its perplexity on the in-domain TED data. The two language models we compare are built from the same dictionary, namely the in-domain words occurring more than a specified frequency. All other words in the in-domain and out-of-domain corpora are taken as out-of-vocabulary words. For this kind of problem it is generally sufficient to work with 3-grams language models estimated on words occurring at least twice in the in-domain set.

Figure 1 shows the effects of data selection on the four out-of-domain corpora used for language modeling in all of our foreign-to-English MT submissions. Three of the corpora are subcorpora drawn from seven available news text sources in the LDC English Gigaword (Fifth Edition) corpus.

[1] http://www.ted.com/talks

The statistics of each corpus are shown in Table 1.

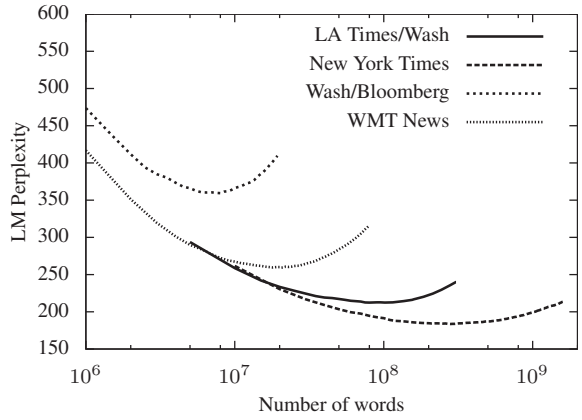


Figure 1: Effects of cross-entropy data selection on perplexity (PP) for the English monolingual out-of-domain data used by all foreign-to-English systems. Sentences are incrementally added based on their rank with trigram PP measures reported against the IWSLT 2010 TED development set. The PP scores reach a saddle point in which the inclusion of additional sentences worsens the language model. Each LM requires only a fraction of the entire available corpus.

	Unfiltered		Filtered		
Corpus	**Lines**	**Tokens**	***Lines**	***Tokens**	**% Filt**
Gigaword LAT	6.73M	312M	1.6M	80M	74.4
Gigaword NYT	38.7M	1.6B	6.75M	300M	81.3
Gigaword WP	421K	19.8M	135K	7M	64.6
WMT News	31M	849M	878K	20M	97.6

Table 1: Filtering statistics on the monolingual English (sub)corpora used in FBK's systems. Sentences were incrementally added until a local minimum perplexity value against the development set was reached.

2.2. Phrase table fill-up

As we did last year, we combine phrase tables via fill-up [10, 11]. Using the recommendations of [11], we add k-1 binary provenance features for each of the k phrase tables to combine. Treating the TED phrase table as in-domain, we merge out-of-domain phrase pairs that do not appear in the in-domain TED table, along with their scores. Moreover, out-of-domain phrase pairs with more than four source tokens are pruned. The fill-up process is performed in a cascaded order, first filling in missing phrases from the corpora that are closest in domain to TED.

2.3. Mixture language model adaptation

After performing data selection and cross-entropy filtering on the provided monolingual corpora, we perform LM domain adaptation via mixture modeling [12].

For our foreign-to-English MT submissions, we construct a common 5-gram mixture LM consisting of TED data, a subset of corpora from the LDC Gigaword fifth edition corpus, and the WMT News Commentary. From the Gigaword corpus, we select the articles from the Los Angeles Times/Washington Post, New York Times, and Washington Post/Bloomberg subcorpora. After performing cross-entropy filtering on each subcorpus, we perform mixture model adaptation with the TED corpus as the in-domain background. French language model statistics are reported in Section 3.3.

3. English-French

More monolingual and parallel data were available in the English-French translation task. Several of the corpora were too large and noisy to use efficiency, which underscored the necessity of data selection and filtering. In the following sections we discuss the data selection, phrase and reordering table fill-up, and mixture language modeling approaches used for our English-French MT systems and report results on the official test sets.

3.1. Data selection

We perform data selection using the cross-entropy filtering technique described above, both for language and for translation modeling. In order to filter parallel corpora, we apply the cross-entropy filtering technique on the French (target-side) texts and prune the corresponding English segments. Table 2 provides statistics on the preprocessed monolingual and parallel corpora used by our systems, before and after filtering. In both monolingual and parallel corpora we observe over a 85% reduction in the number of words by filtering.

	Unfiltered		Filtered		
Corpus	**Lines**	**Tokens**	***Lines**	***Tokens**	**% Filt**
Europarl	2.0M	61.9M	200K	4.2M	93.2
Giga French	19.7M	570M	1.08M	25.5M	96.6
Gigaword AFP	18.3M	668M	1.08M	46.1M	93.1
Gigaword APW	6.5M	255M	660K	34.7M	86.4
MultiUN	10.5M	290M	228K	5.2M	98.2
WMT News	7.5M	182M	900K	20.9M	88.5

Table 2: French filtering statistics on the tokenized and cleaned (sub)corpora used in FBK's systems. Europarl, Giga French, and MultiUN were used for translation model training, while French side of the Giga corpus and the monolingual Gigaword AFP and WMT News corpora were used for language model training.

3.2. Phrase table

More parallel data was available in the English-French translation task than the other MT tracks. In particular, the MultiUN and Giga French corpora were too large and noisy to use reliably for translation modeling without filtering. Table 2 shows that the size of these corpora were reduced by over 95% using cross-entropy filtering.

We use the filtered TED, Europarl, MultiUN, and Giga French parallel corpora for translation model training. Our experiments from last year showed little improvement from

using the parallel WMT News Commentary corpus. In order to reduce the size of the translation models and to stabilize MERT behavior, we independently train phrase and reordering tables on each corpus and experiment with several fill-up configurations with the TED as the in-domain corpus. Table 3 lists BLEU and TER evaluation results[2] on the IWSLT 2010 TED test set, three independent MERT runs for each fill-up combination. Each system uses the mixture LM described later in Section 3.3. In particular, we do not see any significant improvements filling up with using Europarl or MultiUN, but rather with the Giga French corpus. In order to improve the coverage of the TED and Giga fill-up models, we cascaded fill-up with Europarl and MultiUN respectively. While we do not observe significant improvement with the cascaded fill-up from Table 3, we later observe different results on our submitted runs.

	BLEU ↑			TER ↓		
System	**Avg**	$\overline{s}_{\textbf{sel}}$	p	**Avg**	$\overline{s}_{\textbf{sel}}$	p
TED-only	32.2	0.5	-	49.7	0.5	-
Fill(TED+Euro)	32.3	0.5	0.27	49.5	0.5	0.03
Fill(TED+UN)	32.2	0.5	0.60	49.4	0.5	0.00
Fill(TED+Giga)	32.5	0.5	0.03	49.4	0.5	0.01
Fill(TED+Giga+UN)	32.4	0.5	0.09	49.6	0.5	0.14
Fill(TED+Giga+Euro)	32.4	0.5	0.12	49.5	0.5	0.03

Table 3: Evaluation of phrase table combinations on the IWSLT 2010 TED test set, averaged across three MERT runs. Each translation system uses the mixture LM described in Section 3.3. Phrase tables are filled-up in a left-to-right order. p-values are relative to the system trained with only the TED phrase table. $\overline{s}_{sel}$ indicates the variance due to test set selection.

3.3. Language modeling

In order to determine which monolingual data to use for language modeling, we trained 5-gram language models on each unfiltered corpus and evaluated their perplexity scores on the in-domain TED development data. From our experiments last year, the monolingual WMT News Commentary corpus yielded well-performing LMs. The Gigaword corpus consisted of articles from the Agence France-Presse (AFP) and Associated Press Worldstream (APW) newswires. Our perplexity analyses showed that APW did not model the TED domain well; thus, we opt to omit it. To our surprise, the French side of the parallel Giga French corpus modeled the TED domain well after filtering – even better than the TED training data!

Rather than log-linearly combining four distinct LMs and optimizing four feature weights, we combine the LMs with mixture modeling and evaluate their cumulative effects on the IWSLT 2010 development set in Table 4. After confirming that the four LMs in combination improve perplexity, we construct a 5-gram mixture model. Table 5 suggests that the mixture LM alone is responsible for a 2.7 BLEU improvement over a TED-only 5-gram baseline.

[2]Evaluation results were performed with MultEval v0.3 [13].

Corpora	**PP dev$_{2010}$**	**% OOV**
TED	139.40	1.65%
Giga-EF	126.65	0.85%
TED + Giga-EF	85.60	0.7%
+ Gigaword AFP	81.34	0.4%
+ WMT News	80.19	0.4%

Table 4: Perplexity of 3-gram mixture LMs evaluated on the IWSLT 2010 development set. Giga French, Gigaword AFP, and WMT News corpora are incrementally added to the in-domain TED training corpus and provide excellent coverage of the development data.

PT	**LM**	**Metric**	**Opt 1**	**Opt 2**	**Opt 3**	**Avg**
TED	TED	BLEU	29.75	29.95	29.72	29.74
		NIST	7.167	7.184	7.178	7.170
	Mix	BLEU	32.37	32.44	32.44	32.42
		NIST	7.463	7.438	7.438	7.443

Table 5: Effects of mixture LM on the IWSLT 2010 TED test set. Results are calculated across three MERT optimizations with their weights averaged for final evaluation. The mixture LM results in roughly 2.7 BLEU and 0.27 NIST improvements against a TED-only phrase table.

3.4. Submitted runs

Our primary (*P*) and constrastive (*C*) results are reported in Table 6 and are compared to a simple TED baseline (*B*), consisting of TED-only phrase and reordering tables. All systems use the mixture LM described in the previous section. Each system's feature weights are averaged over three MERT optimizations. The fill-up model with Europarl yielded higher BLEU and NIST scores on both the 2010 development and test sets; thus by providing additional phrase coverage we opted to submit it as our primary system. Our TED+Giga fill-up system served as our contrastive baseline. Each system performed similarly on the official test sets, though the MultiUN filled-up model was not consistent across the different test sets. Our primary system performed equally with our contrastive baseline on the 2011 test set in terms of BLEU, but performed slightly (though not significantly) worse in terms of NIST, while on the 2012 test set we observe a 0.3 BLEU improvement.

	PT	**Metric**	**dev$_{2010}$**	**tst$_{2010}$**	**tst$_{2011}$**	**tst$_{2012}$**
B	TED	BLEU	27.71	32.22	–	–
		NIST	6.600	7.397	–	–
P	Fill(TED	BLEU	28.42	32.42	37.43	37.29
	+Giga+Euro)	NIST	6.697	7.443	7.713	8.039
C_1	Fill(TED	BLEU	28.11	32.39	37.43	36.99
	+Giga)	NIST	6.660	7.450	7.737	8.024
C_2	Fill(TED	BLEU	28.23	32.52	37.36	37.24
	+Giga+UN)	NIST	6.681	7.460	7.715	8.051

Table 6: Results of submitted runs evaluated on the IWSLT TED development and test sets. Evaluation on the 2010 data sets are compared against a TED-only phrase table. All systems use the mixture LM described in Section 3.3. MT system weights are averaged across three MERT optimizations for final evaluation.

4. Arabic-English

The Arabic-English language pair is characterized by notable differences in morphological richness and word order. We follow last year's experience to deal with morphology and address word reordering by using an improved version of the distortion penalty that was proposed by [14]. In addition to that, we integrate a hybrid class language model [15] that proved to improve our system of last year.

4.1. Preprocessing

For Arabic we use our in-house tokenizer that also removes diacritics and normalizes special characters and digits. Then, segmentation is performed by the AMIRA toolkit [16] based on SVM classifiers, according to the Arabic Treebank (ATB) scheme that isolates conjunctions *w+* and *f+*, prepositions *l+, k+, b+*, future marker *s+*, pronominal suffixes, but not the article *Al+*. Arabic training data statistics are given in Table 7.

		AR tokens		EN tokens
Corpus	**Lines**	**unsegm.**	**Amira-segm.**	
TED	137K	2.1M	2.5M	2.7M
MultiUN	8M	188M	224M	220M

Table 7: Arabic-English training data statistics showing number of Arabic tokens before and after segmentation.

4.2. Phrase table

While word alignment is obtained on the union of all available data, the translation model is built by filling up a TED-only phrase table with a MultiUN-only phrase table. As previously said, out-of-domain (MultiUN) phrase pairs with more than four source words are filtered out. The lexicalized reordering table is obtained with the same procedure.

4.3. Early distortion cost

Moore and Quirk [14] proposed an improvement to the distortion penalty used in Moses, which consists in "incorporating an estimate of the distortion penalty yet to be incurred into the estimated score for the portion of the source sentence remaining to be translated." The new distortion penalty has the same value as the usual one over a complete translation hypothesis (provided that the jump from the last translated word to the end of the sentence is taken into account). As a difference, though, it anticipates the gradual accumulation of the total distortion cost making partial translation hypotheses with the same number of covered words more comparable with one another. We have implemented this 'early distortion cost' option in the Moses platform and used it in our systems. As shown in Table 8, increasing the distortion limit from the default value of 6 to 8 has normally a negative impact because standard distortion does not properly control long jumps. On the contrary, when *early* distortion cost is used, a slightly higher distortion limit is preferable, yielding an improvement of +0.2 BLEU and +0.04 NIST over the baseline.

DL	DC	BLEU	NIST
6	std	26.12	6.514
8	std	25.95	6.460
8	**edc**	26.31	6.551

Table 8: Effects of distortion limit (DL) and distortion cost (DC), standard or early, on the IWSLT 2010 TED test set.

4.4. Mixture language modeling

In Arabic-English too, we use mixture modeling for domain adaptation. Concerning data selection, we find that a 4-gram LM trained on unfiltered data performs slightly better in terms of BLEU than the filtered 5-gram LM presented in section 2.3 (see first two rows of Table 9). A possible explanation is that, if translation gets more difficult, especially due to reordering, relying on a much larger number of n-grams helps to discriminate correct versus incorrect phrase concatenations. This discrimination capability may not reflect on the perplexity, which only measures how a LM predicts correct text. Thus, we use the unfiltered LM for the Arabic-English systems. It should be noted, though, that this model requires twice as much memory to function.

LM	BLEU	NIST
MixFiltered.5g	25.92	6.465
MixAll.4g	26.31	6.551
MixAll.4g + TED.Hybrid10g	26.65	6.591

Table 9: Effects of data selection and hybrid language modeling on the IWSLT 2010 TED test set.

4.5. Hybrid language modeling

In addition to the mixture model, we use an in-domain hybrid word/class LM that was proposed by [15] to address style adaptation when out-of-domain data is likely to bias the system towards an unsuitable language style (e.g. news versus talks). Following the paper, we train a high order (10-gram) LM on TED data where infrequent words were mapped to their most likely Part-of-Speech tags, and frequent words to their lemma. We set the frequency threshold so that 25% of the tokens – corresponding to about 2% of the types – are replaced by part-of-speech (POS) tags. Adding this model to the log-linear combination yields a gain of +0.3 BLEU and +0.04 NIST (see Table 9).

4.6. Submitted runs

Table 10 presents results of our baseline (B), primary (P) and contrastive (C) systems on the IWSLT 2010, 2011 and 2012 TED test sets. All Arabic-English systems use the same phrase and reordering models, obtained by fill-up of TED

and UN data. Our best submission is obtained with early distortion cost, a distortion limit of 8 words and an in-domain hybrid LM in addition to a large unfiltered mixture LM.

	LM	DL	Metric	tst_{2010}	tst_{2011}	tst_{2012}
B	MixAll.4g	6	BLEU	26.12	–	–
			NIST	6.514	–	–
P	MixAll.4g	8	BLEU	26.65	25.46	27.86
	+TED.Hybrid10g	[edc]	NIST	6.591	6.232	6.881
C_1	MixAll.4g	8	BLEU	26.31	25.19	27.74
		[edc]	NIST	6.551	6.205	6.903
C_2	MixFiltered.5g	8	BLEU	26.11	25.13	27.54
	+TED.Hybrid10g	[edc]	NIST	6.520	6.190	6.828

Table 10: Results of Arabic-English submitted runs evaluated on the IWSLT TED development and test sets.

5. Turkish-English

The additional training data provided for this language pair was limited to the South European Times news corpus. In our experiments we found that this data was not helpful for translation modeling and decided to use it only for word alignment[3]. A reason for this could be the size of this corpus – only slightly larger than the TED data – that is enough to bring noise into the system but not enough to improve its coverage in a significant way.

We then focus on preprocessing techniques to address the agglutinative Turkish morphology and evaluate the performance of phrase-based against hierarchical systems.

5.1. Morphological segmentation

Turkish preprocessing involves supervised morphological analysis [17] and disambiguation [18], followed by selective morpheme segmentation as described in [19]. We compare two of the segmentation schemes that were proposed and tested on the BTEC task by [19] and [20]:

- **'MS6'** deals only with nominal suffixes (case and possessive),
- **'MS15'** deals with nominal suffixes and verbal suffixes (copula, person subject, negation, ability, passive and causative suffixes).

The latter segmentation scheme is more aggressive, which is good for model coverage but can make the translation harder (especially the reordering problem, due to the larger number of possible input permutations).

To evaluate the actual importance of supervised methods, we also build a contrastive system using a fully data-driven segmentation approach proposed by [21] and implemented in the Morfessor Categories-MAP software. We train Morfessor on the TED training corpus, and obtain a unique segmentation of each word type into a sequence of morpheme-like units (*morphs*). As an intermediate solution between words and morphs – which are typically rather short – we concatenate the sequence of non-initial morphs to form so-called *word endings*[4]. In this way, each word can be segmented into at most two parts.

		TR tokens				EN
Corpus	Lines	unsegm.	MS6	MS15	Morf.	tokens
TED	125K	1.8M	2.0M	2.2M	2.4M	2.4M

Table 11: Turkish-English training data statistics showing how the number of Turkish tokens varies according to the segmentation method: supervised (MS6 and MS15) or unsupervised (Morfessor).

Turkish training data statistics in different segmentation settings are given in Table 11, while the effect on translation quality is shown in Table 12. Notice the very high distortion limit chosen because of the important order differences between English and Turkish, a head-final SOV language. In this set of experiments we use a 4-gram mixture LM trained on unfiltered data. The results show that supervised segmentation (MS15) can noticeably outperform the unsupervised one (Morfessor *word endings*), but they also show that the choice of a particular segmentation scheme is very important. In fact, the supervised MS6 scheme does no better than the unsupervised. We decide to use MS15 for the rest of the evaluation, however it is possible that the unsupervised approach may be improved by devising other ways to recombine the morphs.

DL	DC	Segment.	BLEU	NIST
15	std	MS6	13.61	5.280
15	std	MS15	14.38	5.273
15	std	unsup.	13.45	5.080
15	**edc**	MS15	14.53	5.299

Table 12: Effects on translation quality (IWSLT 2010 test set) of Turkish morphological segmentation, and of standard versus *early* distortion cost (see Section 4.3).

5.2. Translation model: phrase-based vs. hierarchical

As we only use TED training data, no adaptation technique is required for translation modeling.

Given the global and hierarchical nature of word reordering patterns in this language pair, we thought that a hierarchical translation system [23] could work better than a regular phrase-based one. We then construct a rule table with maximum rules span 15 and Good Turing score smoothing, and switch to chart decoding (all within the Moses platform).

The hierarchical system strongly outperforms the phrase-based one, with a +1.7 BLEU and 0.25 NIST gain (see Table 13) proving the complexity of the word reordering problem in Turkish-English.

[3] We concatenated the two corpora, ran GIZA++ on them, but only used the TED portion of the result.

[4] This approach is sometimes adopted in language modeling for Turkish speech recognition, see for instance [22].

5.3. Submitted runs

We submitted two systems: the hierarchical as primary and the phrase-based with early distortion cost and a high distortion limit (15) as contrastive. Both of our official systems include a 6-gram mixture LM trained on the filtered data described in Section 2.1.

	System	Segm.	Metric	tst_{2010}	tst_{2011}	tst_{2012}
P	hierarchical	MS15	BLEU	16.61	17.24	17.15
			NIST	5.570	5.560	5.702
C	phrase-based (dl=15, edc)	MS15	BLEU	14.92	15.45	15.24
			NIST	5.318	5.289	5.145

Table 13: Results of Turkish-English submitted runs evaluated on the IWSLT TED development and test sets.

6. German-English

Translating German compound words (also known as "compounds") is a challenge for Machine Translation: the first subsection focuses on the experiments we performed on compounds splitting. We subsequently report on the translation and language models used in our submissions and present our system results on the official test sets.

6.1. Word splitting

In order to choose the best splitter sub-system, we performed some preliminary experiments. We use the splitting tool provided in Moses (see [24]), which is based on a trainable model. We test several splitter configurations with models trained on all the German data available for the MT track of the TED Task, but with different filtering techniques and parameter settings, inspired by [25]). For the sake of efficiency, we perform the experiments on the TED corpora (namely the provided training and 2010 development and test sets). After applying a standard tokenization step, different groups of data sets are obtained, one for each splitting configuration.

We conduct two sets of experiments; in the first we compute the perplexity and OOV-rate on the dev and test sets using the LM learned on the training set, while in the second we build SMT systems for each splitting configuration and evaluate their translations. It is worth noting that the splitters work only on the source language and do not affect the target language (English).

Table 14 lists the outcomes of the first set of experiments: the *normal* splitter utilizes the default parameter setting of the tool, while in the *aggressive* splitter we change the parameters to allow decomposition into short words (minimum 2 characters). The best performance in terms of perplexity and OOV-rate reduction is exhibited by the aggressive splitter.

There are no statistically significant differences among the translations provided by the three systems (unsplitted, normal- and aggressive-splitting). This can be explained mainly by the limited size of the training set. In the same experiments performed with all the available German data, we observe a marginal but statistically significant improvement on translation scores when performing both normal and aggressive splitting.

Split	Set	Tokens	Voc	Perplexity	OOV%
no	training	2419470	101623	–	–
	dev_{2010}	19082	4194	556.26	3.15
	tst_{2010}	30316	5181	417.11	2.66
normal	training	2474654	78113	–	–
	dev_{2010}	19444	4160	497.21	2.37
	tst_{2010}	30924	5072	377.40	1.85
aggressive	training	2508243	72091	–	–
	dev_{2010}	19725	4140	464.94	2.11
	tst_{2010}	31312	5027	355.26	1.62

Table 14: Statistics on the German TED sets obtained by varying the splitting configuration. The *aggressive* splitter exhibits the best performance in terms of perplexity and OOV-rate reduction.

6.2. Phrase table

For translation modeling we use the four provided data sets. The MultiUN bilingual entries are obtained by aligning parallel documents at sentence level with the Hunalign 1.1 tool [26] after standard tokenization. The statistics of the tokenized unsplitted corpora are shown in Table 15.

Corpus	Lines	DE tokens	EN tokens
TED	130K	2.4M	2.6M
news-commentary-v7	159K	4.0M	3.9M
MultiUN	163K	5.6M	5.6M
europarl-v7	1.9M	50.5M	53.0M

Table 15: German-English parallel training corpora statistics.

While word alignment is obtained on the union of all available data, the translation model is built by filling up a TED-only phrase table with two other phrase tables: the former obtained from WMT News Commentary v7 corpus and the latter from the union of MultiUN and Europarl v7 corpora. This partition has been chosen to maximize domain homogeneity in the three sub-corpora. The lexicalized reordering table is obtained with the same procedure.

6.3. Submitted runs

Table 16 presents results of our primary (P) and contrastive (C) systems on the IWSLT 2010, 2011 and 2012 TED test sets. Both systems use the English 5-gram mixture LM previously described in section 2.3 and differ only on the word splitting technique. Evaluation scores are rather close; the aggressive splitter appears to exhibit slightly better (although not statistically significant) performance.

7. Dutch-English

In the following sections we present the systems developed for the Dutch-English MT track of the TED task.

	Splitter	Metric	tst_{2010}	tst_{2011}	tst_{2012}
P	aggressive	BLEU	29.36	32.38	28.17
		NIST	7.257	7.513	7.004
C	normal	BLEU	29.49	32.13	28.12
		NIST	7.224	7.447	7.003

Table 16: Results of submitted runs evaluated on the German-English IWSLT TED development and test sets.

7.1. Word splitting

Like German, the Dutch language includes compounds. However, no specific splitting experiments were performed on Dutch: as splitters, we ported into Dutch the best splitting configurations found in our German experiments. The splitting models were trained on all available Dutch corpora.

7.2. Phrase table

For translation modeling, we use both the TED and Europarl v7 corpora. The statistics of the tokenized unsplit corpora are shown in Table 17.

Corpus	Lines	NL tokens	EN tokens
TED	128K	2.3M	2.5M
europarl-v7	2.0M	55.3M	54.8M

Table 17: Dutch-English parallel training corpora statistics.

Word alignment is obtained on the concatenation of both corpora. The translation model is built by filling up the TED-only phrase table with the out-of-domain Europarl phrase table. The same procedure is applied for the lexicalized reordering table.

7.3. Submitted runs

Table 18 presents results of our primary (P) and contrastive (C_1 and C_2) systems on the IWSLT 2010, 2011 and 2012 TED test sets. The three systems differ in the splitters (normal for P and C_1, aggressive for C_2) and language models: all of them use the English mixture LM previously described in section 2.3, but differ in length (4-gram for P, 5-gram for C_1, 6-gram for C_2). The evaluation scores do not highlight a single outperforming system.

	Splitter	Metric	tst_{2010}	tst_{2011}	tst_{2012}
P	normal	BLEU	33.85	36.11	32.68
		NIST	7.763	7.921	7.743
C_1	normal	BLEU	33.91	36.23	32.48
		NIST	7.759	7.946	7.722
C_2	aggressive	BLEU	33.84	35.82	32.68
		NIST	7.726	7.881	7.725

Table 18: Results of submitted runs evaluated on the Dutch-English IWSLT TED development and test sets.

8. Conclusions

We presented our submission runs to the IWSLT 2012 Evaluation Campaign for the TED MT tracks. Our MT systems benefited most from data filtering techniques and mixture language modeling. In particular, we observed significant BLEU improvements using mixture modeling over TED-only baselines. We also took advantage of phrase and reordering table fill-up models for further domain adaptation that additionally compresses the size of the translation system.

In Arabic-English, we used early distortion cost and incorporated a hybrid word/class language model to adapt to the style of talks, while for Germanic languages, we explored the effects of various compound splitting techniques. For Turkish-English, we compared several approaches to morphological segmentation and used a hierarchical SMT system.

9. Acknowledgements

This work was partially supported by TOSCA-MP project (IST-287532) and the EU-BRIDGE project (IST-287658), which are both funded by the European Commission under the Seventh Framework Programme for Research and Technological Development.

10. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. of the International Workshop on Spoken Language Translation*, December 2012.

[2] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic, 2007, pp. 177–180.

[3] C. Tillmann, "A Unigram Orientation Model for Statistical Machine Translation," in *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, 2004.

[4] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh system description for the 2005 IWSLT speech translation evaluation," in *Proc. of the International Workshop on Spoken Language Translation*, October 2005.

[5] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *EMNLP*

'08: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Morristown, NJ, USA: Association for Computational Linguistics, 2008, pp. 848–856.

[6] H. Johnson, J. Martin, G. Foster, and R. Kuhn, "Improving translation quality by discarding most of the phrasetable," in *In Proceedings of EMNLP-CoNLL 07*, 2007, pp. 967–975.

[7] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models," in *Proceedings of Interspeech*, Melbourne, Australia, 2008, pp. 1618–1621.

[8] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, E. Hinrichs and D. Roth, Eds., 2003, pp. 160–167.

[9] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *ACL (Short Papers)*, 2010, pp. 220–224.

[10] P. Nakov, "Improving English-Spanish Statistical Machine Translation: Experiments in Domain Adaptation, Sentence Paraphrasing, Tokenization, and Recasing. ," in *Workshop on Statistical Machine Translation, Association for Computational Linguistics*, 2008.

[11] A. Bisazza, N. Ruiz, and M. Federico, "Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation," in *International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011.

[12] P. Clarkson and A. Robinson, "Language model adaptation using mixtures and an exponentially decaying cache," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, Munich, Germany, 1997, pp. 799–802.

[13] J. Clark, C. Dyer, A. Lavie, and N. Smith, "Better hypothesis testing for statistical machine translation: Controlling for optimizer instability," in *Proceedings of the Association for Computational Lingustics*, ser. ACL 2011. Portland, Oregon, USA: Association for Computational Linguistics, 2011, available at http://www.cs.cmu.edu/ jhclark/pubs/significance.pdf.

[14] R. C. Moore and C. Quirk, "Faster beam-search decoding for phrasal statistical machine translation," in *In Proceedings of MT Summit XI*, 2007.

[15] A. Bisazza and M. Federico, "Cutting the long tail: Hybrid language models for translation style adaptation," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Avignon, France: Association for Computational Linguistics, April 2012, pp. 439–448.

[16] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks," in *HLT-NAACL 2004: Short Papers*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 149–152.

[17] K. Oflazer, "Two-level description of Turkish morphology," *Literary and Linguistic Computing*, vol. 9, no. 2, pp. 137–148, 1994.

[18] T. G. H. Sak and M. Saraçlar, "Morphological disambiguation of Turkish text with perceptron algorithm," in *Proc. of CICLing*, 2007, pp. 107–118.

[19] A. Bisazza and M. Federico, "Morphological preprocessing for turkish to english statistical machine translation," in *International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, 2009.

[20] A. Bisazza, I. Klasinas, M. Cettolo, and M. Federico, "FBK @ IWSLT 2010," in *International Workshop on Spoken Language Translation (IWSLT)*, Paris, France, 2010.

[21] M. Creutz and K. Lagus, "Inducing the morphological lexicon of a natural language from unannotated text," in *International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, 2005.

[22] H. Erdoğan, O. Büyük, and K. Oflazer, "Incorporating language constraints in sub-word based speech recognition," in *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, 2005, pp. 98–103.

[23] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 263–270.

[24] P. Koehn and K. Knight, "Empirical methods for compound splitting," in *Proceedings of Meeting of the European Chapter of the Association of Computational Linguistics (EACL)*, 2003.

[25] K. Macherey, A. Dai, D. Talbot, A. Popat, and F. Och, "Language-independent compound splitting with morphological operations," in *Proceedings of the 49th Annual Meeting of the Association of Computational Linguistics (ACL)*. Portland, USA: Association for Computational Linguistics, 2011.

[26] D. Varga, , L. Nmeth, P. Halacsy, A. Kornai, V. Tron, and V. Nagy, "Parallel corpora for medium density languages," in *Proceedings of the RANLP 2005*, 2005, pp. 590–596.

The RWTH Aachen Speech Recognition and Machine Translation System for IWSLT 2012

Stephan Peitz, Saab Mansour, Markus Freitag, Minwei Feng, Matthias Huck
Joern Wuebker, Malte Nuhn, Markus Nußbaum-Thom and Hermann Ney

Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
Aachen, Germany
<surname>@cs.rwth-aachen.de

Abstract

In this paper, the automatic speech recognition (ASR) and statistical machine translation (SMT) systems of RWTH Aachen University developed for the evaluation campaign of the *International Workshop on Spoken Language Translation* (IWSLT) 2012 are presented. We participated in the ASR (English), MT (English-French, Arabic-English, Chinese-English, German-English) and SLT (English-French) tracks. For the MT track both hierarchical and phrase-based SMT decoders are applied. A number of different techniques are evaluated in the MT and SLT tracks, including domain adaptation via data selection, translation model interpolation, phrase training for hierarchical and phrase-based systems, additional reordering model, word class language model, various Arabic and Chinese segmentation methods, postprocessing of speech recognition output with an SMT system, and system combination. By application of these methods we can show considerable improvements over the respective baseline systems.

1. Introduction

This work describes the automatic speech recognition (ASR) and statistical machine translation (SMT) systems developed by RWTH Aachen University for the evaluation campaign of IWSLT 2012 [1]. We participated in the ASR track, machine translation (MT) track for the language pairs English-French, Arabic-English, Chinese-English, German-English and the spoken language translation (SLT) track. State-of-the-art ASR, phrase-based and hierarchical machine translation systems serve as baseline systems. To improve the MT baselines, we evaluated several different methods in terms of translation performance. We show that phrase training for the phrase-based (forced alignment) as well as for hierarchical approach (forced derivation) can reduce the phrase table size while even improving translation quality. In addition, different word segmentation methods are tested for both Arabic and Chinese as source language. For English as source language, we perform a part-of-speech-based adjective reordering as preprocessing step. System combination is employed in three language pairs of the MT track to improve the translation quality further. Moreover, we investigate the use of the Google Books n-grams. For the SLT track, an SMT system is applied to perform a postprocessing of the given ASR output. This paper is organized as follows. In Section 2 and 3 we describe our ASR system and baseline translation systems. Sections 4 and 5 give an account of the phrase training procedure for the hierarchical phrase-based system and the system combination applied in several MT tasks. Our experiments for each track are summarized in Section 6. We conclude in Section 7.

2. ASR System

The ASR system is based on our English speech recognition system that we also successfully applied in Quaero evaluations [2].

In the acoustic feature extraction, the system computes Mel-frequency cepstral coefficients (MFCC) from the audio signal, which are transformed with a vocal tract length normalization (VTLN). In addition, a voicedness feature is computed. Acoustic context is incorporated by concatenating nine feature vectors in a sliding window. The resulting feature vector is reduced to 45 dimensions by means of a linear discriminant analysis (LDA). Furthermore, bottleneck features derived from a multilayer perceptron (MLP) are concatenated with the feature vector.

The acoustic model is based on hidden Markov models (HMMs) with Gaussian mixture models (GMMs) as emission probabilities. The GMM has a pooled, diagonal covariance matrix. It models 4500 generalized triphones which are derived by a hierarchical clustering procedure (CART). The parameters of the GMM are estimated with the expectation-maximization (EM) algorithm with a splitting procedure according to the maximum likelihood criterion.

The language model is a Kneser-Ney smoothed 4-gram. Several language models are trained on different datasets. The final language model is obtained by linear interpolation.

Table 1: Acoustic training data of ASR system

Corpus	Amount of data [hours]
quaero-2011	268h
hub4+tdt4	393h
epps	102h

Table 2: Language model training data of ASR system

Corpus	Amount of data [running words]
Gigaword 4	2.6B
TED	2.7M
Acoustic transcriptions	5M

The vocabulary of the recognition lexicon is obtained by applying a count-cut-off on the language model data. Each word in the lexicon can have multiple pronunciations. Missing pronunciations are derived with a grapheme-to-phoneme tool.

The recognition is structured in three passes, In the first pass, a speaker independent model is used. The recognition result of the first pass is used for estimating feature transformations for speaker adaptation (CMLLR). The second pass uses the CMLLR transformed features. Finally, a confusion network decoding is performed on the word lattices obtained from the second pass.

The acoustic model of the ASR system is trained on 793 hours of transcribed acoustic data in total, see Table 1. The acoustic training data consists of American broadcast news data (hub4+tdt4), European parliament speeches (epps), and British broadcast conversations (quaero). The MLP is trained on the 268 hours of the quaero corpus only. We use 4500 triphone states and perform eight EM splits, resulting in a GMM with roughly 1.1 million mixture components.

The language model is trained on a large amount of news data (Gigaword), the transcriptions of the audio training data, and a small amount of in-domain data (TED), see Table 2. The recognition lexicon consists of 150k words.

3. Baseline SMT Systems

For the IWSLT 2012 evaluation RWTH utilized state-of-the-art phrase-based and hierarchical translation systems as well as our in-house system combination framework. GIZA++ [3] was employed to train word alignments, all LMs were created with the SRILM toolkit [4] and are standard 4-gram LMs with interpolated modified Kneser-Ney smoothing, unless stated otherwise. We evaluate in truecase, using the BLEU [5] and TER [6] measures.

3.1. Phrase-based Systems

For the phrase-based SMT systems, we used in this work both an in-house implementation of the state-of-the-art MT decoder (PBT) described in [7] and the implementation of the decoder based on [8] (SCSS) which is part of RWTH's open-source SMT toolkit Jane 2.1 [1]. We use the standard set of models with phrase translation probabilities and lexical smoothing in both directions, word and phrase penalty, distance-based reordering model, an *n*-gram target language model and three binary count features. The parameter weights are optimized with MERT [9] (SCSS, HPBT) or the downhill simplex algorithm [10] (PBT).

3.2. Hierarchical Phrase-based System

For our hierarchical setups, we employed the open source translation toolkit Jane [11], which has been developed at RWTH and is freely available for non-commercial use. In hierarchical phrase-based translation [12], a weighted synchronous context-free grammar is induced from parallel text. In addition to contiguous *lexical* phrases, *hierarchical* phrases with up to two gaps are extracted. The search is carried out with a parsing-based procedure. The standard models integrated into our Jane systems are: phrase translation probabilities and lexical smoothing probabilities in both translation directions, word and phrase penalty, binary features marking hierarchical phrases, glue rule, and rules with non-terminals at the boundaries, four binary count features, phrase length ratios and an *n*-gram language model. Optional additional models are IBM model 1 [13], discriminative word lexicon (DWL) models, triplet lexicon models [14], a discriminative reordering model [15] and several syntactic enhancements like preference grammars and string-to-dependency features [16]. We utilize the cube pruning algorithm [17] for decoding and optimize the model weights with standard MERT [9] on 100-best lists.

4. Forced Derivation

As proposed in [18], an alternative to the heuristic phrase extraction from word-aligned data is to train the phrase table with an EM-inspired algorithm. Since in [18] a phrase table for a phrase-based system was learned, we employed the idea of force-aligning the training data on a hierarchical phrase-based setup [19]. Instead of applying a modified version of the decoder, a synchronous parsing algorithm based on two successive monolingual parses is performed. The idea of the two-parse algorithm is to first parse the source sentence. Then, phrases extracted from the source parse tree are used to parse the target sentence. After parsing, we apply the inside-outside algorithm on the generated target parse tree to compute expected counts for each applied phrase. Using the expected counts, we update the phrase probabilities and apply a threshold pruning on the phrase table. Leave-one-out

[1] http://www-i6.informatik.rwth-aachen.de/jane/

Table 3: Forced Derivation (FD) results for the MT task English-French including phrase table (PT) size.

system	dev		test		PT size
	BLEU	TER	BLEU	TER	# phrases
baseline	27.4	56.9	30.4	51.2	72M
FD	27.6	56.6	30.5	51.3	8.7M

is applied to counteract over-fitting effects. We tested this procedure on the English-French MT task. The results are shown in Table 3. The phrase table size was reduced by 88% without hurting performance.

5. System Combination

System combination is used to produce consensus translations from multiple hypotheses generated with different translation engines. System combination can be divided into two steps. The first step produces a word to word alignment for the given single system hypotheses. In a second step a confusion network is constructed. Then, the hypothesis with the highest probability is extracted from this confusion network. For the alignment procedure, we have to choose one of the given single system hypotheses as primary system. To this primary system all other hypotheses are aligned and thus the primary system defines the word order. In Figure 1 a system combination of four different system is shown. We select the bold hypothesis as primary hypothesis. The other hypotheses are aligned to the primary using the METEOR [20] alignment. The resulting hypotheses have different word lengths and thus it is possible to align a word to an empty word marked as $. Once the alignment is given, we are able to built a confusion network. As the hypotheses consist of different words and may have different sentence length, the unaligned words could produce incorrect arcs. To fix the incorrect arcs, we introduce a reordering model based on the language model scores of the given adjacent incorrect arcs. For unaligned parts, we take the hypothesis with the highest language model score and align the unaligned parts of all hypotheses to that one. As result we get a more meaningful confusion network. In Figure 1 different confusion networks with and without the reordering model are shown. A more compact representation of the confusion network is given in Figure 2.

As choosing a primary hypothesis is a hard decision, we build for each hypothesis as primary system one confusion network. To combine these different networks, we just use the Union operation from the automata theory. The next step is to extract the most probably translation from the confusion network. Each arc in the confusion network is rescored with different statistical models as word or phrase counts of the single systems, a language model score, a word penalty and a binary feature which marks the primary system of the partial confusion network. We give each model a weight and

system hypotheses	**this is it**						
	that was future						
	this is in the future						
	future is this						
alignment	that|**this** was|**is** $\|**it** future\|**$**						
	this|**this** is|**is** $\|**it** in\|**$** the|**$** future\|**$**						
	future|**$** this|**this** is|**is** $|**it**						
confusion network	**$**	**this**	**is**	**it**	**$**	**$**	**$**
	$	that	was	$	future	$	$
	$	this	is	$	in	the	future
	future	that	is	$	$	$	$
reordering of unaligned words	**$**	**this**	**is**	**it**	**$**	**$**	**$**
	$	that	was	$	$	$	future
	$	this	is	$	in	the	future
	$	this	is	$	$	$	future

Figure 1: Example for system combination of four different hypotheses.

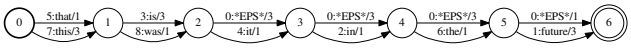


Figure 2: Confusion network of four different hypotheses.

Table 4: System combination results for the MT tasks English-French (en-fr), Arabic-English (ar-en) and Chinese-English (zh-en).

system		tst2010	
		BLEU	TER
en-fr	best single system	32.0	50.1
	system combination	32.9	42.9
ar-en	best single system	27.1	54.4
	system combination	28.0	53.4
zh-en	best single system	14.7	74.5
	system combination	15.4	74.1

combine them in a log-linear model. The weights can be optimized with MERT and the translation with the best score within the lattice is the consensus translation.

By applying system combination in the English-French, Arabic-English and Chinese-English MT task, we achieve improvements of up to +0.9 points in BLEU and up to -1.0 points in TER.

6. Experimental Evaluation

6.1. Automatic Speech Recognition

In Table 5 we compare the word error rate (WER) of the three different passes. A lower WER indicates a better recognition quality. We achieve an improvement of 2.5 points in WER by applying the second pass. Furthermore, the confusion network decoding improves the recognition by 0.2 points.

Table 5: Results of the English ASR task. Our ASR system is incrementally improved with each pass.

	dev2010	**tst2010**
pass 1	20.0	18.4
pass 2	17.5	15.9
cn-decoding	17.3	15.7

6.2. English-French

For the English-French task, RWTH employed both phrase-based decoders (SCSS, PBT), different hierarchical phrase-based systems (HPBT) and a system combination of the best setups. All experimental results are given in Table 6.

The SCSS baseline system is trained on the in-domain data (TED) [21]. For this baseline, we achieve the biggest improvement by training an additional translation model on the available out-of-domain data (+1.1% BLEU). The system is further improved by applying part-of-speech-based adjective reordering rules as preprocessing step [22] (+0.3% BLEU) and a 7-gram word class language model (+0.3% BLEU).

For the PBT setups, the baseline is a system trained on all available data (allData). By adding phrase-level discriminative word lexicons [14] (DWL) and a reordering model, which distinguishes monotone, swap, and discontinuous phrase orientations [23, 24] (MSD-RO), the baseline system is improved by 0.9 points in BLEU and 0.7 points in TER.

The HPBT baseline is trained on the in-domain data. By limiting the recursion depth for the hierarchical rules with a shallow-1 grammar [25], we achieve an improvement of 0.6 points in BLEU. The bigger language model is trained on the target part of the bilingual corpus, the Shuffled News data and the 10^9 and French Gigaword corpora. As for the SCSS system, we trained an additional phrase table on the out-of-domain data. All in all, we are able to improve the HPBT baseline by +2.3% BLEU and -1.8% TER.

To increase the translation quality further, we employed system combination as described in Section 5 on several systems including the last year's primary submission (HPBT.2011). We gain an enhancement of 0.9 points in BLEU and 0.7 points in TER compared to the best single system. Compared to the last year's submission on the 2011 evaluation set, we could improve our best single system by 1.6 points in BLEU and 1.8 points in TER and further 1.0% BLEU with system combination (Table 7).

6.2.1. Google Books n-grams

For the English-French translation task we also investigated upon using the Google Books n-grams [26] which is a collection of n-gram counts extracted from digitized books. These counts are categorized by language and publication year of the books containing the n-grams. Selecting a range of years and using the vanilla n-grams resulted in language models with very high perplexities: The preprocessing steps applied to the underlying corpus do not match the preprocessing used in our system. By adapting the vanilla n-grams reasonable perplexities were obtained. We could further improve the language model by selecting only n-grams from books published in the last few years.

Our final language model uses 4-grams obtained from the

Table 6: Results for the English-French MT task. The open-source phrase-based decoder (SCSS) is incrementally augmented with a second translation model trained on out-of-domain data (*oodDataTM*), adjective-reordering as preprocessing step (*adj-reordering*) and a word class language model (*WordClassLM*). The in-house phrase-based decoder (PBT) is trained on all available bilingual data (allData) and incrementally augmented with a discriminative word lexicon (*DWL*) and an additional reordering model (*MSD-RO*). The hierarchical phrase-based decoder (HPBT) is incrementally augmented with a shallow-1 grammar (*shallow*), a bigger language model (*bigLM*), an alternative lexical smoothing (*IBM-1*), forced derivation (*FD*) and a second translation model trained on out-of-domain data (*oodDataTM*). The primary submission is a system combination of all systems marked with *.

system	**dev2010**		**tst2010**		
	BLEU	TER	BLEU	TER	
SCSS TED	25.9	58.3	29.3	52.1	
+oodDataTM	28.2	56.1	31.4	50.9	
+adj-reordering	28.2	56.4	31.7	50.5	*
+WordClassLM	28.3	56.0	32.0	50.1	*
PBT allData	27.9	55.8	30.9	50.6	*
+DWL	28.0	56.1	31.6	50.3	*
+MSD-RO	28.1	55.8	31.8	49.9	*
HPBT TED	25.7	58.6	29.0	52.8	
+shallow	26.6	57.8	29.6	52.0	
+bigLM	26.8	57.6	30.2	51.7	
+IBM-1	27.4	56.9	30.4	51.2	*
+FD	27.6	56.6	30.5	51.3	*
+oodDataTM	27.7	56.5	31.3	51.0	*
HPBT.2011	27.4	57.0	31.1	50.7	*
system combination	29.5	54.9	32.9	49.2	

Table 7: Comparison of 2011 and 2012 English-French task submission on tst2011.

submission	**tst2011**	
	BLEU	TER
2011 (single system)	36.1	43.8
2012 (best single system)	37.7	42.0
2012 (system combination)	38.7	40.9

Google Books n-grams which are mixed with our previously described language model. The resulting language model has a perplexity of 81.4 on our development set which compares to a perplexity of 85.0 of the original language model. However, we did not use the improved language model in our final system since very small to no increase in translation quality was observed whereas the language model size was increased. We believe that the combination of mismatch in preprocessing, OCR errors and the very broad domain of the Google Books n-grams lead to the rather small improvements. It should be noted that a newer version of the Google Books n-grams [27] is available that was not available during the time of work.

6.3. Arabic-English

RWTH participated last year in the Arabic-English TED task, achieving the best automatic results in the evaluation. This year, the architecture of the Arabic-English system is similar to last year, where a system combination is performed over different systems with differing Arabic segmentation methods. The differences from last year include: larger bilingual in-domain training data (130K versus 90K last year), the inclusion of the English Gigaword for language-modeling, and phrase table interpolation. We experimented with linear phrase table interpolation, where the phrase probabilities in both directions are interpolated linearly with a fixed weight optimized on the development set. We created two phrase tables, one using the TED in-domain and the other using the UN corpus, and interpolated them with a weight of 0.9 for the TED phrase table. The interpolation resulted in 1% BLEU improvement over a system using a phrase table trained over the full data.

The different segmentation methods are similar to last year, and include:

FST A finite state transducer-based approach introduced and implemented by [28]. The segmentation rules are encoded within an FST framework.

SVM A reimplementation of [29], where an SVM framework is used to classify each character whether it marks the beginning of a new segment or not.

CRF An implementation of a CRF classifier similar to the SVM counterpart. We use CRF++[2] to implement the method.

MorphTagger An HMM-based Part-Of-Speech (POS) tagger implemented upon the SRILM toolkit [30].

MADA v3.1 An off-the-shelf tool for Arabic segmentation [31]. We use the following schemes: D1,D2,D3 and ATB (TB), which differ by the granularity of the segmentation.

[2]http://crfpp.sourceforge.net/

Table 8: Arabic-English results on the test set (tst2010) for different segmentations, comparing 2011 and 2012 systems. *MADA-TB ALL* is a system using unfiltered bilingual data. The primary submission is a system combination of all the listed systems.

system	**2011**		**2012**	
	BLEU	TER	BLEU	TER
FST	25.1	57.0	26.5	55.8
SVM	25.4	57.4	26.6	54.4
HMM	25.7	56.9	26.9	55.1
CRF	25.7	56.7	26.9	54.5
MADA-D1	24.7	57.1	26.3	55.4
MADA-D2	25.2	57.1	26.9	54.7
MADA-D3	25.4	57.1	27.0	54.0
MADA-TB	26.1	56.4	-	-
MADA-TB ALL	26.1	56.6	27.1	54.4
system combination	27.0	54.7	28.0	53.4

As in last year, adaptation using filtering is done for both LM training and TM training. To build the LM, we use a mixture of all available English corpora, where News Shuffle, giga-fren.en and the English Gigaword are filtered. For translation model filtering, we use the combined IBM-1 and LM cross-entropy scores. We perform filtering for the MultiUN corpus, selecting $\frac{1}{16}$ of the sentences (400K). Due to the different Arabic segmentations we utilize, we performed the sentence selection only once over the MADA-TB method, and used the same selection for all other setups.

We trained phrase-based systems for all different segmentation schemes using the interpolation of TED and the 400K selected portion of the UN corpus. Additionally, one system was trained on all available data, preprocessed with MADA-TB. The results are summarized in Table 8. The table includes a comparison between the 2011 and 2012 systems on the test set. This year systems clearly improves over last year, with improvements ranging from 1% up-to 1.7% BLEU. The single system *MADA-TB ALL* of 2012 performs similarly to the system-combination submission of 2011. The final system combination improves over last year submission with +1% BLEU and -1.3% TER.

6.4. Chinese-English

Results of Chinese-English systems are given in Table 9. The system combination in Table 9 is RWTH's primary submission. The system combination was done as follows. We use both a phrase-based decoder [7] and a hierarchical phrase-based decoder Jane [11]. For each of the two decoders we do a bi-directional translation, which means the system performs standard direction decoding (left-to-right) and reverse direction decoding (right-to-left). We thereby obtain a total of four different translations.

Table 9: Chinese-English results on the dev test set for different segmentations. The primary submission is a system combination of all the listed systems.

system	dev2010		tst2010	
	BLEU	TER	BLEU	TER
PBT	12.2	80.0	14.2	73.7
PBT-reverse	11.9	79.6	13.7	74.3
HPBT	12.7	80.0	14.7	74.5
HPBT-reverse	12.8	81.0	14.5	76.2
HPBT-withUN-a	12.1	81.4	14.1	76.0
HPBT-withUN-b	12.5	80.4	14.0	75.5
system combination	13.7	78.9	15.4	74.1

To build the reverse direction system, we used exactly the same data as the standard direction system and simply reversed the word order of the bilingual corpora. For example, the bilingual sentence pair "今天␣是␣星期天␣。 ||Today␣is␣Sunday␣." is now transformed to "。␣星期天␣是␣今天||.␣Sunday␣is␣Today". With the reversed corpora, we then trained the alignment, the language model and our translation systems in the exactly same way as the normal direction system. For decoding, the test corpus is also reversed. The idea of utilizing right-to-left decoding has been proposed by [32] and [33] where they try to combine the advantages of both of the left-to-right and right-to-left decoding with a bidirectional decoding method. We also try to gain benefits from two-direction decoding, however, we use a system combination to achieve this goal.

In Table 9, first four systems do not use UN data. For *HPBT-withUN-a* and *HPBT-withUN-b* we additionally select 800k bilingual sentences from UN. *HPBT-withUN-a* and *HPBT-withUN-b* are built using the same setup but with differently optimized feature weights. PBT-reverse is the reverse system of PBT. HPBT-reverse is the reverse system of HPBT. *HPBT-withUN-a* and *HPBT-withUN-b* are trained with normal the left-to-right direction. From the results we draw the conclusions: HPBT performs better than PBT; UN data does not help; system combination of the six systems gets the best result.

6.5. German-English

For the German-English task, RWTH submitted a phrase-based system which is extended by several state-of-the-art improvements. In a preprocessing step, the German source is decompounded [34] and part-of-speech-based long-range verb reordering rules [22] are applied. The baseline uses a 4-gram language model trained on the target side of the bilingual data. When using additional monolingual data, we perform data selection as described in [35].

The results are given in Table 10. We created two baselines, one trained on all available bilingual data, one trained on the in-domain TED data only. The pure in-domain system clearly outperforms the general system on the TED data sets. This baseline is improved by forced-alignment phrase training (+0.1% BLEU) [18], adding $\frac{1}{4}$ of the Shuffled News data (+0.7% BLEU), a 7-gram word class language model (+0.6% BLEU), a second translation model trained on all available out-of-domain data (+0.5% BLEU) and finally by adding $\frac{1}{8}$ of each of the 10^9 and Gigaword corpora to the LM training data (+0.5% BLEU).

Table 10: Results for the German-English MT task. The phrase-based decoder (SCSS) trained on TED data is incrementally augmented with forced alignment phrase training (FA), additional monolingual data (ShuffledNews, Gigaword), a word class language model (WordClassLM) and a second translation model trained on out-of-domain data (oodDataTM).

system	dev2010		tst2010	
	BLEU	TER	BLEU	TER
SCSS allData	29.0	49.5	27.5	51.6
SCSS TED	29.9	48.4	28.4	50.3
+FA	30.3	47.7	28.5	49.9
+ShuffledNews	31.1	47.9	29.2	50.2
+WordClassLM	31.2	47.8	29.8	49.7
+oodDataTM	31.9	47.4	30.3	49.3
+Gigaword	32.6	46.4	30.8	48.6

6.6. Spoken Language Translation (SLT)

The input for the translation systems in the SLT track is the automatic transcription provided by the automatic speech recognition track. In this work, we used the recognitions of our ASR system described in Section 2. Due to the fact that the output of the ASR system does not provide punctuation marks or case information and contains recognition errors, we have to adapt the standard text translation system used in the English-French MT track.

Firstly, as described in [36], we trained a translation system on data without punctuation marks and case information in the source language, but including punctuation and casing in the target language. By translating ASR output with such a system, punctuation and case information are predicted during the translation process. We denote this as IMPLICIT.

As a second approach an SMT system was trained on a corpus with ASR output as source language data and the corresponding manual transcription as target language data, i.e. we interpret the postprocessing of the ASR output as machine translation [37]. We denote this as POSTPROCESSING. In order to built such a corpus we recognized the provided talks with our ASR system. On this corpus a standard phrase-based SMT was trained. During the translation of the ASR output punctuation and case information are restored. The output of this SMT system is the input of a standard text

translation system.

Table 11: Comparison between the methods IMPLICIT and POSTPROCESSING on the SLT task English-French (IWSLT 2012).

system	dev2010		tst2010	
	BLEU	TER	BLEU	TER
IMPLICIT	19.2	67.8	22.5	61.6
POSTPROCESSING	20.1	67.2	23.4	60.7

In Table 11, we compare the IMPLICIT method with our second approach (POSTPROCESSING). Note, for the experiments we utilized the best single system of the MT English-French track. POSTPROCESSING outperforms IMPLICIT and we achieve an improvement of 0.9 points in BLEU and 0.9 points in TER.

7. Conclusion

RWTH participated in ASR, MT (English-French, Arabic-English, Chinese-English, German-English) and SLT tracks of the IWSLT 2012 evaluation campaign.

Considerable improvements over respective baseline systems were achieved by applying several different techniques.

For the MT track, among these are phrase training for the phrase-based as well as for the hierarchical system, an additional reordering model, word class language model, data filtering techniques, phrase table interpolation, and different Arabic and Chinese segmentation tools. To improve the SLT system, postprocessing of the ASR output is modelled as machine translation. By system combination, additional improvements of the best single system were achieved.

8. Acknowledgements

This work was partly achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

9. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. of the International Workshop on Spoken Language Translation*, Hong Kong, HK, December 2012.

[2] M. Sundermeyer, M. Nußbaum-Thom, S. Wiesler, C. Plahl, A. El-Desoky Mousa, S. Hahn, D. Nolden, R. Schlüter, and H. Ney, "The RWTH 2010 Quaero ASR evaluation system for English, French, and German," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP*, 2011, pp. 2212–2215.

[3] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[4] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *Proc. of the Int. Conf. on Speech and Language Processing (ICSLP)*, vol. 2, Denver, CO, Sept. 2002, pp. 901–904.

[5] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, July 2002, pp. 311–318.

[6] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA, August 2006, pp. 223–231.

[7] R. Zens and H. Ney, "Improvements in Dynamic Programming Beam Search for Phrase-based Statistical Machine Translation," in *International Workshop on Spoken Language Translation*, Honolulu, Hawaii, Oct. 2008, pp. 195–205.

[8] J. Wuebker, M. Huck, S. Peitz, M. Nuhn, M. Freitag, J.-T. Peter, S. Mansour, and H. Ney, "Jane 2: Open source phrase-based and hierarchical statistical machine translation," in *International Conference on Computational Linguistics*, Mumbai, India, Dec. 2012, to appear.

[9] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003, pp. 160–167.

[10] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, pp. 308–313, 1965.

[11] D. Vilar, D. Stein, M. Huck, and H. Ney, "Jane: Open source hierarchical translation, extended with reordering and lexicon models," in *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July 2010, pp. 262–270.

[12] D. Chiang, "Hierarchical Phrase-Based Translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[13] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, June 1993.

[14] A. Mauser, S. Hasan, and H. Ney, "Extending statistical machine translation with discriminative and trigger-based lexicon models," in *Conference on Empirical Methods in Natural Language Processing*, Singapore, Aug. 2009, pp. 210–217.

[15] R. Zens and H. Ney, "Discriminative Reordering Models for Statistical Machine Translation," in *Human Language Technology Conf. (HLT-NAACL): Proc. Workshop on Statistical Machine Translation*, New York City, NY, June 2006, pp. 55–63.

[16] D. Stein, S. Peitz, D. Vilar, and H. Ney, "A Cocktail of Deep Syntactic Features for Hierarchical Machine Translation," in *Conf. of the Association for Machine Translation in the Americas (AMTA)*, Denver, CO, Oct./Nov. 2010.

[17] L. Huang and D. Chiang, "Forest Rescoring: Faster Decoding with Integrated Language Models," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, June 2007, pp. 144–151.

[18] J. Wuebker, A. Mauser, and H. Ney, "Training phrase translation models with leaving-one-out," in *Proceedings of the 48th Annual Meeting of the Assoc. for Computational Linguistics*, Uppsala, Sweden, July 2010, pp. 475–484.

[19] S. Peitz, A. Mauser, J. Wuebker, and H. Ney, "Forced derivations for hierarchical machine translation," in *International Conference on Computational Linguistics*, Mumbai, India, Dec. 2012, to appear.

[20] A. Lavie and A. Agarwal, "METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments," Prague, Czech Republic, June 2007, pp. 228–231.

[21] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[22] M. Popović and H. Ney, "POS-based Word Reorderings for Statistical Machine Translation," in *International Conference on Language Resources and Evaluation*, 2006, pp. 1278–1283.

[23] C. Tillmann, "A Unigram Orientation Model for Statistical Machine Translation," in *Proc. of the HLT-NAACL: Short Papers*, 2004, pp. 101–104.

[24] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantine, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," Prague, Czech Republic, June 2007, pp. 177–180.

[25] A. de Gispert, G. Iglesias, G. Blackwood, E. R. Banga, and W. Byrne, "Hierarchical Phrase-Based Translation with Weighted Finite-State Transducers and Shallow-n Grammars," *Computational Linguistics*, vol. 36, no. 3, pp. 505—-533, 2010.

[26] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, T. G. B. Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. Nowak, and E. Lieberman-Aiden, "Quantitative analysis of culture using millions of digitized books," *Science*, vol. 331, pp. 176–182, 2011.

[27] Y. Lin, J.-B. Michel, E. Aiden Lieberman, J. Orwant, W. Brockman, and S. Petrov, "Syntactic annotations for the google books ngram corpus," in *Proceedings of the ACL 2012 System Demonstrations*. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 169–174.

[28] A. El Isbihani, S. Khadivi, O. Bender, and H. Ney, "Morpho-syntactic Arabic Preprocessing for Arabic to English Statistical Machine Translation," in *Proceedings on the Workshop on Statistical Machine Translation*, New York City, June 2006, pp. 15–22.

[29] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks," in *HLT-NAACL 2004: Short Papers*, D. M. S. Dumais and S. Roukos, Eds., Boston, Massachusetts, USA, May 2 - May 7 2004, pp. 149–152.

[30] S. Mansour, "Morphtagger: Hmm-based arabic segmentation for statistical machine translation," in *International Workshop on Spoken Language Translation*, Paris, France, December 2010, pp. 321–327.

[31] R. Roth, O. Rambow, N. Habash, M. Diab, and C. Rudin, "Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking," in *Proceedings of ACL-08: HLT, Short Papers*, Columbus, Ohio, June 2008, pp. 117–120.

[32] T. Watanabe and E. Sumita, "Bidirectional decoding for statistical machine translation," in *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, ser. COLING '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 1–7.

[33] A. Finch and E. Sumita, "Bidirectional phrase-based statistical machine translation," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, ser. EMNLP '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 1124–1132.

[34] P. Koehn and K. Knight, "Empirical Methods for Compound Splitting," in *Proceedings of European Chapter of the ACL (EACL 2009)*, 2003, pp. 187–194.

[35] R. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *ACL (Short Papers)*, Uppsala, Sweden, July 2010, pp. 220–224.

[36] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *International Workshop on Spoken Language Translation*, Kyoto, Japan, Nov. 2006, pp. 158–165.

[37] S. Peitz, S. Wiesler, M. Nussbaum-Thom, and H. Ney, "Spoken language translation using automatically transcribed text in training," in *International Workshop on Spoken Language Translation*, Hongkong, Dec. 2012, to appear.

The HIT-LTRC Machine Translation System for IWSLT 2012

Xiaoning Zhu, Yiming Cui, Conghui Zhu, Tiejun Zhao, Hailong Cao

Language Technology Research Center
Harbin Institute of Technology, China
{xnzhu,ymcui,chzhu,tjzhao,hailong}@mtlab.hit.edu.cn

Abstract

In this paper, we describe HIT-LTRC's participation in the IWSLT 2012 evaluation campaign. In this year, we took part in the Olympics Task which required the participants to translate Chinese to English with limited data.

Our system is based on Moses[1], which is an open source machine translation system. We mainly used the phrase-based models to carry out our experiments, and factored-based models were also performed in comparison. All the involved tools are freely available.

In the evaluation campaign, we focus on data selection, phrase extraction method comparison and phrase table combination.

1. Introduction

This paper describes the Statistical Machine Translation (SMT) system explored by the Language Technology Research Center of Harbin Institute of Technology (HIT-LTRC) for IWSLT 2012. Generally, our system was based on Moses, and phrase-based models were used.

In Olympics shared task, the training data was limited to the supplied data including HIT Olympic Bilingual Corpus (HIT)[2] and Basic Travel Expression Corpus (BTEC)[3]. Although the two corpora are both oral corpus, there are still some differences between them. For example, the BTEC corpus is travel-related, and the HIT corpus is mainly about the Olympic Games. Besides this, the organizer of IWSLT 2012 also provided two development sets which are selected from the HIT and BTEC corpus respectively. Because the training data is limited by the above corpus, in order to get a better performance, we need to excavate all the potential of the two corpora, including the development sets.

One key problem of the SMT system is how to extract the phrase. Giza++[4] is a popular word alignment tool which can produce word alignment information with parallel corpus. By using heuristic phrase extraction method, we can extract phrases with the alignment. Compared with heuristic phrase extraction method, Pialign[5] is an unsupervised model for joint phrase alignment and extraction using nonparametric Bayesian methods and inversion transduction grammars (ITGs). We compared the phrase table extracted by the two phrase extraction methods in many ways, such as the size, the quality, and the differences of two methods.

System combination has been approved to improve machine translation performance significantly. With several machine translation systems' outputs, researchers can get a better translation by combining the outputs. But in this paper, we didn't combine the outputs; instead we combine the models generated by Giza++ and Pialign. It is shown that we can get a better performance by model combination.

The following of the paper is organized as follows. Section 2 describes a phrase-based machine translation system which was used in our work. In section 3, we compared differences of two corpora. The result and phrase extraction are discussed in section 4. And in the last section, we give a conclusion and discuss the future work.

2. Phrase-based System

Our primary system is based on Moses with a phrase-based model. Under the log-linear framework[6], when given a source sentence f , we can get a translation e as follows:

$$p(e \mid f;\lambda) = \frac{\exp(\lambda \bullet h(f,e))}{Z(\lambda)}$$

with

$$Z(\lambda) = \sum \exp(\lambda \bullet h(f,e))$$

where $h(f,e)$ denotes the feature vector of the pair (f,e), and λ is its corresponding weight vector. $h(f,e)$ contains 14 features and they are divided into following categories:

- Bidirectional translation probabilities;
- Bidirectional lexical translation probabilities;
- MSD-reordering model;
- Distortion model;
- Language model;
- Word penalty;
- Phrase penalty.

2.1. Pre-processing

The Chinese sentences supplied by the organizer were not segmented, so we used the Stanford Word Segmenter[7] to segment the Chinese sentences with the PKU model. The English sentences were not tokenized, thus we used the open source tools supplied by Moses to tokenize them. We also lowercased all the English data for training. There are many English punctuation characters in Chinese sentences (and vice versa), so we wrote some scripts to change all the punctuation characters in order.

2.2. Training

In the training step, we used Giza++ to get alignments and combined the alignments with *grow-diag-final-and* method. With the alignments, we can extract phrases with heuristic phrase extraction method and generate the translation model. Besides, we also used Pialign to generate the alignments and phrases.

The language model was built with SRILM toolkit[8]. A 5-gram language model was used for decoding. The corpus we used to build the language model is all the supplied data, including training data and development data.

2.3. Decoder

The decoder used in our system is Moses.

2.4. Tuning

The parameters were tuned on the development set with standard trainer MERT[9]. When running MERT, the k-best-list-size was set as 100 and BLEU4[10] was selected as the evaluation metric.

2.5. Post-processing

The translations were post-processed after decoding.

- All the Chinese words in output were deleted. Because there are many names in the test set, and most of them can't be translated, so we deleted them;
- The English sentences were de-tokenized ;
- The English sentences were re-cased by the recaser tools provided by Moses.

3. Corpus

The IWSLT organizer provided two training corpus, including HIT corpus and BTEC corpus. HIT corpus is a multilingual oral corpus developed for the Beijing 2008 Olympic Games. There are five domains in HIT corpus, including traveling, dining, sports, traffic and business. The BTEC corpus is also an oral corpus containing tourism-related sentences. Besides the training corpus, they also provided two development corpus, which were extracted from the HIT corpus and BTEC corpus. In the following paper, we use HIT_train, HIT_dev, BTEC_train, BTEC_dev to denote four corpora respectively.

In our system, we used HIT_train, BTEC_train, BTEC_dev, HIT_dev as our training data. And HIT_dev was also used as our development set. We also random sampled 1000 sentences from HIT corpus as our test set.

The detail of the corpus is presented in Table 1.

Table 1: Corpus

	BTEC	HIT
Train	19975	52603
Dev	2977	2057
Total	22949	54660

We combined the four corpora as training data, and the new generated corpus is shown in Table 2.

Table 2: Training data

name	corpus	#
Corpus 1	BTEC_train+HIT_train	72575
Corpus 2	Corpus1+BTEC_dev	75552
Corpus 3	Corpus2+HIT_dev	77609

4. Experiments and Results

4.1. The comparison of Giza++ and Pialign

We first trained six models with Giza++ alignments and Pialign alignments. A comparison between the phrase table generated from Giza++ and Pialign is shown in Table 3. Table 4 shows the covering of the six phrase tables of the test set.

Table 3: Comparison between Giza++ and Pialign

Corpus	align	total	common	different
1	Giza++	1182913	409443	773470
	Pialign	1385520		976077
2	Giza++	1208128	418788	789340
	Pialign	1413367		994579
3	Giza++	1236688	428377	808306
	Pialign	1445577		1017200

Table 4: Covering of test set

Corpus	align	Chinese	English
1	Giza++	21.7%	36.0%
	Pialign	23.6%	38.3%
2	Giza++	21.7%	36.1%
	Pialign	23.8%	38.7%
3	Giza++	21.9%	36.6%
	Pialign	23.9%	38.9%

In Table 3, we showed the total number of phrase pairs, the common phrase pairs of Giza++ and Pialign, the different phrase pairs of Giza++ and Pialign. In Table 4, we show the covering capacity of the phrase table. The covering capacity c is defined as follows:

$$c = \frac{\#\text{ of phrases both in test set and in phrase table}}{\#\text{ of phrases in test set}}$$

To note that, the test set was divided into unigram to 5-gram phrases.

From Table 3 we can find that the phrase table generated by Pialign is a little bigger than Giza++. Because we use *-samps* parameters to sample the bilingual parser tree repeatedly. In this experiment, we tuned this parameters from 1(default) to 80. At first, with the increment of the phrase table size, the performance grows at the same time. But after 20^{th} sampling, the bias of sampling adds too many noise phrase pairs. Finally, we set this value to 20. With default value, Pialign only generated 389,982 phrase pairs (32.28% as the Giza++ did), but the performances are still comparable. With the covering capacity, we can estimate the performance of the model. The result is the same with the translation result, which shows that Pialign is better than Giza++ in phrase extraction.

4.2. Results of translation

The result of translation outputs are shown in Table 5 and Table 6.

The result is confusing. After we tuned the parameters with HIT_dev, the result became worse. This may be caused by the mismatch between HIT_dev and HIT_train. The result also shows that although we continue to enlarge the size of

training data, the BLEU score may reduce on the contrary. These remind us that the model is also important.

Table 5: Result without tuning

Corpus	align	BLEU%
1	Giza++	20.76
	Pialign	20.80
2	Giza++	20.62
	Pialign	21.20
3	Giza++	20.51
	Pialign	20.54

Table 6: Result with tuning

Corpus	align	BLEU%
1	Giza++	19.97
	Pialign	19.70
2	Giza++	18.40
	Pialign	19.66
3	Giza++	15.52
	Pialign	15.10

4.3. Combination of two phrase table

We explored Giza++ and Pialign to extract phrases. In this section, we want to combine the two methods by merging two phrase tables using a linear interpolation method. For Giza++, the best result was achieved when we used Corpus1. For Pialign, the best result was achieved when we used Corpus2. So we combined the two phrase tables. The result without tuning is shown in Table 7. The parameter means the weight of Pialign.

Table 7: Phrase Table Combination

parameter	BLEU%
0.4	20.69
0.5	20.78
0.6	20.62

Compared with Table 7 and Table 5, we can draw a conclusion that phrase table combination can improve the performance of machine translation systems a little. Maybe due to the size of the training data, the result is not very clear to see the increment. And our combination method is only a linear interpolation method, which is naive for phrase table combination. We believe that a more complex strategy, such as some machine learning algorithms can improve the phrase table combination results.

4.4. Linguistic knowledge

In recently years, many researchers have focused on how to integrate linguistic knowledge into machine translation systems. In this work, part of speech was introduced to improve the machine translation systems. We used Stanford Log-linear Part-Of-Speech Tagger[11] to get the POS tag. Factored-based model of Moses was used to train a translation model. The result is shown in Table 8.

Table 8: Linguistic features

system	With tuning	Without tuning
baseline	19.97	20.76
+pos tag	18.53	16.63

As we can see that the result with POS tag is also not better than the baseline. We think that linguistic knowledge is a good research field to improve machine translation performance.

4.5. Official Results

We took part in the Olympics task(OLY)[12], and the final translations we submitted was generated by Pialign with corpus 2. And because of the bad performance of tuning, we submit out results without tuning. The final result was shown in Table 9.

Table 9: Official results in BLEU

system	case+punc	no_case+no_punc
Pialign-2	19.10	18.76

5. Conclusions and Future Work

In this paper, we explained our work in the IWSLT 2012 evaluation campaign. We compared two phrase extraction methods and tried to combine the two methods. The results show that the combination method can improve the result of MT systems.

In future, we will still try to study some other advanced combination methods to modify our system.

6. Acknowledgements

The work of this paper is funded by the project of National Natural Science Foundation of China (No. 61100093) and the project of National High Technology Research and Development Program of China (863 Program) (No. 2011AA01A207).

7. References

[1] P. Koehn et al., “Moses: Open Source Toolkit for Statistical Machine Translation”, in Proceedings of the ACL Demo and Poster Sessions, Prague, Czech Republic, 2007, pp.177–180.

[2] Muyun Yang, Hongfei Jiang, Tiejun Zhao and Sheng Li, “Construct Trilingual Parallel Corpus on Demand”, *Chinese Spoken Language Processing*, vol. 4274, pp. 760-767, 2006

[3] Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. “Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversation in the Real World”, in Proceedings of LREC 2002, Las Palmas, Spain, 2002

[4] Franz Josef Och, Hermann Ney. "A Systematic Comparison of Various Statistical Alignment Models", *Computational Linguistics*, volume 29, number 1, pp. 19-51 March 2003.

[5] G. Neubig, T. Watanabe, E. Sumita, S. Mori, and T. Kawahara, “An unsupervised model for joint phrase

alignment and extraction," in Proceedings of ACL, 2011, pp. 632–641.

[6] Franz Josef Och and Hermann Ney. 2002. "Discriminative training and maximum entropy models for statistical machine translation", in Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, pages 295–302, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

[7] Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter. In Fourth SIGHAN Workshop on Chinese Language Processing.

[8] Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In Proceedings of the International Conference on Spoken Language Processing, volume 2, pages 901–904.

[9] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, pages160–167, Morristown, NJ, USA. Association for Computational Linguistics.

[10] Kishore Papineni, Salim Roukos, Todd Ward, andWei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

[11] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In Proceedings of HLT-NAACL, pages 252–259.

[12] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, S. Stüker. Overview of the IWSLT 2012 Evaluation Campaign, In Proc. of IWSLT, Hong Kong, HK, 2012.

FBK @ IWSLT 2012 - ASR track

D. Falavigna, R. Gretter, F. Brugnara, D. Giuliani

HLT research unit, FBK, 38123 Povo (TN), Italy
(falavi,gretter,brugnara,giuliani)@fbk.eu

Abstract

This paper reports on the participation of FBK at the IWSLT2012 evaluation campaign on automatic speech recognition: namely in the English ASR track. Both primary and contrastive submissions have been sent for evaluation.

The ASR system features acoustic models trained on a portion of the TED talk recordings that was automatically selected according to the fidelity of the provided transcriptions. Three decoding steps are performed interleaved by acoustic feature normalization and acoustic model adaptation.

A final rescoring step, based on the usage of an interpolated language model, is applied to word graphs generated in the third decoding step. For the primary submission, language models entering the interpolation are trained on both out-of-domain and in-domain text data, instead the contrastive submission uses both "general purpose" and auxiliary language models trained only on out-of-domain text data. Despite this fact, similar performance are obtained with the two submissions.

1. Introduction

The IWSLT 2012 Evaluation Campaign [1], similarly to the one carried out for IWSLT2011, focused on the automatic transcription/translation of TED Talks [1]: a collection of public speeches on a variety of topics.

This year, for automatic speech recognition (ASR) we mostly focused our work on language modeling, while acoustic models remained unchanged with respect to those used in the evaluation campaign of IWSLT2011. In particular, we propose a method for focusing the language models (LMs) used during a final linguistic rescoring of the word graphs produced by our ASR system, towards the ASR output of previous decoding stages, obtaining significant reduction in word error rate (WER) without the usage of in-domain text data. Although approaches similar to the one used for producing our contrastive submissions are also reported in the literature (see [2] and [3]), there are some substantial differences that make the method reported in this paper quite novel.

More specifically, we propose to apply an automatic selection procedure to the same texts employed to train the "general purpose" LMs used in the various decoding steps of the ASR system. Then, we use the set of selected documents to train auxiliary LMs which are linearly interpolated, on a talk specific basis, with the general ones in order to provide LM probabilities to a final decoding pass based on word-graphs rescoring. In this way, we are able to train LMs focused on the ASR output. We prefer to use the term "LM focusing", instead of LM adaptation, to underline the fact that we are not using new data to train auxiliary LMs but, on the contrary, a subset of existing text data is enhanced in order to better match the linguistic content of the audio to transcribe.

Since we want, in principle, to "frequently" focus LMs using the ASR output corresponding to a given (or automatically detected) segmentation of the audio stream to transcribe, we developed a technique that allows to efficiently select a subset of documents from the large set of available documents. This latter technique is based on a vector space model: each document is represented with a vector of coefficients, while a metric is defined that allows to estimate the distance between two vectors or, equivalently, the similarity between two documents. The "auxiliary" documents are hence obtained as the ones that are most similar to a given query document (i.e. to the ASR output of a piece of speech to transcribe).

The definition of the features and of the metrics have been inspired from TFxIDF (Term Frequency x Inverse Document Frequency) vector space model [4], however the employed features, the way adopted for storing them and the similarity metrics used, has allowed to improve both computation and memory efficiency with respect to TFxIDF approach.

2. Automatic transcription system

In this section we summarize the main features of the FBK primary system used in the IWSLT2012 Evaluation Campaign for transcribing TED talks delivered in English. For each talk, in addition to the audio file, time boundaries of speech segments to be transcribed are given. The word transcription of a talk is generated in three decoding passes. All the decoding passes make use of a 4-gram language model and are interleaved by acoustic feature normalization and Acoustic Model (AM) adaptation.

2.1. Acoustic data selection for training

For AM training, domain specific acoustic data were exploited. Recordings of TED talks released before the cut-off date, 31 December 2010, were downloaded with the corre-

[1] http://www.ted.com/talks

sponding subtitles which are content-only transcriptions of the speech. In content-only transcriptions anything irrelevant to the content is ignored, including most non-verbal sounds, false starts, repetitions, incomplete or revised sentences and superfluous speech by the speaker. A simple but robust procedure was implemented to select only audio data with an accurate transcription.

The collected data consisted in 820 talks, for a total duration of $\sim$216 hours, with $\sim$166 hours of actual speech. The provided subtitles are not a verbatim transcription of the speeches, hence the following procedure was applied to extract segments that can be deemed reliable. The approach is that of selecting only those portions in which the human transcription and an automatic transcription agree. To this end, a "background" 4-gram language model was first trained on all the talk transcriptions. Subsequently, a specific Language Model (LM) was built for each talk by adapting the language model to the human transcription of the talk. A preliminary automatic transcription was performed on the talks with a pre-trained general AM for English and the talk-specific LM. The output of the system was aligned with the reference transcriptions, and the matching segments were selected, resulting in an overlap of $\sim$120 hours of actual speech out of the total of 166. By using these segments together with the segments labeled as silence, a TED-specific acoustic model was trained, as detailed in the following section. The label/select/train procedure was repeated two more times, resulting in a portion of selected actual speech that grew to $\sim$142 hours and then to $\sim$144 hours. Given the modest improvement in the third iteration, the procedure was not repeated further. In conclusion, the method made available 87% of the training speech, which was considered satisfactory.

2.2. Acoustic model

Thirteen Mel-frequency cepstral coefficients, including the zero order coefficient, are computed every 10ms using a Hamming window of 20ms length. First, second and third order time derivatives are computed after segment-based cepstral mean subtraction to form 52-dimensional feature vectors. Acoustic features are normalized and HLDA-projected to obtain 39-dimensional feature vectors as described below.

AMs were trained exploiting a variant of the speaker adaptive training method based on Constrained Maximum Likelihood Linear Regression (CMLLR) [5]. In our training variant [6, 7, 8] there are two sets of AMs: the target models and the recognition models. For each cluster of speech segments, an affine transformation is estimated through CMLLR [5] with the aim of minimizing the mismatch between the cluster data and the target models. Once estimated, the affine transformation is applied to cluster data in order to normalize acoustic features with respect to the target models. Recognition models are then trained on the normalized data. Leveraging on the possibility that the structure of the target and recognition models can be determined independently, a Gaussian Mixture Model (GMM) can be adopted as the target model for training AMs used in the first decoding pass [6]. This has the advantage that, at recognition time, word transcriptions of test utterances are not required for estimating feature transformations. Instead, target models for training recognition models used in a second or third decoding pass are usually triphones with a single Gaussian per state [7]. In all cases, the same target models are used for estimating cluster-specific transformations during training and recognition.

In the current version of the system, a projection of the acoustic feature space based on Heteroscedastic Linear Discriminant Analysis (HLDA) is embedded in the feature extraction process as follows. A GMM with 1024 Gaussian components is first trained on an extended acoustic feature set consisting of static acoustic features plus their first, second and third order time derivatives. For each cluster of speech segments, a CMLLR transformation is then estimated w.r.t. the GMM and applied to acoustic observations. After normalizing the training data, an HLDA transformation is estimated w.r.t. a set of state-tied, cross-word, gender-independent triphone Hidden Markov Models (HMMs) with a single Gaussian per state, trained on the extended set of normalized features. The HLDA transformation is then applied to project the extended set of normalized features in a lower dimensional feature space, that is a 39-dimensional feature space. Recognition models used in the first and subsequent decoding passes are trained from scratch on normalized HLDA-projected features. HMMs for the first decoding pass are trained through a conventional maximum likelihood procedure. Recognition models used in the second or third decoding pass are speaker-adaptively trained, exploiting as target-models triphone HMMs with a single Gaussian density per state.

2.3. Lexica

Two different lexica were used to provide phonetic transcriptions of words:

- *USLex*: Pronunciations in the lexicon are based on a set of 45 phones. The lexicon was generated by merging different source lexica for American English (LIMSI '93, CMU dictionary, Pronlex). In addition, phonetic transcriptions for a number of missing words were generated by using the phonetic transcription module of the Festival speech synthesis system.

- *BEEPLex*: This lexicon was generated by exploiting the British English Example Pronunciations (BEEP) lexicon. Pronunciation models in this lexicon are based on a set of 44 phones. Transcription for a number of missing words were obtained by exploiting the pronunciation models in the *USLex* lexicon and mapping phonetic symbols into the BEEP phone set.

For each phone set and decoding pass, a set of state-tied, cross-word, gender-independent triphone HMMs were trained for recognition. Around 170,000 Gaussian densities, with diagonal covariance matrices, were allocated for each model set.

2.4. Language models

Text data used for training the LMs are those released for the IWSLT2012-SLT Evaluation Campaign. Before training LMs, texts were cleaned, normalized (punctuation was removed, numbers and dates were expanded) and double lines were removed. Then, they have been grouped into the following three sets, on which a corresponding LM was trained:

- **giga5** GIGAWORD 5-th edition. Contains documents stemming from seven distinct international sources of English newswire. It is released from the Linguistic Data Consortium (see http://www.ldc.upenn.edu/). In total it contains about 4G words.
- **wmt12** Formed by documents in WMT12 news crawl, news commentary v7 and Europarl v7 (see IWSLT2012 official web site for some more details about these corpora). In total it contains about 830M words.
- **ted12** An in-domain set of texts extracted from TED talks transcriptions used for training. It contains about 2.4M words.

For each of the three sources listed above, we trained a 4-gram backoff LM using the modified shift beta smoothing method as supplied by the IRSTLM toolkit [9]. The three LMs resulted, respectively, into about:

- **giga5** 128M bigrams, 231M 3-grams, 422M 4-grams;
- **wmt12** 44M bigrams, 50M 3-grams, 68M 4-grams;
- **ted12** 599K bigrams, 199K 3-grams, 125K 4-grams.

The **wmt12** LM is used to compile a static Finite State Network (FSN) which includes LM probabilities and lexicon for the first two decoding passes. The latter LM was pruned in order to obtain a network of manageable size, resulting in a recognition vocabulary of 200K words and into about: 42M bigrams, 34M 3-grams and 31M 4-grams.

The non-pruned LMs, **giga5** and **wmt12**, are instead linearly interpolated (as explained below) in order to provide LM probabilities for expanding word graphs to be used in the third decoding step.

2.5. Word graphs generation

Word graphs (WGs) are generated in the second decoding step. To do this, all of the word hypotheses that survive inside the trellis during the Viterbi beam search are saved in a word lattice containing the following information: initial word state in the trellis, final word state in the trellis, related time instants and word log-likelihood. From this data structure and given the LM used in the recognition steps, WGs are built with separate acoustic likelihood and LM probabilities associated to word transitions. To increase the recombination of paths inside the trellis and consequently the densities of the WGs, the so called word pair approximation [10] is applied. In this way the resulting graph error rate was estimated to be 8.8% on the development set used for IWSLT2012 evaluation campaign, less than $\frac{1}{2}$ of the corresponding WER (which resulted to be 18.9%, as reported in section 4).

2.6. Transcription process

In the IWSLT2012 ASR evaluation, time boundaries of speech segments to be transcribed are given for each audio file. These non-overlapping speech segments are clustered by using a method based on the Bayesian information criterion [11]. The resulting clustering is exploited by the transcription system to perform cluster-based acoustic feature normalization and AM adaptation.

The first decoding pass is carried out with acoustic models based on *BEEPlex*, while the second and third decoding passes make use of acoustic models based on *USLex*. This configuration was chosen based on preliminary experiments on development data. In addition, as previously seen, the **wmt12** LM has been used in both first and second decoding pass.

Cluster-based, text-independent acoustic feature normalization is first performed before HLDA projection. The output of the first decoding pass on these acoustic features is used as a supervision for conducting cluster-based CMLLR acoustic feature normalization and MLLR-based acoustic model adaptation [12] before the second decoding pass, where both the first-best output and word graphs are generated.

The search space employed in the third decoding pass is obtained after expansion of WGs produced in the second decoding pass. The LMs used for WG expansion is a combination of non pruned **giga5** and **wmt12** LMs.

The simplest way for combining LMs trained on different sources is to compute the probability of a word w, given its past history h, as:

$$P[w \mid h] = \sum_{j=1}^{j=J} \lambda_j P_j[w \mid h] \quad (1)$$

where $P_j[w \mid h]$ are LM probabilities trained on the j^{th} source, λ_j are weights estimated with the aim of minimizing the overall perplexity on a development set and J is the total number of LMs to combine. In this case, the development set on which weights λ_j are trained is the one given by the (second pass) ASR output of each TED talk. Note that, in this way, we estimate interpolation weights that depend on each given talk.

The expanded WGs are compiled into corresponding decoding networks using the *USLex* lexicon. Also in this case, the best recognition hypothesis generated in the second decoding pass is exploited for conducting cluster-based CMLLR acoustic feature normalization and MLLR-based acoustic model adaptation. Finally, WGs are again generated in the third decoding pass and successively rescored for providing both primary and contrastive submissions, as will be explained below.

2.7. Primary submission

WGs generated in the third decoding step are rescored using an interpolated LM that combine all of the three LMs described above, **giga5**, **wmt12** and the in-domain LM **ted12**. To do this, the original LM probability on each arc of each WG is substituted with the linearly interpolated probability given by equation 1. The development set used to train the interpolation weights is the ASR output of the third decoding step and, therefore, also in this case talk specific interpolation weights are estimated.

Note that in the latter WG based rescoring phase acoustic model probabilities associated to arcs of word graphs remain unchanged, i.e. a pure linguistic rescoring is implemented.

2.8. Contrastive submission

As mentioned in the introduction our contrastive submission involves the usage of focused LMs. Figure 1 shows a block diagram of the ASR system employing these LMs, emphasizing both the procedure for selecting auxiliary documents for LM training and the WG based rescoring pass.

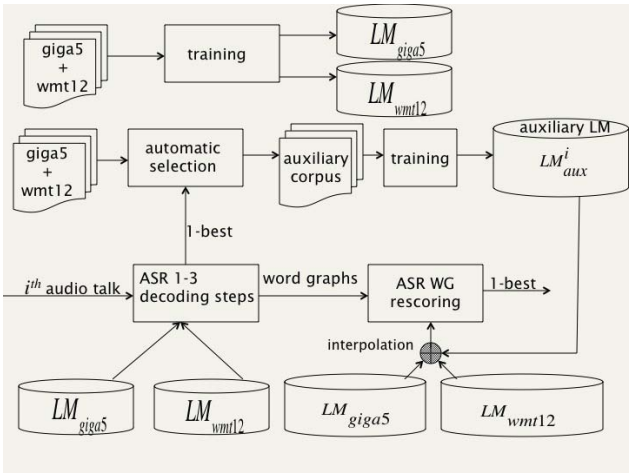


Figure 1: *Block diagram of the ASR system using focused LMs.*

The best word sequences generated in the third decoding pass are used to evaluate the baseline performance, as well as for selecting auxiliary documents. For each given i^{th} talk an auxiliary LM (LM^i_{aux}) is trained on data automatically selected from the out-of-domain text resources **giga5** and **wmt12**, with the selection method described below. The i^{th} query document used to score the out-of-domain text corpora consists of the 1-best output produced in the third ASR decoding step. Then, similarly to primary submission, the original LM probability on each arc of each WG is substituted with the probability given by the interpolation, using equation 1, of the three LMs: **giga5**, **wmt12** and LM^i_{aux}. Also in this case interpolation weights, $\lambda^i_j, 1 \leq j \leq 3$, associated to the three LMs are estimated so as to minimize the overall LM perplexity on the 1-best output (the same used to build the i^{th} query document), of the third ASR decoding step. For clarity reasons this latter procedure is not explicitly shown in Figure 1. The resulting WGs are rescored using the new interpolated LM probabilities.

Note that for this submission no LM trained on in-domain data is used in the last WG rescoring pass, actually the difference between contrastive and primary submission only relies on entering LM^i_{aux} instead of **ted12** in the LM probability interpolation.

3. Auxiliary data selection

In this section we describe the processes for selecting documents (rows in the corpus formed by **giga5** plus **wmt12** text resources) which are semantically similar to a given automatically transcribed document. In the following, N is the number of total rows in the corpus (about 42M for this work) and D is the total number of unique words in the corpus.

The result of this process is to obtain a sorted version of the whole corpus according to similarity scores. The most similar documents will be used to build talk-dependent auxiliary LMs.

3.1. Preprocessing stage

First, we build a table containing all the different words found in the corpus to select, each one with an associated counter of the related number of occurrences.

Then, a dictionary $\mathcal{V}$ is built containing the words that, according to inverse order of occurences, have an index $D^{''} \leq i \leq D^{'}$, where $D^{''} = 100$ and $D^{'} = 200,000$.

Then, every word in the corpus is replaced with its corresponding index in $\mathcal{V}$. Words outside $\mathcal{V}$ are discarded.

Indices of each row are then sorted to allow quick comparison (this point will be discussed later). The rationale behind this approach is the following:

- very common words only carry syntactic information, therefore they are useless if the purpose is to find semantically similar sentences;
- very uncommon words will be used rarely so they will just slow down the search process.

The choice for the reported values of D' and D'' has been done on the basis of preliminary experiments carried out on a development data set (see section 4) and did not result to be critical. With the chosen values about half of the words of the corpus were discarded: i.e. about 2.6M millions of indices survived. We keep alignment between the original corpus and its indexed version.

3.1.1. Searching stage

From the sequence of automatically recognized words $W^i = w^i_1, \ldots, w^i_{\texttt{len}(W^i)}$ of the given i^{th} query document (i.e. the i^{th} automatically transcribed talk) we derive a corresponding sequence of numerically sorted indices. Hence, both the i^{th} talk and the n^{th} document in the corpus are represented by two vectors (containing integer indices): $\mathbf{C}'^i$ and $\mathbf{R}'^n$, respectively. The similarity score is:

$$s'(\mathbf{C}'^i, \mathbf{R}'^n) = \frac{e(\mathbf{C}'^i, \mathbf{R}'^n)}{\mathtt{dim}(\mathbf{C}'^i) + \mathtt{dim}(\mathbf{R}'^n)} \quad (2)$$

where $e(\mathbf{C}'^i, \mathbf{R}'^n)$ is the number of common indices between the two vectors $\mathbf{C}'^i$ and $\mathbf{R}'^n$. Note that the two vectors $\mathbf{C}'^i$ and $\mathbf{R}'^n$ have dimensions exactly equal to the number of the corresponding indexed words survived after pruning of dictionary, as explained above.

The proposed approach is similar to the well known method based on TFxIDF [4]. However, while the latter allows to compare two documents by weighting same words both with their frequencies and with their relevance in the documents to select, the proposed approach is essentially a method to count the number of same words in the documents (word counters are not used in the similarity metric). However, since components of index vectors are numerically ordered, the computation of the similarity score $s'(\mathbf{C}'^i, \mathbf{R}'^n)$ results very efficient. This is essential given the large number of documents in the corpus to score.

In addition, differently from TFxIDF, the proposed approach doesn't require to load into memory of the computer any parameter related to the whole dictionary, instead only the sequence of indices (i.e. one sequence of integer values for each row in the corpus to select) entering equation 2 is needed. In our implementation the latter indices are conveniently stored and read from a file. Therefore, the memory requirements of the proposed approach are negligible. Furthermore, since the resulting document scores are not normalized, the estimate of the threshold to be used for selecting the subset of the documents to sort from the whole corpus is based on a preliminary computation of a histogram of scores.

Finally, in order to measure the complexities of proposed method and TFxIDF based one, we led three different selection runs using ASR output of a predefined TED talk. For processing the whole **giga5** + **wmt12** corpus the proposed method took on average about $16min$, with a memory occupation of about 10MB, while the TFxIDF based method took on average about $114min$, with a memory occupation of about 650MB. These runs were carried out on the same Intel/Xeon E5420 machine, free from other computation loads.

A more detailed comparison among: the proposed selection approach, the TFxIDF based one and another one based on perplexity minimization is reported in a companion paper.

4. System run

In order to tune some parameters of our automatic transcription system we carried out some preliminary experiments on the development set of IWSLT2012 evaluation campaign. The latter is made by 19 TED talks derived from the union of the IWSLT 2010 development and evaluation sets. In particular, we need to choose, for the contrastive submission, an optimal number of words on which to train auxiliary LMs as explained in section 3. To do this we evaluated, on the above mentioned dev set, both perplexity (PP) and WER as functions of the latter number of words. Results are given in Figures 2 and 3.

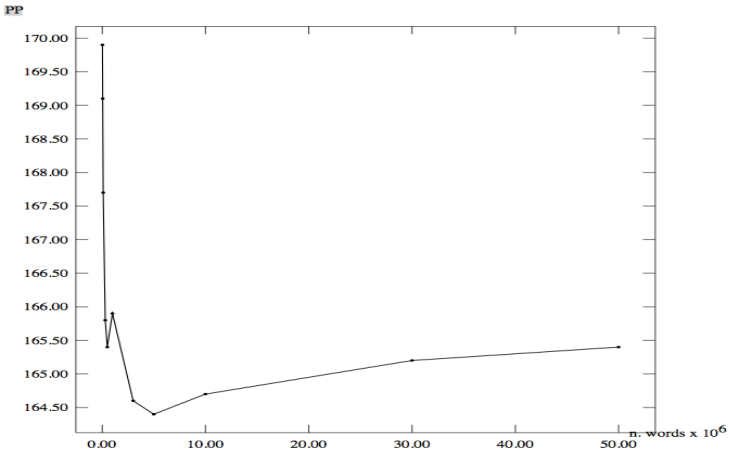


Figure 2: *Perplexity on dev set of focused LMs, as a function of the number of words used to train auxiliary LMs (the point corresponding to 0 words on the abscissa refers to the usage of the baseline LM).*

In the figures the point corresponding to 0 words on the abscissa indicates performance obtained with the baseline LM, i.e through the interpolation of **giga5** and **wmt12** without including auxiliary LMs.

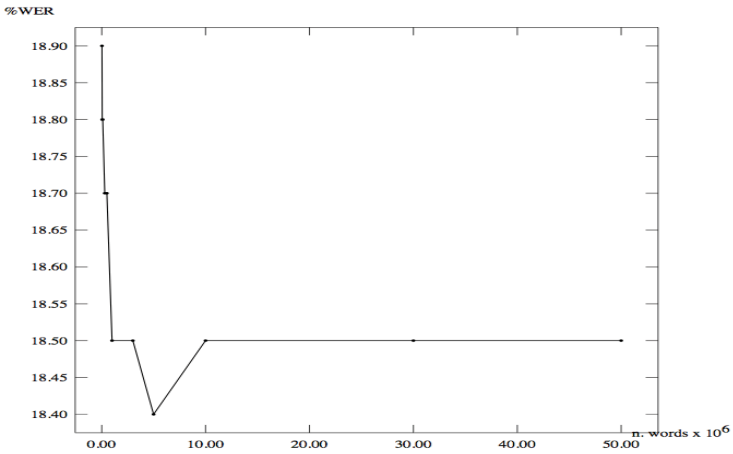


Figure 3: *%WER on dev set, using focused LMs in the final WG based rescoring step, as a function of the number of words used to train auxiliary LMs (the point corresponding to 0 words on the abscissa refers to the usage of the baseline LM).*

Note that the overall perplexity on the dev set PP_{dev} is computed summing the LM log-probabilities of each reference talk and dividing by the total number of words, according to the following equation:

$$PP_{dev} = 10^{\frac{\sum_{i=1}^{i=19} -\log_{10}(P^i_{LM}[W_i])}{NW}} \quad (3)$$

where $P^i_{LM}[W_i]$ is the probability of the reference word sequence in the i^{th} talk, computed using the i^{th} talk-

dependent interpolated LM, and NW is the total number of words in the dev set.

Performance, both in terms of PP and WER, obtained on test set 2011 are reported in Table 1. According to experiments led on dev set, the number of words used to train auxiliary LMs was chosen to be equal to 5M. In Table 1 performance are given for ASR decoding passes two and three and for the final WG based rescoring step. The latter, as explained in section 2.6, has been executed twice: once for producing the primary submission and once for generating the contrastive one. Primary submission is obtained through rescoring of WGs with interpolated LM $\mathbf{wmt12} \oplus \mathbf{giga5} \oplus \mathbf{ted12}$, where $\oplus$ denotes linear interpolation according to equation 1. Contrastive submission is obtained substituting auxiliary LMs LM^i_{aux}, as depicted in figure 1, to **ted12** in the interpolation.

Table 1: *Results obtained on test set 2011 in the various decoding steps, and on test set2012 for both primary and contrastive submissions.*

	test2011		**test2012**
	PP	%WER	%WER
step 2	160	17.1	
step 3	159	16.7	
WG rescoring (primary)	126	15.4	16.8
WG rescoring (contrastive)	146	15.7	17.3

In Table 1 the WERs obtained on test set 2012 are also given for both primary and contrastive submissions. Note that on both test sets the usage of focused LMs (contrastive submissions) allows to achieve performance comparable with that of primary submissions, but without using in-domain data for LM training.

5. Conclusions

We presented our submission runs to the IWSLT2012 Evaluation Campaign for the ASR English track. Our ASR system was trained on a significant portion of TED talk recordings, by exploiting an automatic data selection method evaluating the fidelity of the provided transcripts.

We have described a method for focusing LMs towards the output of the ASR system. The approach is based on the useful and efficient selection, according to a novel similarity score, of documents belonging to large sets of text corpora on which the general purpose LM, used along the various ASR decoding steps, was trained. Significant improvement on WER has been reached without making use of in-domain text data.

Future work will address domains different from TED, the usage of larger sets of text corpora and more efficient selection methods.

6. Acknowledgements

This work was partially supported by the European project EU-BRIDGE, under the contract FP7-287658.

7. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. of the International Workshop on Spoken Language Translation*, Hong Kong, HK, December 2012.

[2] S. Maskey and A. Sethy, "Resampling Auxiliary Data for Language Model Adaptation in Machine Translation for Speech," in *Proc. of ICASSP*, Taipei,Taiwan, April 2009, pp. 4817–4820.

[3] G. Lecorve, J. Dines, T. Hain, and P. Motlicek, "Supervised and unsupervised Web-based language model domain adaptation," in *Proc. of INTERSPEECH*, Portland, USA, September 2012.

[4] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," in *First International Conference on Machine Learning*, New Brunswick:NJ, USA, 2003.

[5] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[6] G. Stemmer, F. Brugnara, and D. Giuliani, "Using Simple Target Models for Adaptive Training," in *Proc. of ICASSP*, vol. 1, Philadelphia, PA, March 2005, pp. 997–1000.

[7] D. Giuliani, M. Gerosa, and F. Brugnara, "Improved automatic speech recognition through speaker normalization." *Computer Speech and Language*, vol. 20, no. 1, pp. 107–123, Jan. 2006.

[8] D. Giuliani and F. Brugnara, "Experiments on Cross-System Acoustic Model Adapatation," in *ASRU Workshop 2007*, Kyoto, Japan, Dec. 2007, pp. 117–122.

[9] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models," in *Proc. of INTERSPEECH*, Brisbane, Australia, September 2008, pp. 1618–1621.

[10] X. Aubert and H. Ney, "A word graph algorithm for large vocabulary continuous speech recognition," in *Proc. of ICSLP*, 1994, pp. 1355–1358.

[11] M. Cettolo, "Segmentation, classification and clustering of an italian broadcast news corpus," in *Proc. of Content-Based Multimedia Inf. Access Conf. (RIAO)*, Paris, France, 2000, pp. 372–381.

[12] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

The 2012 KIT and KIT-NAIST English ASR Systems for the IWSLT Evaluation

Christian Saam[1], Christian Mohr[1], Kevin Kilgour[1], Michael Heck[1], Matthias Sperber[1], Keigo Kubo[2], Sebastian Stüker[1], Sakriani Sakti[2], Graham Neubig[2], Tomoki Toda[2], Satoshi Nakamura[2] and Alex Waibel[1]

[1]Institute for Anthropomatics, Karlsruhe Institute of Technology, Germany
[2]Augmented Human Communication Laboratory, Nara Institute of Science and Technology, Japan
{michael.heck,matthias.sperber}@student.kit.edu
{sebastian.stueker,kevin.kilgour,christian.mohr,christian.saam,alex.waibel}@kit.edu
{keigo-k,ssakti,neubig,tomoki,s-nakamura}@is.naist.jp

Abstract

This paper describes our English *Speech-to-Text* (STT) systems for the 2012 IWSLT TED ASR track evaluation. The systems consist of 10 subsystems that are combinations of different front-ends, e.g. MVDR based and MFCC based ones, and two different phone sets. The outputs of the subsystems are combined via confusion network combination. Decoding is done in two stages, where the systems of the second stage are adapted in an unsupervised manner on the combination of the first stage outputs using VTLN, MLLR, and cMLLR.

Index Terms: speech recognition, IWSLT, TED talks, evaluation system, system development

1. Introduction

The *International Workshop on Spoken Language Translation* (IWSLT) offers a comprehensive evaluation campaign on spoken language translation. One part of the campaign focuses on the translation of TED Talks[1], short 5-25min presentations by people from various fields related in some way to Technology, Entertainment, and Design (TED) [1]. In order to evaluate different aspects of this task IWSLT organizes several evaluation tracks on this data covering the aspects of automatic speech recognition (ASR), machine translation (MT), and the full-fledged combination of the two of them into speech translation systems.

The goal of the TED ASR track is the automatic transcription of TED lectures on a given segmentation, in order to interface with the machine translation components in the speech-translation track. The quality of the resulting transcriptions are measured in word error rate (WER).

In this paper we describe our English ASR systems with which we participated in the TED ASR track of the 2012 IWSLT evaluation campaign. This year, our system is a further development of our last year's evaluation system [2] and makes use of system combination and cross-adaptation, by utilising acoustic models which are trained with different acoustic front-ends and employ two different phoneme sets. In addition to last year, we also included TED talks available via TED's website by training on them in a slightly supervised manner.

We submitted two primary systems. One was solely developed by KIT, the other one was developed in cooperation with NAIST in Japan. A description of the additional work done by NAIST on the KIT-NAIST (contrastive) submission can be found in [3].

On the 2011 evaluations set, which serves as a progress test set, we were able to reduce the word error rate of our transcription systems from 17.1% to 12.0%, a relative reduction of 29.8%. On the 2012 evaluation set, the KIT-NAIST primary system reached a WER of 12.4%.

The rest of this paper is structured as follows. Section 2 describes the data that our system was trained on. This is followed by Section 3 which provides a description of the two acoustic front-ends used in our system. An overview of the techniques used to build our acoustic models is given in Section 4. We describe the language model used for this evaluation in Section 5 and our decoding strategy and results are presented in Section 6.

[1]http://www.ted.com/talks

Text corpus	Word Count	sources
IWSLT training data transcripts	3 million	2
News (+news commentary)	2114 million	4
Parallel Giga Corpus	523 million	1
LDC English Gigaword 4	1800 million	6
UN + Europarl documents	376 million	1
Google Books Ngrams (subset)	1000 million ngrams	1
total	4816 million	15

Table 1: *Language Model training data word count per corpus after cleaning and data selection and number of text sources included in corpus. The total word count does not include the Google Books Ngrams.*

2. Training Data

For acoustic model training we used the following data sources:

- 237 hours of Quaero training data from 2010 to 2012.
- 157 hours of data downloaded from the TED talks website, including the subtitles provided by the TED conferences archive

For the language model and vocabulary selection we used the subtitles of the TED talks and text data from various sources (see Table 1) totalling about 4816 million words.

3. Front-Ends

We trained systems for two different kinds of acoustic front-ends. One is based on the widely used *mel-frequency cepstral coefficients* (MFCC) obtained from a discrete Fourier transform and the other on the *warped minimum variance distortionless response* (MVDR). The second front-end replaces the Fourier transformation by a warped MVDR spectral envelope [4], which is a time domain

technique to estimate an all-pole model using a warped short time frequency axis such as the mel-scale. The use of the MVDR eliminates the overemphasis of harmonic peaks typically seen in medium and high pitched voiced speech when spectral estimation is based on linear prediction.

For training, both front-ends provided features every 10 ms. During decoding this was changed to 8 ms after the first stage. The altered frame-shift introduces a slight variation in the decoding results which can be exploited in the ROVER stage of the decoding process.

For the MVDR front-end we used a model order of 22 without any filter bank since the warped MVDR already provides the properties of the mel-scale filter bank, namely warping to the mel-frequency and smoothing. The advantage of this approach over the use of a higher model order and a linear filter bank for dimensionality reduction is an increase in resolution in low frequency regions which cannot be attained with traditionally used mel-scale filter banks. Furthermore, with the MVDR we apply an unequal modelling of spectral peaks and valleys that improves noise robustness, due to the fact that noise is mainly present in low energy regions.

Both front-ends apply *vocal tract length normalization* (VTLN) [5]. For MFCC this is done in the linear domain, for MVDR in the warped frequency domain. The MFCC front-end uses 13 or 20 cepstral coefficients, the MVDR front-end uses 15. The mean and variance of the cepstral coefficients were normalized on a per-utterance basis. For both front-ends 15 adjacent frames were combined into one single feature vector. The resulting feature vectors were then reduced to 42 dimensions using *linear discriminant analysis* (LDA). Through the temporal context present in the stacked super-vectors the LDA can implicitly perform an approximation of dynamic spectral features. The dimensionality of the final feature vectors was empirically proven to work well and coincides with the dimensionality of a 14 dimensional static feature vector augmented with first and second order dynamic features.

In recent years neural network based features have been shown to improve ASR systems [6]. A typical setup involves training a neural network to recognize phones (or phone-states) from a window of ordinary (e.g. MFCC) feature vectors. With the help a hidden bottleneck layer the trained network can be used to project the input features onto a feature vector with an arbitrarily chosen dimensionality [7]. The input vector is derived from a 15 frame context window with each frame containing 20 MFCC or MVDR coefficients. So far, we used LDA to reduce the dimensionality of this input vector, which limits the resulting LDA-features to linear combinations of the input features. A *multi layer perceptron* (MLP) with the bottleneck in the 2nd hidden layer can make use of non-linear information.

For our IWSLT systems we used bottleneck features for both our MVDR and MFCC front ends.

4. Acoustic Modeling

4.1. Data Preprocessing

For the TED data only subtitles were available so the data had to be split into sentence-like chunks. Therefore the data was decoded to discriminate speech and non-speech and a forced alignment given the subtitles was done where only the relevant speech parts detected by the decoding were used. All this preprocessing was done at NAIST.

4.2. AM Training

We used a context dependent quinphone setup with three states per phoneme, and a left-to-right topology without skip states. All acoustic models initially used 8,000 distributions and codebooks derived from decision-tree based clustering of the states of all possible quinphones. The models were trained using *incremental splitting of Gaussians* (MAS) training, followed by *optimal feature space* training and 2 iterations of Viterbi training. All models use *vocal tract length normalization* (VTLN). After training the continuous density tied state models we further split the state clusters to arrive at 24000 distributions over the 8000 codebooks again based on a decision-tree. Then we trained these semi-continuous models with two iterations of Viterbi training. For some systems the semi-continuous models were worse than the fully-continuous ones, so for the final decoding we used the ones that achieved lower WER on the development data.

We used two different phoneme sets. The first one is based on the CMU dictionary [2] and is the same phoneme set as the one used in last year's system. It consists of 45 phonemes and allophones. The second phoneme set is derived from the BEEP dictionary [3] and contains 52 phonemes and allophones. For the CMU phoneme set we generated missing pronunciations with the help of FESTIVAL [8], while for the beep dictionary we used Sequitur [9] for this. Both grapheme to phoneme converters were trained on subsets of the respective dictionaries.

In total we trained 9 different acoustic models, combining different front-ends and different phoneme sets, which were combined for decoding as described in 6. We found that not all possible combinations need to be trained. The improvements of adding models with new combinations of techniques already used in other systems in different combinations is very small especially when the number of single systems is large.

5. Language Modeling

A 4gram case sensitive language model with modified Kneser-Ney smoothing was built for each of the text sources listed in Table 1. This was done using the SRI Language Modelling Toolkit [10]. Only half the transcripts of the IWSLT develpoment data were used to build a language model, the other half was used as our tuning set. The aforementioned language models built from the text sources in Table 1 were interpolated using interpolation weights estimated on this tuning set resulting in a 4 GB language model with $56,300k$ 2grams, $330,488$ 3grams and $909,927k$ 4grams. The NAIST language model [3] used in KIT-NAIST primary was built with the same sources and tools but applied more thorough data selection strategies for the LDC Gigaword texts.

5.1. Vocabulary Selection

To select the vocabulary the development data text was randomly split in half. For each of our text sources, except the Gigaword and Google Books ngrams (see Table 1) we built a Witten-Bell smoothed unigram language model using the union of the text sources' vocabulary as the language models' vocabulary (global vocabulary). With the help of the maximum likelihood count estimation method described in [11] we found the best mixture weights for representing the tuning set's vocabulary as a weighted mixture of the sources' word counts thereby giving us a ranking of all the words in global vocabulary by their relevance to the tuning set. The top 130k words were selected as our vocabulary. Unknown pronunciations were automatically generated using the aforementioned grapheme to phones conversion.

[2] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[3] ftp://svr-ftp.eng.cam.ac.uk/pub/comp.speech/dictionaries/beep.tar.gz

6. Decoding Strategy and Results

The decoding was performed with the *Janus Recognition Toolkit* (JRTk) developed at Karlsruhe Institute of Technology and Carnegie Mellon University [12]. Our decoding strategy is based on the principle of system combination and cross-system adaptation. System combination works on the principle that different systems commit different errors that cancel each other out. Cross-system adaptation profits from the fact that the unsupervised acoustic model adaptation works better when performed on output that was created with a different system that works approximately equally well [13]. The final step in our system decoding set-up is the ROVER combination of several outputs [14].

We trained 9 different acoustic models as described in section 4 and a language model as described in section 5. An additional acoustic model and an additional language model was trained at NAIST. For the IWLST ASR track 3 different submissions were done, which are described in the following.

6.1. KIT Primary Submission

The decoding strategy of the KIT primary submission is described in Figure 1. The set-up used for our evaluation system consists of two stages. In each stage multiple systems are run, and their output is combined with the help of *confusion network combination* (CNC) [15]. On this output the acoustic models of the next stage are then adapted using *Vocal Tract Length Normalization* (VTLN) [5], *Maximum Likelihood Linear Regression* (MLLR) [16], and *feature space constrained MLLR* (fMLLR) [17]. Finally the ten second pass decodings and the CNC outputs of the first pass results as well as the CNC outputs over the second pass decodings are combined using ROVER.

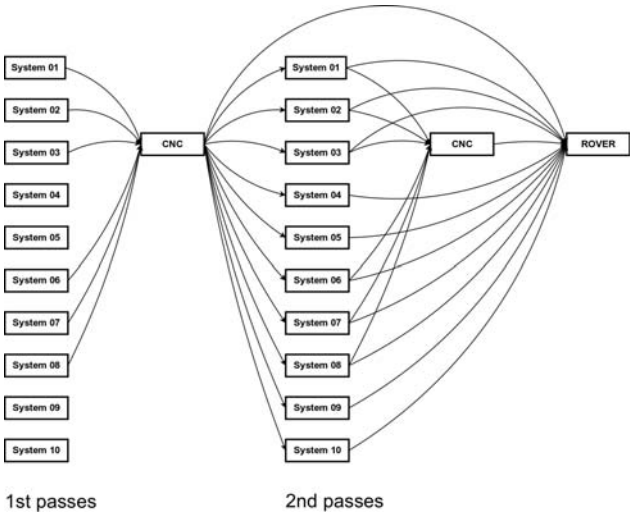


Figure 1: Decoding Strategy of the KIT Primary Submission

6.2. KIT-NAIST Primary and Contrastive Submission

Further to the KIT primary submission we submitted the outputs of two more systems in the IWLST ASR track namely the KIT-NAIST primary and contrastive submissions. Figure 2 shows the principal decoding strategies for all submissions done. The three submissions are depicted as the two rightmost rectangles and the central rectangle.

The KIT-NAIST contrastive submission differs from the KIT primary submission in the fact that a different language model and pronunciation dictionary was used for the decoding which were trained in cooperation with NAIST.

The KIT-NAIST primary submission then is a combination of the KIT primary and the KIT-NAIST contrastive submissions. We combined a subset of outputs of the second passes and CNCs done for both the KIT primary submission and for the KIT-NAIST contrastive submission. In order to let the ROVER combine the most diverse outputs we selected ten second pass systems using the most diverse techniques plus two CNCs. That is the five most diverse of the ten KIT systems and the five most diverse of the ten KIT-NAIST systems respectively, together with the CNC of the KIT-NAIST first pass outputs and the CNC of the KIT second pass outputs. The final system output for the KIT-NAIST primary submission is depicted in Figure 2 by the central rectangle.

System	WER
KIT 2011	17.4%
KIT 2012	12.0%

Table 2: *WER on tst2011 with KIT's system for the evaluation campaign of 2011 compared to the system for the campaign of 2012.*

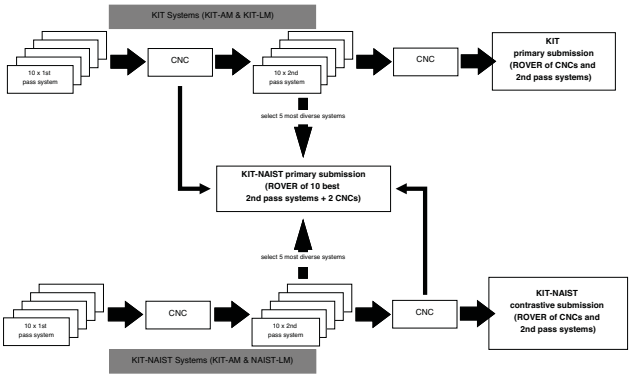


Figure 2: Decoding Strategy of the KIT Primary, KIT-NAIST Primary and Contrastive Submissions

6.3. Results

We evaluated our systems on the IWSLT test sets from 2010 (tst2010), 2011 (tst2011) and 2012 (tst2012). We used the tst2010 set as development set and for parameter optimization. Sets tst2011 and tst2012 were used for this years evaluation campaign, set tst2011 also for last years campaign.

Since the tst2011 set was used for this years and last years evaluation campaign we can indicate our progress over the last year. The compared results are shown in Table 2.

Table 3 shows the results of the KIT primary decoding strategy and its intermediate steps on the development set tst2010.

Table 4 shows the results of all our submissions on all described test sets.

7. Conclusion

In this paper we presented our English LVCSR systems, with which we participated in the 2012 IWSLT evaluation.

System	WER
Single best 1st pass system	17.8%
CNC 1st pass	16.6%
Single best 2nd pass system	15.3%
CNC 2nd pass	14.7%
ROVER	14.3%

Table 3: *WER of the decoding strategy for the KIT primary submission and its intermediate steps on the development set.*

	KIT primary	KIT-NAIST primary	KIT-NAIST contrastive
tst2010	14.3%	14.0%	14.4%
tst2011	12.0%	12.0%	12.3%
tst2012	12.7%	12.4%	12.6%

Table 4: *WER for our three submissions for the three different test sets.*

8. Acknowledgements

This work was supported in part by an interACT student exchange scholarship. '*Research Group 3-01*' received financial support by the '*Concept for the Future*' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative. The work leading to these results has received funding from the European Union under grant agreement $n \circ 287658$. This work was partly realized within the Quaero Programme, funded by OSEO, French State agency for innovation.

9. References

[1] M. Federico, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the iwslt 2012 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) 2012*, Hong Kong, December 6-7 2012.

[2] S. Stüker, K. Kilgour, C. Saam, and A. Waibel, "The 2011 kit english asr system for the iwslt evaluation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) 2011*, San Francisco, December 8-9 2011.

[3] M. Heck, K. Kubo, M. Sperber, S. Sakti, S. Stüker, K. Kilgour, C. Mohr, C. Saam, G. Neubig, T. Toda, S. Nakamura, and A. Waibel, "The kit-naist (contrastive) english asr system for iwslt 2012," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) 2012*, Hong Kong, December 6-7 2012.

[4] M. Wölfel and J. McDonough, "Minimum variance distortionless response spectralestimation, review and refinements," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 117–126, September 2005.

[5] P. Zhan and M. Westphal, "Speaker normalization based on frequency warping," in *ICASSP*, Munich, Germany, April 1997.

[6] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "On using mlp features in lvcsr," in *Proceedings of ICSLP*. Citeseer, 2004.

[7] K. Kilgour, C. Saam, C. Mohr, S. Stüker, and A. Waibel, "The 2011 kit quaero speech-to-text system for spanish," 2011.

[8] A. W. Black and P. A. Taylor, "The Festival Speech Synthesis System: System documentation," Human Communciation Research Centre, University of Edinburgh, Edinburgh, Scotland, United Kongdom, Tech. Rep. HCRC/TR-83, 1997.

[9] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, May 2008.

[10] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *ICSLP*, 2002.

[11] A. Venkataraman and W. Wang, "Techniques for effective vocabulary selection," *Arxiv preprint cs/0306022*, 2003.

[12] H. Soltau, F. Metze, C. Fuegen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *ASRU*, 2001.

[13] S. Stüker, C. Fügen, S. Burger, and M. Wölfel, "Cross-system adaptation and combination for continuous speech recognition: The influence of phoneme set and acoustic front-end," in *Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech 2006, ICSLP)*. Pittsburgh, PA, USA: ISCA, September 2006, pp. 521–524.

[14] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proceedings the IEEE Workshop on Automatic Speech Recognition and Understanding*. Santa Barbara, CA, USA: IEEE, December 1997, pp. 347–354.

[15] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, October 2000.

[16] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[17] V. Digalakis, D. Rtischev, and L. Neumeyer, "Speaker adaptation using constrained estimation of gaussian mixtures," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 5, pp. 357–366, 1995.

The KIT-NAIST (Contrastive) English ASR System for IWSLT 2012

Michael Heck[1], *Keigo Kubo*[2], *Matthias Sperber*[1], *Sakriani Sakti*[2], *Sebastian Stüker*[1], *Christian Saam*[1]
Kevin Kilgour[1], *Christian Mohr*[1], *Graham Neubig*[2], *Tomoki Toda*[2], *Satoshi Nakamura*[2], *Alex Waibel*[1]

[1]Institute for Anthropomatics, Karlsruhe Institute of Technology, Germany
[2]Augmented Human Communication Laboratory, Nara Institute of Science and Technology, Japan

{michael.heck,matthias.sperber}@student.kit.edu
{sebastian.stueker,christian.saam,kevin.kilgour,christian.mohr,alex.waibel}@kit.edu
{keigo-k,ssakti,neubig,tomoki,s-nakamura}@is.naist.jp

Abstract

This paper describes the KIT-NAIST (Contrastive) English speech recognition system for the IWSLT 2012 Evaluation Campaign. In particular, we participated in the ASR track of the IWSLT TED task. The system was developed by Karlsruhe Institute of Technology (KIT) and Nara Institute of Science and Technology (NAIST) teams in collaboration within the interACT project. We employ single system decoding with fully continuous and semi-continuous models, as well as a three-stage, multipass system combination framework built with the Janus Recognition Toolkit. On the IWSLT 2010 test set our single system introduced in this work achieves a WER of 17.6%, and our final combination achieves a WER of 14.4%.

1. Introduction

Similar to the IWSLT 2011 Evaluation Campaign [1], IWSLT 2012 featured an Automatic Speech Recognition (ASR) track whose task it was to recognize the recordings made available by TED on their website[1][2]. The TED talks collection is a web repository of recordings of public speeches/talks of about 5-25 minutes by people from various fields of expertise covering repetitive topics related to technology, entertainment and design (TED). This paper describes the ASR (contrastive) system developed for this campaign by the KIT-NAIST team in collaboration under the interACT project. Detail descriptions of the KIT-NAIST primary submission which was a system combination between the KIT primary submission and this contrastive submission can be found in [3].

The main challenge of this ASR track is to develop a system that is capable of recognizing spontaneous and open-domain speeches. Here, we employ: (1) acoustic models trained on European Parliament Plenary Sessions (EPPS) recordings [4] and additional publicly available transcribed TED audio data crawled from the web; (2) 4-gram language models that were trained by interpolating TED data with other provided corpora, as well as a topic adaptated LM using latent Dirichlet allocation (LDA); (3) a pronunciation dictionary in which the pronunciations of unknown words were constructed using several grapheme-to-phoneme methods; (4) single system decoding with fully continuous and semi-continuous models, as well as a three-stage, multipass system combination framework.

The rest of this paper is structured as follows. Section 2 summarizes data resources used for the experiments, and Section 3 provides a description of acoustic front-ends used in our system. An overview of the techniques and data used to build our acoustic models is given in Section 4. We describe the language model used for this evaluation in Section 5 and pronunciation lexicon in Section 6. Our decoding strategy and experimental results are explained in Section 7. Finally, the conclusion is drawn in Section 8.

2. Data Resources

2.1. Training Corpora

For acoustic model training, the following speech corpora were used:

- 80 hours of manually transcribed English European Parliament Plenary Session (EPPS) speeches, provided by RWTH Aachen within the TC-STAR project [4].
- 157 hours of TED talks released before the cut-off date of 31 December 2010, downloaded from the TED websites with the corresponding subtitles.

For language model training, the following text corpora provided by the IWSLT organizer were used:

- 2M words of TED transcripts.
- The English portion of the English-French training data from the Sixth Workshop on Statistical Machine Translation (WMT 2011), including News Commentary (NC), EuroParl (EPPS), NEWS, and GIGA data.

2.2. Test Corpora

Table 1 describes both test sets ("tst2011" and "tst2012") used for this year's evaluation campaign, as well as our de-

[1]http://www.ted.com/talks

velopment set for system development and parameter optimization ("tst2010"). "tst2010" is a data set which was also used as development set for last year's ASR task. "tst2011" comprises of TED talks newer than December 2010, is the test set for the IWSLT 2011 ASR task and serves as progress test set to measure the improvement in systems from 2011 to 2012. "tst2012" is a collection of some of the most recent recordings made available by TED. All sets were used with the original pre-segmentation provided by the IWSLT organizers.

Set	#talks	#utt	dur	dur/utt
tst2010	11	1664	2.5h	5.4s
tst2011	8	818	1.1h	4.9s
tst2012	11	1124	1.7h	5.6s

Table 1: Statistics of the development set ("tst2010") and the test sets ("tst2011" and "tst2012"), including the total number of talks (*#talks*), the total number of utterances (*#utt*), the overall speech duration (*dur*), and average speech duration per utterance (*dur/utt*).

3. Front-end

We trained the system with a front-end based on the widely used mel-frequency cepstral coefficients (MFCC). The front-end provides features every 10ms. During decoding this was changed to 8ms after the first stage, so that in ROVER hypotheses from first and second pass can be combined. This is done because it may be beneficial for various sounds to have a higher frame rate, while for some other that may not be the case. Therefore a hypotheses combination from different frame rates may lead to better results. During training and decoding, the features were obtained by a discrete Fourier transform followed by a Mel-filterbank. Vocal tract length normalization (VTLN) is done in the linear domain [5]. The MFCC front-end uses 13 cepstral coefficients. Mean and variance are normalized on a per-utterance basis. Finally, to incorporate the temporal structures and dependencies, 15 adjacent (center, 7 left, and 7 right) frames are stacked into one single feature vector leading to 195 dimensional super vector (15x13 dimensions). It then reduced to an optimum 42 dimensions by applying a linear discriminant analysis.

4. Acoustic Modeling

4.1. Data Preprocessing

Segmenting the TED data into sentence-like chunks used for building a training set was performed with the help of a decoding pass on the input data in order to discriminate speech and non-speech regions and doing a forced alignment given the subtitles. Beforehand, the relevant speech part of each downloaded video soundtrack was cut with the time stamps given by the subtitle files. The segmentation was done by splitting at non-speech regions of notable length. In order to compensate for occasional inaccuracies of the computed time stamps, we merged successive segments by the simple heuristic, "As long as the transcription of the subsequent segment does not start with an uppercase letter, add it to the current segment." This resulted in a sentence-like segmentation of the TED data. While the manually transcribed EPPS data has predefined speaker labels and therefore does not need to be clustered, we made the simple assumption for the TED data, that each talk is spoken by exactly one speaker. Table 2 lists the details of the resulting utterances.

Data	#talks	#utt	dur	dur/utt
EPPS	1,894	52,464	80h	5.5s
TED	711	105,692	157h	5.3s

Table 2: Statistics of speech data for acoustic model training, including the total number of talks (*#talks*), the total number of utterances (*#utt*), the overall speech duration (*dur*), and average speech duration per utterance (*dur/utt*).

4.2. AM Training

All models are context-dependent quinphones with a standard three-state left-to-right HMM topology without skip states. The models use 24,000 distributions over 8,000 codebooks. First, a fully continuous system using 2,000 distributions and codebooks was trained by using incremental splitting of Gaussians training (MAS) [6], followed by optimal feature space training (OFS) which is a variant of semi-tied covariance (STC) [7] training using one global transformation matrix. After generating new labels for the training data, a system using 8,000 distributions and codebooks was trained in the same way, and further refined by 2 iterations of Viterbi training. The semi-continuous system was trained after clustering the models resulting in 24,000 distributions over 8,000 codebooks with 2 iterations of Viterbi training.

5. Vocabulary and Language Model

5.1. Data Preprocessing

We normalized the training data sources of TED, NEWS, NC, EPPS, and GIGA, in a case-insensitive fashion. Noisy parts were omitted from the GIGA corpus, using rules to detect, e.g., HTML tags and very short sentences. Table 3 shows the resulting text corpora along with their total size (word count) and vocabulary size.

Data	Size	Vocabulary
TED	2.4m	43k
EPPS	52m	79k
NC	4.5m	50k
NEWS	2,300m	986k
GIGA	576m	501k

Table 3: Total size (word count) and vocabulary size of the individual text corpora.

5.2. Vocabulary

For the vocabulary selection, we followed an approach proposed by Venkataraman et al. [8]. We built unigram lan-

guage models using Witten-Bell smoothing [9] from all text sources except GIGA, and determined unigram probabilities that maximized the likelihood of a held-out TED data set. We then defined the 150k most probable words as the vocabulary.

5.3. LM Training

Using the SRILM toolkit [10], we built 4-gram language models with modified Kneser-Ney smoothing [11] from each of the text corpora. These were then combined using linear interpolation as follows:

$$P(w|h) = \lambda_1 P_1(w|h) + \lambda_2 P_2(w|h) + \cdots + \lambda_k P_k(w|h). \quad (1)$$

The interpolation weights $\lambda_1, \ldots, \lambda_k$ were chosen to maximize the likelihood of a held-out TED data set. The resulting language model contains 43 million bigrams, 190 million trigrams, and 382 million 4-grams. The effect of the different training corpora on the language model perplexity is summarized in Table 4.

Data	Perplexity
TED only	184.03
+ EPPS, NC	167.84
+ NEWS	133.51
+ GIGA	133.16

Table 4: Language model perplexities on tst2010 for different amounts of training data.

5.4. Topic Adaptation

During development, we further applied topic adaptation using LDA (see [12]). Using the given document structure of the TED corpus, we inferred 50 topics, using a vocabulary of 10k words. We estimated a separate 4-gram language model for each topic by using all sentences in the TED training data that had at least one word assigned to this topic. This strategy allows assigning a sentence to several topics, as opposed to much of the previous work that enforces a hard assignment decision for each training unit (e.g. see [13]). For the actual decoding of a specific talk, all words from the first-pass hypothesis that have a confidence value higher than a certain threshold are used to estimate the current topic distribution. The top 10 topics (a limitation imposed by SRILM) are linearly interpolated with weights according to that distribution. Finally, this adapted language model is interpolated with the background language model described above. The confidence threshold and the weight for the interpolation of adapted and background language models were chosen to optimize perplexity on a development data set. Topic model adaptation reduced the perplexity on the talks in the development set ("tst2010") by 0.9% on average. The effect in overall system performance is discussed in Section 7.1.

6. Pronunciation Lexicon

6.1. Phoneme Set

We employ the same phoneme set used by KIT with 45 phonemes, and utilize the existing pronunciation dictionary: (1) the CMU Pronouncing Dictionary [14]; a machine-readable pronunciation dictionary for North American English that contains over 125,000 words and their transcriptions based on 39 phonemes; (2) the EPPS dictionary with KIT phoneme set. Since both pronunciation dictionaries use different phoneme sets, our first step is to convert the 39-phonemes of the CMU dictionary into the KIT phoneme set. This is done using the Sequitur grapheme-to-phoneme (G2P) tool based on joint n-gram models [15]. All words that were covered by both the CMU dictionary and the EPPS dictionary were used as phoneme-to-phoneme training data. Then, by utilizing the trained phoneme-to-phoneme model, the pronunciation of words included in CMU dictionary but not included in EPPS dictionary were converted into new pronunciations based on the KIT phoneme set. Finally, we obtained 135k words of the CMU dictionary with the KIT phoneme set (45 phonemes) as baseline dictionary.

6.2. G2P Conversion

Next, we explored various G2P conversion techniques for handling pronunciations of words that have not been covered by the baseline CMU dictionary (135k words, 45 phonemes). These include: (1) Sequitur G2P based on joint n-gram models (denoted as *Sequitur*); (2) DirecTL+ based on online discriminative training [16, 17] (denoted as *DirecTL+*); and (3) merging 1-best of *Sequitur* and *DirecTL+* results (denoted as *Merge*(1)+(2)).

To find the optimum G2P technique, we employed the baseline CMU dictionary (135k words, 45 phonemes) with a 10% test set, a 5% development set, and the remaining data as training set. Table 5 summarizes the results in terms of Recall, Precision, F-value.

	Recall	Precision	F-measure
(1) *Sequitur*	55.19	55.16	55.17
(2) *DirecTL+*	55.61	55.61	55.61
Merge(1)+(2)	63.23	49.80	55.71

Table 5: Recall, Precision and F-measure for various G2P conversion techniques on the baseline CMU dictionary (135k words, 45 phonemes).

Note that, the *Merge*(1)+(2) G2P may result in one or two pronunciations per word, while other techniques only result in one pronunciation per word. In our experiments the DirecTL+ obtains 55.61% in terms of F-value and Sequitur is 55.17%. These results are lower than those of previous research [15, 16, 17] because we employ a more complex phoneme set than the CMU phoneme set and did not delete heteronyms, which are words that share the same written form but have different pronunciations and meanings. Finally, the optimum *DirecTL+* G2P conversion is selected for dictionary construction.

6.3. Dictionary Construction

Last, we constructed a dictionary that would be used for open domain TED talks. Here, we retrain the selected *DirecTL+* G2P conversion using the baseline CMU dictionary (135k words, 45 phonemes) with a 5% development set, and the remaining data as training set. Then, for all words that are included in the LM, but have not been covered by the baseline CMU dictionary (except the capitalized words), the pronunciations were constructed based on *DirecTL+* G2P conversion. For capitalized words, the pronunciations were converted based on rule in which each alphabet included in the word is converted to the alphabetical sound. The number of the converted words was 65k words in the defined 150k vocabulary (see Section 5.2).

7. Decoding Strategy and Results

During development, we evaluated our system using the IWSLT 2010 test set for the lecture task, which was explicitly declared held out data during model training due to the fact that both the IWSLT 2010 development set and test set were initially included in the downloaded raw TED talks intended for training. For comparison we also evaluated the performance on the test2011 set released by the IWSLT organizers.

All speech recognition experiments, i.e. the decoding—as well as acoustic model training—were performed with the Janus Recognition Toolkit (JRTk) that includes the IBIS single pass decoder, developed at Karlsruhe Institute of Technology and Carnegie Mellon University [18]. During development, we evaluated our system mainly using the IWSLT 2010 test set for the lecture task, which was explicitly declared held out data during model training due to the fact that both the IWSLT 2010 development set and test set were initially included in the downloaded raw TED talks intended for training. We observed the recognition accuracy in terms of word error rate (WER) after first pass decoding.

7.1. Single System

Table 6 shows the results given various configurations of the fully continuous system after MAS, OFS and Viterbi training, and the performance of the semi-continuously trained system after two iterations of Viterbi training. For comparison we also evaluated the performance on the test set ("tst2011").

The "tst2010" set was further used for tuning the system and determining the best language model size and dictionary size for decoding data that is very close to the target domain. The IBIS decoder used by JRTk scores the hypothesis related to an input utterance [18] as follows:

$$score(W|X) = logP(X|W) + logP(W) \cdot lz + lp \cdot |W| \quad (2)$$

The lz parameter defines the language model weight, i.e. determines the impact of the language model on the decoding process relative to the acoustic model. The parameter lp is a word transition penalty, helping to normalize the sequence lengths of words W. Note that applying topic model adaptation LM on our development systems improved the WER by up to 2.2% relative. However, results using the final system were mixed, and the adaptation scheme was not included in the final submission.

Data	System		tst2010	tst2011
EPPS	FCHMMs	MAS	36.5%	31.6%
+TED	FCHMMs	MAS	18.8%	16.5%
		OFS	18.8%	16.0%
		VIT1	18.1%	15.9%
		VIT2	18.2%	16.1%
	SCHMMs	VIT1	17.7%	15.6%
		VIT2	17.6%	15.5%

Table 6: Performance of the single system on the development set ("tst2010") and test set ("tst2011") in WER. The fully continuous system uses 8000 codebooks and distributions, the semi-continuous system 24000 distributions.

7.2. System Combination

The decoding strategy for the final submission is based on the principle of system combination and cross-system adaptation. The underlying assumption of system combination is that different systems commit different errors which may cancel each other out. Cross-system-adaptation profits from the fact that the unsupervised acoustic model adaptation methods work better when applied on hypotheses generated by multiple systems that perform about equally well [19].

Our framework for system combination consists of three stages. In the first stage multiple systems, including our system described in this paper, are run. The additional systems differ in the applied front-ends and acoustic models (see [3]) in a way that achieves a high system diversity among the full set of applied systems. The same combination of dictionary and language model is used for all decoding runs. The system outputs of the first stage are combined via confusion network combination (CNC) [20]. The acoustic models of all systems for the second pass are then adapted on this output using VTLN, maximum likelihood linear regression (MLLR) [21] and feature space constrained MLLR (fMLLR) [22]. After the first stage, the frame shift was changed to 8 ms. In the second stage a second CNC is performed. The third and final stage of our system combination framework is a ROVER combination of seven second pass outputs and both CNC outputs [23]: A majority vote among all CNC results and second stage system outputs gave the best results.

The segmentation of the test data was used as is. For simplicity reasons no extra speaker clustering was performed, assuming one speaker per test recording. Table 7 shows the performance of the system combination on the development set ("tst2010") in WER, and Table 8 shows the summary of the final system combination results on various development and test sets in WER. The results shown on test set ("tst2011" and "tst2012") are based on IWSLT 2012 evaluation feedback.

System	WER
Single 1st pass	17.6%
CNC 1st pass (CNC_1)	17.1%
Single 2nd pass	16.1%
CNC 2nd pass (CNC_2)	14.5%
ROVER ($CNC_1 + CNC_2 + 7 * 2$nd pass)	14.4%

Table 7: Comparison of the single system performance and the system combination results on the development set ("tst2010") in WER.

Test set	WER
test2010	14.4%
test2011	12.3%
test2012	12.6%

Table 8: Summary of final system performance performed with ROVER ($CNC_1 + CNC_2 + 7 * 2$nd pass). The results shown on test set ("tst2011" and "tst2012") are based on IWSLT 2012 evaluation feedback.

8. Conclusion

In this paper we described our English speech-to-text system with which we participated in the IWSLT 2012 TED task evaluation on the ASR track. Besides utilizing already existing systems by adjusting them to the new domain, we trained a completely new system by including annotated audio data extracted from TED talks into acoustic model training. Furthermore, we built a dictionary and trained a language model specific to the TED task of this year's evaluation campaign. Our final system utilizes a three-stage, multipass system combination framework. On the IWSLT 2010 test set our single system introduced in this work achieves a WER of 17.6%, and our final combination achieves a 14.4% WER.

9. Acknowledgements

This work was supported in part by an interACT student exchange scholarship. The research leading to these results has in part received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 287658. 'Research Group 3-01' received financial support by the 'Concept for the Future' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative.

10. References

[1] M. Federico, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2011 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) 2011*, San Francisco, CA, USA, December 8-9 2011.

[2] ——, "Overview of the IWSLT 2012 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) 2012*, Hong Kong, December 6-7 2012.

[3] C. Saam, C. Mohr, K. Kilgour, M. Heck, M. Sperber, K. Kubo, S. Stüker, S. Sakti, G. Neubig, T. Toda, S. Nakamura, and A. Waibel, "The 2012 KIT and KIT-NAIST english ASR systems for the IWSLT evaluation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) 2012*, Hong Kong, December 6-7 2012.

[4] C. Gollan, M. Bisani, S. Kanthak, R. Schluter, and H. Ney, "Cross domain automatic transcription on the TC-STAR EPPS corpus," in *Proc. of ICASSP*, Philadelphia, USA, 2005, pp. 825–828.

[5] P. Zhan and M. Westphal, "Speaker normalization based on frequency warping," in *Proc. of ICASSP*, Munich, Germany, 1997, pp. 1039–1042.

[6] T. Kaukoranta, P. Fränti, and O. Nevalainen, "Iterative split-and-merge algorithm for VQ codebook generation," *Optical Engineering*, vol. 37, no. 10, pp. 2726–2732, 1998.

[7] M. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[8] A. Venkataraman and W. Wang, "Techniques for effective vocabulary selection," in *Proc. of EUROSPEECH*, Geneva, Switzerland, 2003, pp. 245–248.

[9] I. Witten and T. Bell, "The zero-frequency problem: estimating the probabilities of novelevents in adaptive text compression," *IEEE Transactions on Information Theory*, vol. 37, no. 4, pp. 1085–1094, 1991.

[10] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. of ICSLP*, Denver, USA, 2002, pp. 901–904.

[11] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proc. ICASSP*, 1995, pp. 181–184.

[12] X.-H. Phan and C.-T. Nguyen, "GibbsLDA++: A C/C++ implementation of latent dirichlet allocation (LDA)," http://jgibblda.sourceforge.net/, 2007.

[13] F. Liu and Y. Liu, "Unsupervised language model adaptation incorporating named entity information," in *Proc. of ACL*, Prague, Czech Republic, 2007, pp. 672–679.

[14] "The carnegie mellon university pronouncing dictionary," http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[15] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[16] S. Jiampojamarn and G. Kondrak, "Online discriminative training for grapheme-to-phoneme conversion," in *Proc. INTERSPEECH*, Beijing, China, 2009, pp. 1303–1306.

[17] C. C. S. Jiampojamarn and G. Kondrak, "Integrating joint n-gram features into a discriminative training framework," in *Proc. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, Beijing, China, 2010, pp. 697–700.

[18] C. F. H. Soltau, F. Metze and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *Proc. of ASRU*, Madonna di Campiglio, Italy, 2001.

[19] S. Stüker, C. Fügen, S. Burger, and M. Wölfel, "Cross-system adaptation and combination for continuous speech recognition: The influence of phoneme set and acoustic front-end," in *Proc. of INTERSPEECH*, Pittsburg, PA, USA, 2006, pp. 521–524.

[20] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[21] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.

[22] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[23] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. of ASRU*, Santa Barbara, USA, 1997, pp. 347–354.

EBMT System of Kyoto University in OLYMPICS Task at IWSLT 2012

Chenhui Chu, Toshiaki Nakazawa, Sadao Kurohashi

Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku
Kyoto, 606-8501, Japan
{chu, nakazawa, kuro}@nlp.ist.i.kyoto-u.ac.jp

Abstract

This paper describes the EBMT system of Kyoto University that participated in the OLYMPICS task at IWSLT 2012. When translating very different language pairs such as Chinese-English, it is very important to handle sentences in tree structures to overcome the difference. Many recent studies incorporate tree structures in some parts of translation process, but not all the way from model training (alignment) to decoding. Our system is a fully tree-based translation system where we use the Bayesian phrase alignment model on dependency trees and example-based translation. To improve the translation quality, we conduct some special processing for the IWSLT 2012 OLYMPICS task, including sub-sentence splitting, non-parallel sentence filtering, adoption of an optimized Chinese segmenter and rule-based decoding constraints.

1. Introduction

We consider that it is quite important to use linguistic information in the translation process when tackling very different language pairs such as Chinese-English and Japanese-English, and one of the most important pieces of information is sentence structure. Many recent studies incorporate some structural information into decoding, but rarely into alignment. In this paper, we adopt a fully tree-based translation framework based on dependency tree structures [1]. In the alignment step, we use Bayesian subtree alignment model based on dependency trees. Section 2 shows a brief description of the model. It is a kind of tree-based reordering model, and can capture non-local reorderings which sequential word-based models cannot often handle properly. In the translation step, we adopt an example-based machine translation (EBMT) system, handling examples which are discontinuous as a word sequence, but continuous structurally. It also considers similarities of neighboring nodes, which are useful for choosing suitable examples matching the context.

Figure 1 shows the overview of our EBMT system on Chinese-English translation. The translation example database is automatically constructed from a parallel training corpus by means of a Bayesian subtree alignment model. Note that both source and target sides of all the examples are stored in dependency tree structures. An input sentence is also parsed and transformed into a dependency structure. For all the sub-trees in the input dependency structure, matching examples are searched in the example database. This step is the most time consuming part, and we exploit a fast tree retrieval method [2]. There are many available examples for one sub-tree, and also, there are many possible sub-tree combinations. The best combination is detected by a log-linear decoding model with features described in Section 3.

In the example in Figure 1, five examples are used. They are combined and produce an output dependency tree. We call nodes surrounding those of the example, "bond" nodes. The bond nodes of one example are replaced by other examples, and thus examples can be combined.

We attended the IWSLT 2012 OLYMPICS task which is a Chinese-to-English text translation task. Based on the characteristic of this task, we conducted some special processing. We split sub-sentences and filtered non-parallel sentences to improve the quality of the supplied corpora. We adopted an optimized Chinese segmenter which can generate segmentation results that are much more similar to English to improve the alignment accuracy. To reduce the computational complexity, we adopted rule-based decoding constraints on the decoding. Details of the above special processing for this task are described in Section 4.

2. Bayesian Subtree Alignment Model based on Dependency Trees

Alignment accuracy is crucial for providing high quality corpus-based machine translation systems because translation knowledge is acquired from an aligned training corpus. For distant language pairs such as Chinese-English and Japanese-English, the word sequential models such as IBM models are quite inadequate (about 20% alignment error rate (AER)), and therefore it is important to improve the alignment accuracy itself. The differences between languages can be seen in Figure 2, which shows an example of Japanese-English. The word or phrase order is quite different for these languages. Another important point is that there are frequent many-to-one or many-to-many correspondences. For example, the Japanese noun phrase "受 光 素子" is composed of three words, whereas the corresponding English phrase consists of only one word "photodetector", and the English func-

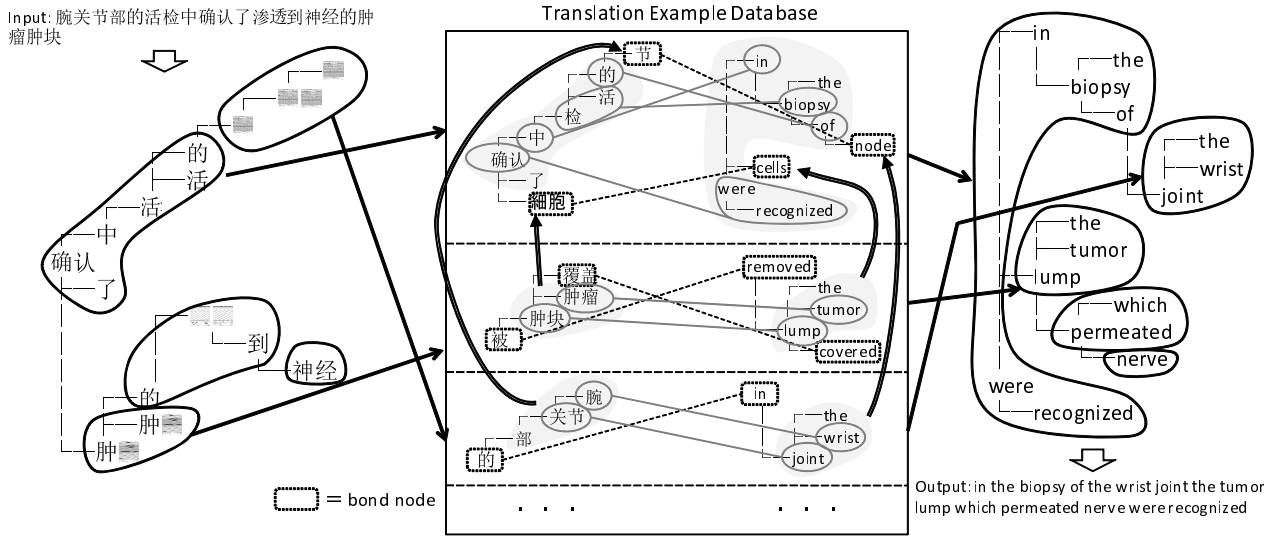


Figure 1: An example of Chinese-English translation.

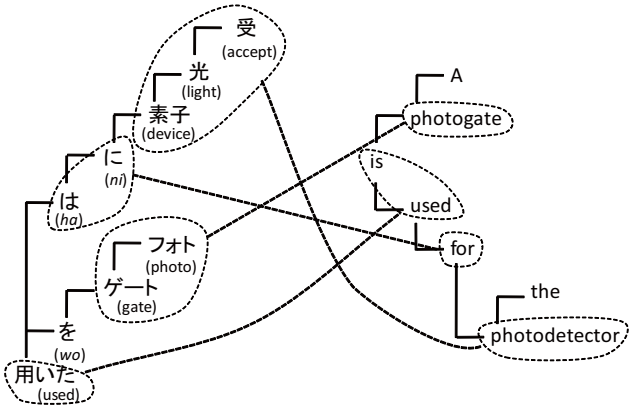


Figure 2: Example of dependency trees and alignment of subtrees. The root of the tree is placed at the extreme left and words are placed from top to bottom.

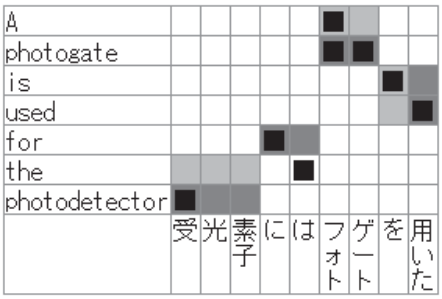


Figure 3: Alignment results from bi-directional GIZA++. Black boxes depict the system output, while dark (Sure) and light (Possible) gray cells denote gold-standard alignments.

tion word "for" corresponds to two Japanese function words "に は". In addition, there are basically no counterparts for the English articles (a, an, the). Figure 3 shows the alignment results from bi-directional GIZA++ together with a combination heuristic called grow-diag-final-and for the same sentence pair given in Figure 2. The system failed to align some words in the Japanese noun phrase, and incorrectly aligned "the ↔ は". The word sequential model is prone to many such errors even for short simple sentences of a distant language pair.

Even if the word order differs greatly between languages, phrase dependencies tend to hold between languages. This can be seen in Figure 2. Therefore, incorporating dependency analysis into the alignment model is useful for distant language pairs. We exploit Bayesian subtree alignment model based on dependency trees [3]. This model incorporates dependency relations of words into the alignment model and define the reorderings on the word dependency trees. Figure 2 shows an example of the dependency trees for Japanese and English.

3. Tree-based Translation

As a tree-based translation method, we adopt an example-based machine translation system [1]. In this section, we briefly introduce the translation procedure in our EBMT system.

3.1. Retrieval of Translation Examples

The input sentence is converted into the dependency structure as in the alignment step. Then, for each sub-tree, avail-

able translation examples are retrieved from the example database. Here the word "available" means that all the words in the focusing input sub-tree appear in the source tree of the example, and the dependency relations between the words are same. We use a fast, on-line tree retrieval technique [2] to get all the available examples from a large training corpus.

3.2. Selection of Translation Examples

We find the best combination of examples by tree-based log-linear model with features shown below:

- **Size of examples**
- Translation probability
- Root node of examples
- Parent node
- Child nodes
- Bond nodes
- NULL-aligned words
- Language model

Among the features, an important one is "Size of examples". Translations that are composed of larger examples can achieve higher quality because translations inside the examples are stable.

3.3. Combination of Translation Examples

When combining examples, in most cases, *bond nodes* are available outside the examples, to which the adjoining example is attached. Using the bond information, we don't need to consider word or phrase order. Bond information naturally solves the reordering problem. Figure 1 is an example of combining translation examples. The combination process starts from the example used for the root node of the input tree (the first one in Figure 1). Then the example for the child node of the sub-tree covered by the initial example is combined (the second and third examples). When combining the second example to the first one, "细胞↔ cells" is used as bond node, and for the third example, "节↔ node" is used as bond node. The combination repeated until all the examples are combined into one target tree. Finally, the output sentence is generated from the tree structure.

Note that there are NULL-aligned nodes in the examples (the nodes which are not circled, such as '了', '部(*part*)' and articles in English).

4. IWSLT 2012 OLYMPICS Task

In this section, we first briefly introduce the IWSLT 2012 OLYMPICS task. We then describe the special processing for this task including sub-sentence splitting, non-parallel sentence filtering, adoption of an optimized Chinese segmenter and rule-based decoding constraints. Finally we report the formal run evaluation results with discussion.

4.1. Task Description

The OLYMPICS task is carried out using parts of the HIT Olympic Trilingual Corpus (HIT) [4] and the Basic Travel Expression Corpus (BTEC) as an additional training corpus. The HIT corpus is a multilingual corpus that covers 5 domains (traveling, dining, sports, traffic and business) that are closely related to the Beijing 2008 Olympic Games. The HIT corpus contains around 52k sentences 2.8 million words in total. The BTEC corpus is a multilingual speech corpus containing tourism-related sentences. The BTEC corpus consists of 20k sentences including the evaluation data sets of previous IWSLT evaluation campaigns. For more details of this task, please refer to [9].

4.2. Sub-sentence Splitting

The corpora supplied for this task have a problem that there are many parallel sentences containing multiple sub-sentences. Since multiple sub-sentences in a single sentence decrease the parsing accuracy, splitting the sentences containing sub-sentences into individual sentences is necessary. Based on our observation, there are two different patterns in the HIT and BTEC corpus for this sub-sentences problem. In the HIT corpus, there are same number of punctuation marks (including comma, period, question mark and exclamation mark) in most parallel sentences with this problem, and can be split using these punctuation marks. Here is one example:

Zh: 我带了些矿泉水和茶，您喜欢喝什么？
(I've brought some mineral water and some tea, which do you prefer?)

En: I've brought some mineral water and some tea. Which do you prefer?

In this example, Chinese sentence and Engligh sentence have the same number of punctuation marks. Moreover, "我带了些矿泉水和茶" corresponds to "I've brought some mineral water and some tea" and "您喜欢喝什么" corresponds to "Which do you prefer". Therefore, it can be split based on the punctuation.

In the BTEC corpus, most parallel sentences with this problem contain same number of EOS punctuation marks (i.e. period, question mark and exclamation mark), and can be split using EOS punctuation marks. Here is one example:

Zh: 非常感谢。你知道我不想赶不上它。
(Thank you so much. You see I don't want to miss it.)

En: Thank you so much. You see, I don't want to miss it.

Therefore, we split the sub-sentences in the HIT and BTEC corpus based on the punctuation marks and EOS punctuation marks respectively.

4.3. Non-parallel Sentence Filtering

Another problem of the supplied corpora is that there are many non-parallel sentences in the HIT corpus. Here is one example:

Zh: 我上牛津大学。
(I am studying at Oxford University.)

En: What about you?

Also, since Chinese and English may use punctuation (especially for the usage of comma) in different places of parallel sentences, the sub-sentence splitting method for the HIT corpus that we described in Section 4.2 can lead to non-parallel sentences. Here is one example:

Zh: 是的，这位女士要一杯曼哈顿酒，我要一杯马丁尼。
(Yes, this lady will have a Manhattan, and I'll have a martini.)

En: Yes, I think so. This lady will have a Manhattan and I'll have a martini.

These non-parallel sentences can decrease the accuracy of alignment and translation performance. Therefore, we propose a filtering method to automatically filter the non-parallel sentences. Our proposed method is an extension of [5], which extracted parallel sentences from comparable corpora by treating it as a classification problem. We think non-parallel sentences filtering can also be solved by classification. We use the same features and classification model described in [5]. The dictionary we used is created from the lexical translation table obtained by running GIZA++ on the whole supplied corpora. We extract the best 5 translation equivalents having translation probability above 0.1 from the lexical translation table as our dictionary. For training data, we use 5,000 parallel sentences from the BTEC corpus, because of the good quality of the BTEC corpus. We create non-parallel sentences from the parallel sentences following the method described in [5]. We generate all the sentence pairs except the original parallel sentence pairs in the Cartesian product, and discard the pairs that do not fulfill the condition of a sentence ratio filter and a word-overlap filter. Then we randomly select 500 non-parallel sentences and add them to the training data. Test data is created using the same method by using another 5,000 parallel sentences from the BTEC corpus. Our data filtering method achieved high accuracy with precision of 97.10%, recall of 84.81% and F-score of 90.54% in the experiment.

We then applied the trained classifier to the HIT corpus for non-parallel sentence filtering and filtered around 1,000 sentence pairs. We conducted translation experiments to investigate the effect of non-parallel sentence filtering on translation quality. Preliminary experimental results showed that non-parallel sentence filtering has little effect on translation quality (only 0.02% BLEU score increased). We think the reason is that the classifier trained on the BTEC corpus does not work well on the HIT corpus because of the difference between these two corpora, thus some parallel sentences are also filtered in this process.

	BLEU
Baseline	0.1162
Optimized	0.1209
Optimized+Constrained	0.1271

Table 1: Results of preliminary translation experiments.

4.4. Optimized Chinese Segmenter

As there are no explicit word boundary markers in Chinese, word segmentation is considered as an important first step in machine translation. Research shows that optimal Chinese word segmentation for machine translation is dependent on the other language, therefore, a bilingual approach is necessary [6]. In this task, we adopted a Chinese segmenter optimized based on a bilingual perspective, which exploits common Chinese characters shared between Chinese and Japanese for Chinese word segmentation optimization [7]. The BLEU scores with and without Chinese segmenter optimization are given in Table 1, indicated as "Optimized" and "Baseline" respectively. Although the Chinese segmenter we used is optimized for Chinese-Japanese machine translation, it shows better translation performance compared to the Chinese segmenter without optimization. We think the reason is that the optimized segmentation results are much more similar to English in number, which can reduce the number of 1-to-n alignments and improve the alignment accuracy.

4.5. Rule-based Decoding Constraints

Translating long and complex sentences is a critical problem in machine translation, because it increases the computational complexity. Finch et al. [8] presented a simple yet efficient method to solve this problem. They split a sentence into smaller units based on part-of-speech (POS) tags and commas, and translate the split units separately. Following their method, we also split a sentence into smaller units during decoding. Our EBMT system tends to choose large examples. Since the development data of this task also has the sub-sentence problem (described in Section 4.3), our system may use examples across punctuation boundaries which can generate translations with unnatural word order. Therefore, we split a source sentence based on comma, period, question mark and exclamation mark for decoding. The BLEU score after constrained decoding is given in Table 1, indicated as "Optimized+Constrained". The result shows that our method achieved better translation performance compared to unconstrained decoding.

4.6. Results

The official scores for the our EBMT system with respect to several of the automatic metrics used for the official evaluation are given in Table 2 (For rankings, please refer to [9]). The scores are low for this task. There are several reasons:

The major reason is the quality and quantity of the sup-

Case/Punctuation	BLEU	METEOR	WER	PER	TER	GTM	NIST
Case and punc	0.1273	0.4628	0.7552	0.6398	71.1530	0.4591	4.1138
No case and no punc	0.1228	0.4137	0.8288	0.6860	79.7690	0.4301	4.3104

Table 2: The official results for the our EBMT system in terms of a variety of automatic evaluation metrics.

plied training data. As described in the previous sections that the supplied data is noisy. To improve the quality of the supplied data, we conducted sub-sentence splitting and non-parallel sentence filtering. However, sub-sentence splitting can lead to additional non-parallel sentences. Although we ran non-parallel sentence filtering, not all of the non-parallel sentences were filtered. Moreover, some parallel sentences may be filtered during this process. Also, there were many out-of-vocabulary (OOV) words during decoding, because of the limited small training data. Sublexical translations could be used to handle the OOV problem [10]. Another possible approach to solve this problem is using external resources such as Wikipedia [11] and Wiktionary. We extracted bilingual titles based on inter-language links in Wikipedia and bilingual terms existed in Wiktionary, and constructed an additional parallel corpus. We conducted translation experiment by adding this corpus to the supplied data. Preliminary experimental results indicated that the additional parallel corpus has bad effect to this task (0.51% BLEU score decreased). We think the reason is the domain difference of the supplied data, Wikipedia and Wiktionary.

Another important reason is the low parsing accuracy of Chinese sentence. The English parser used in the experiments can analyze sentences with over 90% accuracy, whereas the accuracy of the state-of-the-art Chinese parser is not satisfactory. Though the parsing accuracy using gold-standard word segmentation and POS-tags is reasonably high, starting with raw sentences results in less than 80% accuracy (this information was obtained from communication with the authors of [12]). However, the improvement of Chinese parsing in the long run, would also improve the translation quality of our EBMT system. One possible short-term solution for the parsing problem is to use the n-best parsing results in the model. Another kind of solution was proposed by Burkett et al. [13], which described a joint parsing and alignment model that can exchange useful information between the parser and aligner.

5. Conclusions

In this paper, we adopted a linguistically-motivated translation framework for the IWSLT 2012 OLYMPICS task. This framework is composed of Bayesian subtree alignment model based on dependency tree structures, and example-based translation method where the examples are expressed in dependency tree structures. Furthermore, we conducted some special processing for this task to improve the translation quality.

Although our EBMT system can generate adequate and fluent translations, we could not achieve satisfactory results in the run submission. Besides the difficulty of this task itself, our EBMT system suffers from the low accuracy of the Chinese parser. In the future, we aim to improve our system to achieve better translation quality even on limited small training data.

6. References

[1] T. Nakazawa and S. Kurohashi, "Ebmt system of KYOTO team in patentmt task at NTCIR-9," in *Proceedings of the 9th NTCIR Workshop Meeting on Evaluation of Information Access Technologies (NTCIR-9)*, Tokyo, Japan, December 2011, pp. 657–663.

[2] F. Cromieres and S. Kurohashi, "Efficient retrieval of tree translation examples for syntax-based machine translation," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, July 2011, pp. 508–518.

[3] T. Nakazawa and S. Kurohashi, "Bayesian subtree alignment model based on dependency trees," in *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, November 2011, pp. 794–802.

[4] M. Yang, H. Jiang, T. Zhao, and S. Li, *Construct Trilingual Parallel Corpus on Demand*. Chinese Spoken Language Processing, 2006, vol. 4274, ch. Lecture Notes in Computer Science, pp. 760–767.

[5] D. S. Munteanu and D. Marcu, "Improving machine translation performance by exploiting non-parallel corpora," *Computational Linguistics*, vol. 31, no. 4, pp. 477–504, December 2005.

[6] Y. Ma and A. Way, "Bilingually motivated domain-adapted word segmentation for statistical machine translation," in *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Athens, Greece: Association for Computational Linguistics, March 2009, pp. 549–557.

[7] C. Chu, T. Nakazawa, D. Kawahara, and S. Kurohashi, "Exploiting shared Chinese characters in Chinese word segmentation optimization for Chinese-Japanese machine translation," in *Proceedings of the 16th Annual*

Conference of the European Association for Machine Translation (EAMT'12), Trento, Italy, May 2012.

[8] A. Finch, C. ling Goh, G. Neubig, and E. Sumita, "The NICT translation system for IWSLT 2011," in *Proceedings of the International Workshop on Spoken Language Translation 2011*, San Francisco, 12 2011, pp. 49–56.

[9] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 Evaluation Campaign," in *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT)*, Hong Kong, 2012.

[10] C. Huang, H. Yen, P. Yang, S. Huang, and J. S. Chang, "Using sublexical translations to handle the OOV problem in machine translation," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 10, no. 3, pp. 16:1–16:20, Sept. 2011.

[11] J. Niehues and A. Waibel, "Using Wikipedia to translate domain-specific terms in SMT," in *Proceedings of the International Workshop on Spoken Language Translation 2011*, San Francisco, 12 2011, pp. 230–237.

[12] W. Chen, D. Kawahara, K. Uchimoto, Y. Zhang, and H. Isahara, "Dependency parsing with short dependency relations in unlabeled data," in *In Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP-08)*, 2008, pp. 88–94.

[13] D. Burkett, J. Blitzer, and D. Klein, "Joint parsing and alignment with weakly synchronized grammars," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, June 2010, pp. 127–135.

The LIG English to French Machine Translation System for IWSLT 2012

Laurent Besacier, Benjamin Lecouteux, Marwen Azouzi, Luong Ngoc Quang

LIG
University of Grenoble
firstname.lastname@imag.fr

Abstract

This paper presents the LIG participation to the E-F MT task of IWSLT 2012. The primary system proposed made a large improvement (more than 3 point of BLEU on tst2010 set) compared to our last year participation. Part of this improvment was due to the use of an extraction from the Gigaword corpus. We also propose a preliminary adaptation of the driven decoding concept for machine translation. This method allows an efficient combination of machine translation systems, by rescoring the log-linear model at the N-best list level according to auxiliary systems: the basis technique is essentially guiding the search using one or previous system outputs. The results show that the approach allows a significant improvement in BLEU score using Google translate to guide our own SMT system. We also try to use a confidence measure as an additional log-linear feature but we could not get any improvment with this technique.

1. Introduction

This paper describes LIG approach for the evaluation campaign of the 2012 International Workshop on Spoken Language Translation (IWSLT-2012), English-French MT task. This year the LIG participated only to the E-F MT task and focused on the use of driven decoding to improve statistical machine translation. In addition, we used much more parallel data than last year (trying to make use of the Giga-10^9 corpus). Some (un-successful) attempts to use confidence measures to re-rank our N-best hypotheses were also investigated. The remainder of the paper is structured as follows. Section 2 describes the data we used for training our translation and language models. Section 3 presents the concept of driven decoding that allowed us to get improvements using an auxiliary translation (of an online system) to guide the decoding process. Section 4 presents our attempt to use confidence measures and section 5 details the experiments as well as the LIG official results obtained this year.

2. Resources used in 2012

The following sections describe the resources used to build the translation models as well as the language models.

2.1. Translation models training data

We built three translation models for our machine translation systems (see table 1).

- An in-domain translation model trained on TED Talks collection (TED) corpus.
- A (bigger) out-of-domain translation model trained on six different (freely available) corpora in which three of them are part of the WMT 2012 shared task training data:
  - the latest version of the Europarl (version 7) corpus (EUROPARL[1] [1])
  - the latest version of the News-Commentary (version 7) corpus (NEWS-C)
  - the United Nations corpus (UN [2] [2])
- We also used the Corpus of Parallel Patent Applications (PCT[3]), the DGT Multilingual Translation Memory of the Acquis Communautaire (DGT-TM [3]), and the EUconst corpus (EU-CONST [4]). These three corpora are all freely available.
- An additional out-of-domain translation model was trained on a subset of the French-English Gigaword corpus (GIGA-5M). After cleaning, the whole Gigaword corpus was sorted at sentence level according to the sum of perplexities of the source (English) and the target (French) based on two French and English pretrained language models. For this, LMs were trained separately on all the data listed in table 2 except the Gigaword corpus itself (the News Shuffle corpus was also available on the source English side). The separate LMs were then interpolated using weights estimated on dev2010 using EM algorithm (more details on this process are given in the next section). Finally, the GIGA-5M subset was obtained after filtering out the whole Gigaword corpus with a cut-off limit of 300 (ppl). This leads to a subset of 5M aligned sentences.

[1] http://www.statmt.org/europarl/
[2] http://www.euromatrixplus.net/multi-un/
[3] http://www.wipo.int/patentscope/en/data/pdf/wipo-coppa-technicalDocumentation.pdf

System	Corpus	Aligned Sentences
IN-DOMAIN	TED	139,763
OUT-OF-DOMAIN	EU-CONST	4,904
	NEWS-C	124,081
	EUROPARL	1,743,110
	DGT-TM	1,657,662
	PCT	7,739,299
	UN	10,573,628
Additional GIGA-5M	GIGA-TOP-5M	4,392,530

Table 1: Data used for training the translation model.

Corpus	French words	Alpha	Perplexity
TED	2,798,705	0.536023	103.5
EU-CONST	104,698	5.84281e-06	1074.2
NEWS-C	3,224,063	0.0539594	179.4
EUROPARL	44,116,533	0.119409	156.2
DGT-TM	27,582,544	0.0422644	452.5
PCT	164,936,865	0.0484619	625.3
UN	252,849,705	0.0225498	229.4
NEWS-SHUFFLE	608,297,082	0.0834454	162.2
GIGA-5M	117,985,209	0.131878	141.4

Table 2: Data used for training the language model.

These data were used to train three different translation tables in a multiple phrase table decoding framework (corresponding to the *either* option defined in the Moses advanced features).

2.2. Language model training data

For the language model training, in addition to the French side of all of the parallel corpora described above, we used the News Shuffle corpus provided by the WMT 2012 shared task. First a 5-gram back-off interpolated language model with the modified (improved) Kneser-Ney smoothing was trained on each resource using the SRI language modeling toolkit [5]. Then we created a merged LM optimized on a development corpus (dev2010) using EM algorithm. The details on these LM resources and their weights are given in table 2. The table shows that the in-domain data obviously have a strong weight and that the LM trained on Gigaword subset is also well matched to the TED task. On the contrary, the 3 additional corpora PCT, DGT-TM and EU-CONST are the ones that lead to the highest perplexities and they seem quite far from the TED domain (PCT covers different topics like patents, EU-CONST is too small and DGT-TM covers a topic too far from TED).

2.3. Development and test sets

The TED dev2010 set (934 aligned sentences) was used for tuning and the TED tst2010 set (1 664 aligned sentences) was used for testing and making a choice on the best systems to be presented at the evaluation. These sets will be referred to as dev2010 and tst2010 in the rest of this paper. In addition, the TED tst2011 set (818 aligned sentences) and the TED tst2012 set (1 124 aligned sentences) were used for the official evaluation.

2.4. Data pre-processing

This year we used a fully in-house pre-processing. The goal was to use a more specific pre-processing and post-processing steps for English as well as for French. In short, we applied the following steps:

- filter out badly aligned sentences (using several heuristics)
- filter out empty sentences and sentences having more than 50 words
- filter out pairs of sentences where the ratio is more than 9
- punctuation normalization (extra punctuation mark deletion, transform several encodings of a same punctuation mark function to a canonical version, etc.)
- tokenize (different to the default Moses tokenizer using French grammar rules)

- truecase (remove case for the words at the beginning of the sentence while keeping information on the word position)
- spell correction on both source and target sides
- diacritics restoration (notably on uppercase letters at the beginning of sentences)
- Unicode normalization (NFKC)
- normalization of several words (e.g. coeur)
- disambiguate abbreviations and clitics
- HTML entities conversion

To clean the GigaWord corpus, we applied additional cleaning steps. Many heuristics (rules) were used in order to keep only good quality bi-texts.

2.5. System configuration

In the experiments reported here, 26 or 38 features (according to the total number of PT used) were used in our statistical machine translation system: 10 or 15 translation model scores, 14 or 21 distortion scores, 1 LM score, and 1 word penalty score. We used the Minimum Error Rate Training (MERT) method to tune the weights on dev2010 corpus. We are aware that in the future better optimization techniques like MIRA should be used for such a large number of parameters.

3. Driven Decoding for SMT

Recently, the concept of driven decoding (DD), introduced by [6] has been successfully applied to the automatic speech recognition (speech-to-text) task. This idea is to use an auxiliary transcription (coming from another system output or from another source of information) to guide the decoding process. There is a strong interest in applying this concept to statistical machine translation (SMT). The potential applications are: system combination, multi-source translation (from several languages, from several ASR outputs in the case of speech translation), use of an online system (like Google-translate) as auxiliary translation, on-line hypothesis re-calculation in a post-edition interface, etc.

In short, our first attempt in driven decoding consists in adding several feature functions corresponding to the distance between the current hypothesis decoded (called H) and the auxiliary translation available (T) : d(T,H). Different estimation methods to calculate d(T,H) can be proposed : edit-distance, metrics based-on information theory (entropy, perplexity), metrics based on n-gram coverage (BLEU), etc. As a first attempt, we started to experiment in a re-scoring framework for which N-Best hypotheses from the baseline MT system are re-ordered after adding the new feature functions proposed.

3.1. Related Work

This section presents a brief description of related works. They are found mainly in system combination for both speech recognition and machine translation. Unlike speech recognition, system combination in statistical machine translation involves systems based on potentially different standards such as phrasal, hierarchical and syntax based. This introduces new issues such as breaking up of phrases and alterations of word order. We first propose a description of the application of Driven Decoding (DD) algorithm in ASR systems. Then, various system combination attempts in Machine Translation are presented. Detailed presentation of these two concepts - DD and SMT systems combination - is needed to understand our approach.

3.1.1. Imperfect transcript driven speech recognition

In the paper introduced by [6], the authors try to make use of auxiliary textual information associated with speech signals (such as subtitles associated to the audio channel of a video) to improve speech recognition performance. It is demonstrated that those imperfect transcripts which result in misalignments between the speech and text could actually be taken advantage of. In brief, two methods were proposed. The first method involved the combination of generic language model and a language model estimated on the imperfect transcript resulting in cutting down the linguistic space. The second method involved modifying the decoding algorithm by rescoring the estimate function. The probability of the current hypothesis which results from partial exploration of the search graph is dynamically rescored based on the alignment (with imperfect transcript) scores (done using Dynamic Time Warping). The experimental results which used both dynamic synchronization and linguistic rescoring displayed interesting gains. Another kind of imperfect transcript that can be used is the output hypothesis of another system, leading to an integrated approach for system combination. Thus, in the same paper is proposed a method in which the outputs of the contrastive system drives the decoder of the primary system. The results showed that the new system run by driven decoding algorithm outperformed both primary and contrastive systems. Various cross adaptation schemes were also examined. The principle proposed is that firstly, one-best hypothesis is generated from the auxiliary system and a confidence score is evaluated for each word. Then these informations are used to dynamically modify the linguistic score during decoding. The method was evaluated on a radio broadcast transcription task and it was found that WER reduced significantly (about 1.9%) . The WER gain was even better (2.9%) by combining DD and cross adaptation.

3.1.2. System Combination for Machine Translation

-Confusion Network (CN) Decoding

There are important issues to address for machine translation system combination using confusion network decoding. An important one is the presence of errors in the alignment of hypotheses which lead to ungrammatical combination outputs. [7] proposed arbitrary features that can be added log-linearly into the objective function in this method. This addition of new features is the core idea we followed in our proposal.

Confusion Network decoding for MT system combination has been proposed in [8]. The hypothesis have to be aligned using Levenshtein alignment to generate the confusion network. One hypothesis is chosen as skeletal hypothesis and others are aligned against it. In [7], 1-best output from each system is used as the skeleton to develop the confusion network and the average of the TER scores between the skeleton and other hypotheses were used to evaluate the prior probability. Finally a joint lattice is generated by aggregating all the confusion networks parallely. Through this work it is shown that arbitrary features could be added log-linearly by evaluating log-posterior probabilities for each confusing network arc. In confusion network decoding, the word order of the combination is affected by the skeletal hypothesis. Hence the quality of the output from the combination also depends on the skeletal hypothesis. The hypothesis with the minimum average TER-score on aligning with all other hypothesis is proposed as an improved skeletal hypothesis.

$$E_s = \arg\min_{E \in E_i} \sum_{j=1}^{N_s} TER(E_j, E_i) \qquad (1)$$

where N_s is the number of systems and E_s is the skeletal hypothesis.

In [9] system specific confidence scores are also introduced. The better the confidence score the higher the impact of that system. In the experimental part of this same work, three phrase-based (A,C,E), two hierarchical (B,D) and one syntax based (F) systems are combined. All of them are trained on the same data. The decoder weights are tuned to optimize TER for systems A and B and BLEU for the remaining systems. Decoder weight tuning is done on the NIST MT02 task. The results of the combination system were better than single system on all the metrics but for only TER and BLEU tuning. In the case of METEOR tuning, the combination system produced high TER and low BLEU score. The experiments were performed on Arabic and Chinese NIST MT tasks.

-N-Best Concatenation and Rescoring

Another paper [10] presents a slightly different method where N-Best hypotheses are re-scored instead of building a synthesis (CN) of the MT outputs (as described in previous sub-section). The N-Best list from all input systems are combined and then the best hypothesis is selected according to feature scores. Three types of features are: language model features, lexical features, N-Best list based features. The feature weights are modified using Minimum Error Rate Training (MERT). Experiments are performed to find the optimal size for N-Best list combination. Four systems are used and analysed on combination of two best systems and all the systems. 50-best list was found to be optimal size for both cases. The authors showed that the impact of gradually introducing a new system for combination becomes lower as the number of systems increases. Anyway the best result is obtained when all of the systems are combined.

-Co-decoding

Recently, the concept of collaborative decoding (co-decoding) was introduced by [11] to improve machine translation accuracy by leveraging translation consensus between multiple machine translation decoders. Different from what we described earlier (postprocess the n-best lists or word graphs), this method uses multiple machine translation decoders that collaborate by exchanging partial translation results. Using an iterative decoding approach, n-gram agreement statistics between translations of multiple decoders are employed to re-rank full and partial hypotheses explored in decoding.

3.2. Overview of the Driven Decoding Concept

3.2.1. Driven Decoding

As said in the introduction part, driven decoding consists in adding several feature functions to the log-linear model before N-Best list re-ordering. Practically, after N-Best lists are generated by an individual system, additional scores are added to each line of the N-Best list file. Theses additional scores correspond to the distance between the current hypothesis decoded (called H) and the auxiliary translation available (T) : d(T,H). Let's say that 2 auxiliary translations are available (from system 1 and system 2) and that 4 distance metrics are available (BLEU, TER, TERp-A and PER); in that case, 8 scores are added to each line of the N-Best list. The distance metrics used in our experiments are described in the next section and then N-Best reordering process is detailed.

3.2.2. Distance Metrics used

The distance metrics used are Translation Error Rate (TER), Position independent Error Rate (PER), TERp-A and BLEU [12]. The TER score reflects the number of edit operations (insertions, deletions, words substitutions and blocks shifts) needed to transform a hypothesis translation into the reference translation, while the BLEU score is the geometric mean of n-gram precision. Lower TER and higher BLEU score suggest better translation quality. In addition, we use PER score (position independent error rate) which can be seen as a bag-of-words metric potentially interesting in the context of the driven decoding proposed. In addition we use TERp [13] which is an extension of TER eliminating its

shortcomings by taking into account the linguistic edit operations, such as stem matches, synonyms matches and phrase substitutions besides the TER's conventional ones. These additions allow us to avoid categorizing the hypothesis word as Insertion or Substitution in case that it shares same stem, or belongs to the same synonym set represented by WordNet, or is the paraphrase of word in the reference. More precisely, we used TERp-A, another version of TERp, in which each above mentioned edit cost has been tuned to maximize the correlation with human judgment of Adequacy at the segment level (from the NIST Metrics MATR 2008 Challenge development data). However, it is worth mentionning that for this particular task, we use a degraded version of TERp-A which does not take into account synonymy, because the target language is French while the TERp-A metric only implements the use of (English) Wordnet.

3.2.3. N-Best Reordering and Combination

In this framework the system combination is based on the 1000-best outputs (we generally have less on IWSLT data) generated by the LIG primary system using the "uniq" option. Our primary system uses 3 different translation and re-ordering tables. So each N-best list is associated with a set of 38 scores: 1 LM score, 15 translation model scores, 1 distance-based reordering score, 21 lexicalized reordering scores. In addition we introduce 8 distance metrics scores for each sentence.

-The training step

The score combination weights are optimized in order to maximize the BLEU score at the sentence level. This step is performed by using the MERT tool. The weights of "standard" scores are initialized with the tuned weights computed during the usual tuning phase. In a second time, we fine tune weights of the introduced distance metrics (this can be seen as an additional iteration of MERT).

-The decoding step

The decoding step combines all the scores: a global score is computed for each sentence (i.e. the log-linear score) and sentences are reordered according to the final combined score.

4. Use of Confidence Measures for SMT

Besides driven decoding (DD) scores, a sentence confidence score can be added as an additional feature in the N-best list to improve the re-ordering performance. To obtain such a confidence score, a classifier must be constructed. We concatenate two data sets dev2010 + tst2010 to form the training data. Features used to train our model come from the baseline features of the WMT2012 quality estimation shared task (features originally presented in [14]), which can be summarized as follows:

- Source and target sentence: number of tokens and their ratio, number of punctuation marks.
- Source and target sentence's language model probabilities.
- Percentage of unigrams / bigrams / trigrams in quartiles 1 (and 4) of frequency (lower and higher frequency ngrams) in a corpus of the source language.
- Average number of translation per source word in the sentence, unweighted or weighted by the inverse frequency of each word in the source corpus.

The core element needed for the classifier construction process is the training label for each sentence. The TERp-A metric [13], which we select to perform this task, provides the linguistic and semantic matching between each sentence in training set and its reference (available for dev2010 and tst2010 corpora), then yields the minimum cost for matching normalized by its number of tokens as its score. We then categorize them in a binary set: sentences with score higher than 0.3 is assigned with "Good" (G) label, otherwise, "Bad" (B). A CRF-based toolkit, WAPITI [15], is then called to build the classifier. The training phase is conducted using stochastic gradient descent (SGD-L1) algorithm, with values for maximum number of iterations done by the algorithm (-maxiter), stop window size (–stopwin) and stop epsilon (–stopeps) to 200, 6, and 0.00005 respectively.
Applying this classifier in both test sets (test2011 + test2012, with WAPITI's default threshold = 0.5) gives us the result files detailing hypothesized label along with its probability at the sentence level. Then, the confidence score used is the probability of sentence to be regarded as a "Good" sentence. For instance, a sentence classified as "G" with related probability of 0.8 gets obviously the confidence score of 0.8; meanwhile the other one labeled as "B" with probability of 0.7 will have the score of 0.3. This score is used as an additional feature in the log-linear model just as it is done for driven decoding (see previous section).
Performance of the re-ordering task with and without the use of confidence measure will be shown in Table 3.

5. Experimental Results of LIG Systems

We recall that our systems were systematically tuned on dev2010 corpus. Our baseline system, trained as described in section 2, lead to a BLEU score of 30.28 on tst2010 using 2 translation and re-ordering models (no GIGAword) while it improves to 30.80 using 3 translation and reordering models (using GIGAword). This result has to be compared with 27.58 obtained on tst2010 with our system last year.

As far as the driven decoding is concerned, the results show that using the Google 1best hypothesis to guide the

system	dev2010	tst2010	tst2011	tst2012	submission
Baseline (2TM)	27.41	30.28	x	x	
Baseline+GIGAword (3TM)	27.84	30.80	36.88	37.58	**primary**
+DD-google	28.69	**32.01**	39.09	39.36	**contrastive**
+conf	27.84	30.80	x	x	
+DD-google+conf	**28.77**	31.87	x	x	
+DD-ref	32.84	37.26	x	x	oracle
online-google	26.90	33.77	40.16	x	

Table 3: Performances (BLEU case+punct) for several LIG systems

rescoring of the LIG Nbest list leads to significant improvements on all data sets. On dev2010 data, the performance obtained is even better that both LIG and Google systems evaluated separately. On tst2010 and tst2011 the driven decoding is slightly below google. This can be explained by the fact that google has a very different behavior from one set to another (on the dev google is significantly worse than LIG system while he gets better results on tst2011). The LIG system driven by Google 1best was, however, not submitted as a primary run since we used an online system to improve our own module (contrastive system).

On the contrary, adding confidence measures gives only slight improvement on the dev2010 set and does not generalize on tst2010 so it was finally not used in our final submission. According to our analysis, this unsuccessful experiment can be originated from the following reasons: (1) The feature set is simply and superficially constructed hence fails to cover all aspect of quality. This hypothesis can motivate us to explore more types of features (lexical, syntactic, semantic...) in the future work ; (2) the whole combination of features without any selection strategy might be an unskilful option weakening our classifier capability. For information, the oracle obtained, using the golden reference as an auxiliary system, is given in the last line of the table, as well as the performance of the online Google system.

6. Conclusions

This paper described the LIG participation to the E-F MT task of IWSLT 2012. The primary system proposed made a large improvement (more than 3 point of BLEU on tst2010 set) compared to our last year participation. Part of this improvement was due to the use of an extraction from the Gigaword corpus. We have proposed a preliminary adaptation of the driven decoding concept for machine translation. This method allows an efficient combination of machine translation systems, by rescoring the log-linear model at the N-best list level according to auxiliary systems: the basis technique is essentially guiding the search using one or previous system outputs. The results show that the approach allows a significant improvement in BLEU score using Google translate to guide our own SMT system (such system was submitted as contrastive since it uses an online translation). We also tried to use a confidence measure as an additional log-linear feature but we could not get any improvement with this technique.

7. References

[1] P. Koehn, "Europarl: a parallel corpus for statistical machine translation." in *MT Summit X*, Phuket, Thailand, 2005, pp. 79–86.

[2] A. Eisele and Y. Chen, "Multiun: a multilingual corpus from united nation documents." in *LREC 2010*, Valletta, Malta, 2010, pp. 2868–287.

[3] R. Steinberger, A. Eisele, S. Klocek, P. Spyridon, and P. Schlter, "Dgt-tm: A freely available translation memory in 22 languages," in *LREC 2012*, Istanbul, Turkey, 2012.

[4] J. Tiedemann, "News from opus - a collection of multilingual parallel corpora with tools and interfaces," in *Recent Advances in Natural Language Processing*, 2009.

[5] A. Stolcke, "SRILM — an extensible language modeling toolkit," in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, Denver, CO, USA, 2002.

[6] B. Lecouteux, G. Linares, and S. Oger, "Integrating imperfect transcripts into speech recognition systems for building high-quality corpora," *Computer Speech and Language*, vol. 26, no. 2, pp. 67 – 89, 2012.

[7] A.-v. Rosti, S. Matsoukas, and R. Schwartz, "Improved word-level system combination for machine translation," in *In Proceedings of ACL*, 2007.

[8] S. Bangalore, "Computing consensus translation from multiple machine translation systems," in *In Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU-2001*, 2001, pp. 351–354.

[9] A.-v. Rosti, N.-F. Ayan, B. Xiang, S. Matsoukas, R. Schwartz, and B. Dorr, "Combining outputs from multiple machine translation systems," in *In Proceedings of the North American Chapter of the Association*

for Computational Linguistics Human Language Technologies, 2007, pp. 228–235.

[10] A. S. Hildebrand and S. Vogel, "Combination of machine translation systems via hypothesis selection from combined n-best lists," in *Proceedings of Association for Machine Translation in the Americas (AMTA)*, Hawai, USA, 2009.

[11] M. Li, N. Duan, D. Zhang, C.-h. Li, and M. Zhou, "Collaborative decoding: Partial hypothesis re-ranking using translation consensus between decoders," in *Joint ACL IJCNLP*, 2009.

[12] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation." in *ACL*. ACL, 2002.

[13] M. Snover, N. Madnani, B. Dorr, and R. Schwartz, "Terp: A system description," in *Proceedings of the First NIST Metrics for Machine Translation Challenge (MetricsMATR)*, Waikiki, Hawaii, October 2008.

[14] L. Specia, M. Turchi, N. Cancedda, M. Dymetman, and N. Cristianini, "Estimating the sentence-level quality of machine translation systems," in *13th Conference of the European Association for Machine Translation*, Barcelona, Spain, 2009, p. 2837.

[15] T. Lavergne, O. Cappe, and F. Yvon, "Practical very large scale crfs," in *Proceedings ACL*, 2010, p. 504513.

The MIT-LL/AFRL IWSLT-2012 MT System[†]

Jennifer Drexler, Wade Shen,
Terry Gleason

MIT/Lincoln Laboratory
Human Language Technology Group
244 Wood Street
Lexington, MA 02420, USA
{j.drexler,swade,tpg}@ll.mit.edu

Tim Anderson, Raymond Slyh,
Brian Ore, Eric Hansen

Air Force Research Laboratory
Human Effectiveness Directorate
2255 H Street
Wright-Patterson AFB, OH 45433
{first.last}@wpafb.af.mil

Abstract

This paper describes the MIT-LL/AFRL statistical MT system and the improvements that were developed during the IWSLT 2012 evaluation campaign. As part of these efforts, we experimented with a number of extensions to the standard phrase-based model that improve performance on the Arabic to English and English to French TED-talk translation task. We also applied our existing ASR system to the TED-talk lecture ASR task, and combined our ASR and MT systems for the TED-talk SLT task.

We discuss the architecture of the MIT-LL/AFRL MT system, improvements over our 2011 system, and experiments we ran during the IWSLT-2012 evaluation. Specifically, we focus on 1) cross-domain translation using MAP adaptation, 2) cross-entropy filtering of MT training data, and 3) improved Arabic morphology for MT preprocessing.

1. Introduction

During the evaluation campaign for the 2012 International Workshop on Spoken Language Translation (IWSLT-2012) [1] our experimental efforts centered on 1) cross-domain translation using MAP adaptation, 2) cross-entropy filtering of machine translation (MT) training data, and 3) improved Arabic morphology for MT preprocessing.

In this paper we describe improvements over our 2011 baseline systems and methods we used to combine outputs from multiple systems. For a more in-depth description of the 2011 baseline system, refer to [3].

The remainder of this paper is structured as follows. Section 2 presents our work on the MT task, and section 3 presents our work on the automatic speech recognition (ASR) and spoken language translation (SLT) tasks. In section 2 we describe our baseline MT system, the improvements made to that system over the course of this evaluation, the experiments performed to test those improvements, and our evaluation results. In section 3 we describe our existing ASR system that was applied to both the ASR and SLT tasks, and present evaluation results for those tasks.

1.1. IWSLT-2012 Data Usage

We submitted systems for the ASR task, SLT task, and English-to-French and Arabic-to-English MT tasks. In each case, we used data supplied by the evaluation for each language pair for training and optimization. For English-to-French translation, several out-of-domain corpora were used for language model training, phrase table training, and cross-entropy filtering. For Arabic, our systems were strictly limited to the TED training supplied by the evaluation.

We employ a minimum error rate training (MERT) [20] process to optimize model parameters with a held-out development set (`dev2010`). The resulting models and optimization parameters can then be applied to test data during the decoding and rescoring phases of the translation process.

2. Machine Translation

2.1. Baseline MT System

Our baseline system implements a fairly standard SMT architecture allowing for training of a variety of word alignment types and rescoring models. It has been applied successfully to a number of different translation tasks in prior work, including prior IWSLT evaluations. The training/decoding procedure for our system is outlined in Table 1. Details of the training procedure are described in [13].

2.1.1. Phrase Table Training

When building our phrase table, we applied Kneser-Ney discounting [6] to the forward and backward translation probabilities of the phrases extracted during word alignment. In the past, we have combined multiple word alignment strategies, as described in [14]. For the experiments described here, we used only IBM model 5 (see [17] and [18]) for word alignment, to keep the statistics appropriate for discounting.

[†]This work is sponsored by the Air Force Research Laboratory under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

Training Process	
1.	Segment training corpus
2.	Compute GIZA++, Berkeley and Competitive Linking Alignments (CLA) for segmented data [14] [15] [16]
3.	Extract phrases for all variants of the training corpus
4.	Split word-segmented phrases into characters
5.	Combine phrase counts and normalize
6.	Train language models from the training corpus
7.	Train TrueCase models
8.	Train source language repunctuation models
Decoding/Rescoring Process	
1.	Decode input sentences use base models
2.	Add rescoring features (e.g. IBM model-1 score, etc.)
3.	Merge N-best lists (if input is ASR N-best)
4.	Rerank N-best list entries

Table 1: *Training/decoding structure*

Decoding Features
$P(\mathbf{f}\|\mathbf{e})$
$P(\mathbf{e}\|\mathbf{f})$
$LexW(\mathbf{f}\|\mathbf{e})$
$LexW(\mathbf{e}\|\mathbf{f})$
Phrase Penalty
Lexical Backoff
Word Penalty
Distortion
$\hat{P}(\mathbf{E})$ – 6-gram language model
Rescoring Features
$\hat{P}_{rescore}(\mathbf{E})$ – 7-gram LM
$\hat{P}_{class}(\mathbf{E})$ – 7-gram class-based LM
$P_{Model1}(\mathbf{F}\|\mathbf{E})$ – IBM model 1 translation probabilities

Table 2: *Independent models used in log-linear combination*

2.1.2. Language Model Training

During the training process we built n-gram language models (LMs) for use in decoding/rescoring, TrueCasing and repunctuation. In all cases, the MIT Language Modeling Toolkit [19] was used to create interpolated Kneser-Ney LMs. Additional class-based language models were also trained for rescoring. Some systems made use of 3- and 7-gram language models for rescoring trained on the target side of the parallel text.

2.1.3. Optimization, Decoding, and Rescoring

Our translation model assumes a log-linear combination of phrase translation models, language models, etc.

$$\log P(\mathbf{E}|\mathbf{F}) \propto \sum_{\forall r} \lambda_r h_r(\mathbf{E}, \mathbf{F})$$

To optimize system performance we train scaling factors, λ_r, for both decoding and rescoring features so as to minimize an objective error criterion. This is done using a standard Powell-like grid search performed on a development set [20].

A full list of the independent model parameters that we used in our baseline system is shown in Table 2. All systems generated N-best lists that are then rescored and reranked using either a maximum likelihood (ML) or an minimum Bayes risk (MBR) criterion.

These model parameters are similar to those used by other phrase-based systems. For IWSLT, we also add source-target word translation pairs to the phrase table that would not have been extracted by the standard phrase extraction heuristic from IBM model 5 word alignments. These phrases have an additional lexical backoff penalty that is optimized during MERT.

The `moses` decoder [21] was used for our baseline system.

This system serves as the basis for a number of the contrastive systems submitted during this year's evaluation. As described in the following sections, we implemented several techniques for generating improved phrase tables and language models, and experimented with using these techniques both individually and in combination.

2.2. English-To-French Domain Adaptation

During this evaluation we re-examined the approach to cross domain adaptation that we presented in last year's evaluation [3]. Instead of training a single out-of-domain model to adapt to the TED domain, we trained individual models for each available parallel corpus and combined them using hierarchical MAP adaptation [2]. In this technique, models trained on corpora that are more distant from the test domain are successively MAP-adapted with models estimated from less distant corpora, using the following equation:

$$\hat{p_i}(s|t,\lambda) = \frac{N_i(s,t)}{N_i(s,t)+\tau_i} p_i(s|t,\lambda_i) + \frac{\tau_i}{N_i(s,t)+\tau_i} \hat{p_{i+1}}(s|t,\lambda_{i+1}) \quad (1)$$

where $N_i(s,t)$ is the count of the phrase pair (s,t) in model i, $p_i(s|t,\lambda_i)$ is the probability of the source phrase given the target phrase in model i, and $\hat{p_{i+1}}(s|t,\lambda_{i+1})$ is the MAP estimate from the previous step. The final probability estimate for the given phrase pair is $\hat{p}_1(s|t)$. The full hierarchy can be seen in Figure 1.

For the experiments presented here, the ordering of the MAP hierarchy was determined based on the BLEU score of each individual translation model on the held-out TED development set, with low-scoring models adapted towards higher-scoring ones.

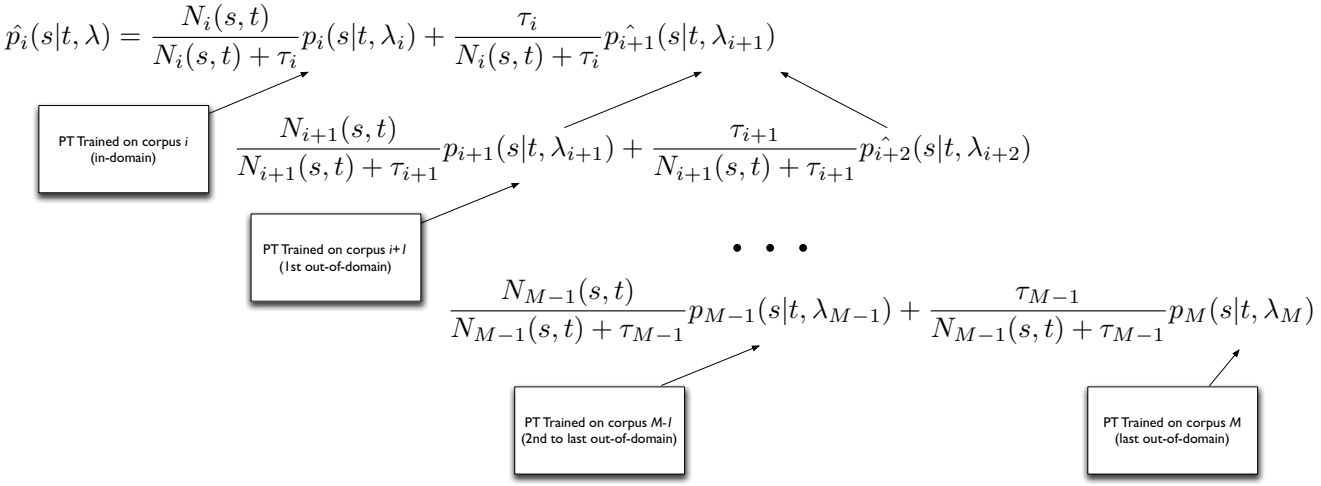


Figure 1: MAP with multiple corpora

2.3. English-To-French Cross-Entropy Filtering

As a comparison to domain adaptation, we experimented with cross-entropy training data filtering, as in [38]. We tested both language model- and translation model-based filtering, but used only LM-based filtering for the experiments performed here, as we found no significant improvement from the inclusion of translation model scores.

We performed LM cross-entropy filtering separately on the parallel portions of the Europarl, Giga-FrEn, News Commentary, and UN corpora. For each of these corpora, for both the source and target sides, we trained a language model on a random subset of the sentences of the same size as the TED training data. We then sorted all sentences in the corpus based on the difference between their cross-entropy given this model and their cross-entropy given the TED language model. We trained new language models on the best $1/64$, $1/32$, $1/16$, $1/8$, $1/4$, and $1/2$ of the corpus. We selected the filter size that produced the language model with the minimum perplexity on the `dev2010` dataset.

To filter the parallel data, we combined the perplexity thresholds that produced the best source and target language models for the `dev2010` dataset. This resulted in the selection of 3.2 percent of the overall data for translation model and language model training, as shown in Table 3.

Two translation models were trained using the filtered parallel data. For the first, which we refer to as A3part, the alignments were generated using all the filtered data but then only the alignments from the TED portion were used to build the translation model. For the second, called TMFilt, the translation model was fully generated from all of the filtered data.

Corpus	Before Filtering	After Filtering
TED	141,387	141,387
Giga-FrEn	24,116,560	824,698
UN	12,886,831	220,066
Europarl	2,007,723	76,554
News Commentary	137,097	1,735
TOTAL	39,289,598	1,264,441

Table 3: *Cross-entropy filtering results in term of number of sentence pairs*

2.4. Alternate French Language Models for Rescoring

Continuous space language model (CSLM) [37], and recurrent neural network language model (RNNLM) [36] were trained on the target side of the TED data. The continuous space language model contained 256 hidden units and an input context of 4 words. The recurrent neural network contained 160 hidden units, 300 classes and backpropagation through time of 4. These language models were used as additional rescoring models on the n-best list. A recurrent neural network language model was also trained on the target side of the bilingual cross-entropy filtered data (RNN-TMfilt). Another language model used for rescoring was the maximum entropy language model(MELM). The 3-gram language model was adapted from a background MELM trained on gigaword and TED data. These models were trained with an extension of the SRILM toolkit.

2.5. Arabic Morphological Processing

In our Arabic-to-English MT systems for prior year evaluations [10, 9, 8, 7, 3], we normalized various forms of alef and hamza and removed the tatweel character and some diacritics before applying a light Arabic morphological analysis procedure that we called AP5. This year, we modified the AP5 procedure to more closely conform to the Arabic Treebank (ATB) segmentation format used in the MADA Arabic morphological analysis, diacritization, and lemmatization system

		Arabic	English
train	Sentences	90,542	
	Running words	1,235,359	1,477,768
	Avg. Sent. length	13.64	16.32
	Vocabulary	46,780	34,447
dev2010	Sentences	934	
	Running words	13,719	17,451
	Avg. Sent. length	14.68	18.68
tst2010	Sentences	507	
	Running words	23,080	26,786
	Avg. Sent. length	13.87	16.10
		English	French
train	Sentences	141,387	
	Running words	2,356,136	2,468,430
	Avg. Sent. length	16.66	17.46
	Vocabulary	41,466	53,997
dev2010	Sentences	934	
	Running words	17,451	17043
	Avg. Sent. length	18.68	18.25
tst2010	Sentences	1664	
	Running words	26,786	27,802
	Avg. Sent. length	16.10	16.71

Table 4: *Corpus statistics for all language pairs*

[4]. In [5], it was shown that the ATB format performed the best of the various MADA segmentation formats tried on the IWSLT 2011 evaluation. In particular, we kept the definite article (Al-) attached to its corresponding noun or adjective. We denote this modified AP5 system as AP5ATBLite.

2.6. MT Experiments

With each of the enhancements presented in prior sections, we ran a number of development experiments in preparation for this year's evaluation. This section describes the development data that was used for each evaluation track, and results comparing the aforementioned enhancements with our baseline system.

2.6.1. Development Data

Tables 4 describes the development and training set configurations used for each language pair in this year's evaluation. We used the WMT-supplied segmenters for preprocessing and normalization, as well as in-house tokenizers for Arabic and French.

2.6.2. English-to-French MT Experiments

We ran a number of baseline and experimental systems on the talk task data set using the methods described in prior sections. In order to perform development experiments, we used supplied development data (`dev2010`) for optimization, and we held out `tst2010` for development testing. Table 5 summarizes the results on the held-out `tst2010` set. For these experiments, the reported scores are an average of ten optimization/decoding runs with different random weight initializations. In all cases we use at at least a 6-gram LM for decoding and rescore with a 7-gram class LM and model1.

Table 5 contains results of our experiments with training data filtering, and with the use of additional language models for rescoring. The three sections of this table show results obtained with three different phrase tables. The first of these, the baseline phrase table, was generated using only the supplied TED training data. The next phrase table, A3Part, was generated using the cross-entropy filtering method described in Section 2.3. Specifically, the word alignments were generated using all of the filtered data, but the phrases were extracted only from the TED data. This phrase table gives an improvement of more than one BLEU point over the baseline. The last phrase table, referred to as TMFilt, was again generated from the filtered data, this time using all of the data for both word alignment and phrase extraction. This phrase table gives an additional improvement of more than half a BLEU point over the A3part phrase table.

Within each section of Table 5, the experiments differ based on their language model configurations. The baseline TED language model was used in all cases. For all except the first line in each section, a language model trained from the monolingual Gigaword data was also used. This language model is a 6-gram language model interpolated by year over the afp portion of the French Gigaword corpus. It adds more than half a BLEU point, regardless of the phrase table it is used with. We also show results using additional language models (CSLM, RNN, MELM) for rescoring. These language models provided little or no additional gain in performance, and in one case reduced the overall gain.

System	`tst2010`
TED Models Only (baseline)	32.06
TED PT + InterpGiga LM	32.61
A3part	33.16
A3part + InterpGiga LM	33.80
A3part + InterpGiga LM + RNN	33.57
A3part + InterpGiga LM + MELM	33.79
A3part + InterpGiga LM + CSLM	33.91
A3part + InterpGiga LM + CSLM + RNN-TMfilt	33.83
TMFilt	33.71
TMFilt + InterpGiga LM	34.22
TMFilt + InterpGiga LM + RNN	34.26
TMFilt + InterpGiga LM + MELM	34.35
TMFilt + InterpGiga LM + CSLM	34.40
TMFilt + InterpGiga LM + CSLM + RNN-TMfilt	34.24

Table 5: *Summary of English-French filtering experiment results*

Table 6 contains results from our domain adaptation experiments. The MAP phrase table was produced through

hierarchical MAP adaptation of phrase tables trained with the following parallel corpora (in order): News Commentary, Europarl, Giga-FrEn, and TED. On its own, this phrase table improves the baseline score by about half a BLEU point. We combined our phrase table domain adaptation with language models that were trained individually on each parallel corpus and included in the log-linear model. Using these language models adds an additional half BLEU point to our scores.

System	tst2010
TED Models Only (baseline)	32.06
TED PT + Parallel LMs	32.58
MAP	32.60
MAP + Parallel LMs	33.27

Table 6: *Summary of English-French domain adaptation experiment results*

The overall best result was achieved with the TMFilt phrase table, when combined with rescoring using a CSLM language model. This score, 34.40, represents a gain of 2.34 BLEU points over the baseline score of 32.06. Unfortunately, the TMFilt phrase table results were generated too late to be included in the evaluation. At submission time, our best individual system used the same configuration, but with the A3Part phrase table instead of the TMFilt phrase table, for an average BLEU score of 33.91.

As described in section 2.7, we were able to combine our domain adaptation system with one of our filtering systems to produce a better result than any of the individual systems available at submission time. In the future, we plan to experiment with ways of combining the best techniques from domain adaptation and filtering into a single system, rather than relying on system combination.

2.6.3. Arabic-To-English MT Experiments

Table 7 shows the mean BLEU scores for individual Arabic-to-English MT systems trained on the 2011 and 2012 training data and tested on the tst2010 data versus the morphology segmentation system. For both the 2011 and 2012 training data, the AP5ATBLite system performs slightly better than the AP5 system. Also, the extra training data in the 2012 system provides approximately one BLEU point of improvement over the systems trained on the 2011 data.

Table 7: *Mean BLEU scores for individual Arabic-to-English MT systems tested on the tst2010 data versus morphology segmentation system and year of training data.*

Morphology System	Training Data	
	2011	2012
AP5	21.13	22.24
AP5ATBLite	21.57	22.45

In addition to the AP5ATBLite modification, we investigated the use of Kneser-Ney (KN) phrase table smoothing [6] using the AP5ATBLite system trained on the 2012 training data. The combination of AP5ATBLite and KN smoothing yielded a mean BLEU score of 23.60 compared to the mean of 22.45 for the AP5ATBLite system without phrase table smoothing.

2.7. MT Submission Summary

As part of this year's evaluation we experimented with training data filtering, improved cross-domain adaptation, and improved Arabic morphological processing. These developments have helped to improve our system when compared with our 2011 system.

The overall submitted Arabic-to-English system was a combination of individual component systems that were each the best in terms of BLEU score after ten MERT optimization runs. Two of the component systems were (1) the best AP5ATBLite system (with no phrase table smoothing) and (2) the best AP5ATBLite system with KN phrase table smoothing.

The majority of our English-To-French submissions are also combinations of multiple systems. Our primary submission is a combination of the *MAP + Parallel LMs* system and the *A3part + InterpGiga LM + MELM* system. We also submitted the individual system that had the best single MERT run, in terms of BLEU score on the tst2010 data set, which was a run of the *A3part + InterpGiga LM + CSLM + RNN-TMfilt* system.

Table 8 summarizes each of the systems submitted for this year's evaluation and how they compare with our 2011 submission (when applicable) on the tst2011 and tst2012 data sets. Due to a de-tokenization error, our official English-to-French submissions had much lower scores; the scores reported here reflect the performance of our system after the correction of that error.

3. Automatic Speech Recognition and Spoken Language Translation

3.1. ASR System

Acoustic models were developed using the same TED data and training procedure as our IWSLT 2011 system [3]. In addition to training models using Perceptual Linear Prediction (PLP) features, we trained a second set of acoustic models using Mel-Frequency Cepstral Coefficients (MFCCs).

Cross-entropy difference scoring [35] was used to select subsets of the Europarl, Gigaword, news 2007–2011, and news commentary texts for training the language models. The provided TED training data was used for the in-domain text, and the selection threshold for each out-of-domain data set was chosen to minimize the perplexity on dev2010. This process selected 7.3% of the data for LM development.

The SRILM Toolkit[1] was used to estimate interpolated

[1] Available at: http://www.speech.sri.com/projects/srilm

Arabic-to-English Systems			
System	*Features*	tst2011	tst2012
AE-primary 2011	2011 combined system	19.56	N/A
AE-primary	2012 primary combination	17.99	19.30
AE-contrast1	2012 contrast1	17.28	18.36
English-to-French Systems			
System	*Features*	tst2011	tst2012
EF-primary 2011	2011 best system	34.19	N/A
EF-primary	2012 primary combination	36.10	37.32
EF-contrast1	2012 best individual system	36.16	36.75
EF-contrast2	2012 best combination	36.39	37.10

Table 8: *Summary of submitted 2012 MT systems*

trigram and 4-gram LMs for decoding and rescoring, respectively. Recurrent Neural Network Maximum Entropy (RNNME) LMs [36] were developed for rescoring using the RNNLM Toolkit.[2] One RNNME LM was trained on Gigaword, and a second RNNME LM was trained on news 2007–2011. As suggested in [39], the number of classes was set to 300 and 4-gram features were used for the ME model. Each network included 160 hidden units, which was selected to minimize the perplexity on the TED training data.[3] The vocabulary for the LMs included 95,000 words.

Recognition lattices were produced using the same procedure as last year [3], and 1000-best lists were extracted for rescoring with the 4-gram and RNNME LMs. The scores from each LM were linearly interpolated using weights chosen to minimize the perplexity on the development partitions. The final transcripts were produced by combining the MFCC and PLP systems using a Confusion Network Combination system (CNC).[4]

Our implementation of CNC starts by creating confusion networks for each recognizer's rescored N-best list. These confusion networks are then aligned to each other using a time-weighted Levenschtein distance computed over the max posterior hypothesis per recognizer. The resulting alignment is used to merge columns of each individual confusion network into a single confusion network, where language model and acoustic model scores for each recognizer's hypotheses are combined in a log-linear way, with weights for each system and each individual model. System weights were set through a Powell-like grid search using the supplied development data.

Table 9 shows the Word Error Rates (WERs) obtained on the IWSLT dev2010 and tst2010partitions. According to the unofficial results, the submitted system yielded a 12.6% WER on tst2011 and a 14.3% WER on tst2012.

[2]Available at: http://www.fit.vutbr.cz/∼imikolov/rnnlm

[3]Due to time constraints we only compared networks with 80, 120, and 160 hidden units.

[4]Due to a bug in the submitted system, the submitted combination did not result in significant differences between the PLP baseline and the submitted combination. This was due to an error in setting the prior weight per system.

	dev2010		tst2010	
	MFCC	PLP	MFCC	PLP
1st pass	19.0	18.3	18.7	17.9
2nd pass	16.6	16.5	15.4	15.0
4-gram	15.3	15.4	14.1	13.9
4-gram + RNNME	14.4	14.4	13.0	12.5
CN combination	13.7		12.9	

Table 9: WERs obtained on the IWSLT dev2010 and tst2010 partitions using the MFCC and PLP systems.

3.2. SLT System

For the SLT task, we used a combination of the ASR and MT systems described above. We used only ASR input from our own system.

3.3. SLT Submission

Table 10 summarizes the results of our submission for the SLT tasks. Our official SLT evaluation scores were impacted by the same de-tokenization error that lowered our English-to-French MT scores. Again, these scores reflect the performance of our system once that error was corrected.

System	tst2011	tst2012
Primary	27.82	27.54
Contrastive	27.52	27.51

Table 10: *Summary of submitted 2012 SLT systems*

4. Acknowledgments

We would also like to thank Katherine Young and Jeremy Gwinnup for their help in processing the English-French and TED task data sets and the staff of the Human Language Technology group at MIT Lincoln Lab for making machines available for this evaluation effort.

5. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, S. Stüker "Overview of the IWSLT 2012 Evaluation Campaign," In *Proc. of IWSLT*, Hong Kong, HK, 2012.

[2] A. R. Aminzadeh, J. Drexler, T. Anderson, and W. Shen, "Improved Phrase Translation Modeling Using MAP Adaptation," in *Proceedings of TSD 2012* (Brno, Czech Republic), September 2012.

[3] A. R. Aminzadeh, T. Anderson, R. Slyh, B. Ore, E. Hansen, W. Shen, J. Drexler, and T. Gleason, "The MIT-LL/AFRL IWSLT-2011 MT system," in *Proceedings of IWSLT 2011,* (San Francisco CA), December 2011.

[4] R. Roth, O. Rambow, N. Habash, M. Diab, and C. Rudin, "Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking," in *Proceedings of ACL-08: HLT, Short Papers,* (Columbus OH), June 2008.

[5] J. Wuebker, M. Huck, S. Mansour, M. Freitag, M. Feng, S. Peitz, C. Schmidt, and H. Ney, "The RWTH Aachen machine translation system for IWSLT 2011," in *Proceedings of IWSLT 2011,* (San Francisco CA), December 2011.

[6] G. Foster, R. Kuhn, and H. Johnson, "Phrasetable smoothing for statistical machine translation," in *Proceedings of EMNLP 2006,* (Sydney, Australia), July 2006.

[7] Shen, Anderson, T., Slyh, R., and Aminzadeh, A.R., "The MIT-LL/AFRL IWSLT-2010 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Paris, France, 2010.

[8] Shen, W., Delaney, B., Aminzadeh, A.R., Anderson, T., and Slyh, R. "The MIT-LL/AFRL IWSLT-2009 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Tokyo, Japan, 2009.

[9] Shen, W., Delaney, B., Anderson, T., and Slyh, R. "The MIT-LL/AFRL IWSLT-2008 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Honolulu, HI, 2008.

[10] Shen, W., Delaney, B., Anderson, T., and Slyh, R. "The MIT-LL/AFRL IWSLT-2007 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Trento, Italy, 2007.

[11] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," In Proc. of MT Summit, 2005.

[12] Munteanu, D. S. and Marcu, D., "ISI Arabic-English Automatically Extracted Parallel Text," Linguistic Data Consortium, Philadelphia, 2007.

[13] Shen, W., Delaney, B., and Anderson, T. "The MIT-LL/AFRL IWSLT-2006 MT System," In Proc. Of the International Workshop on Spoken Language Translation, Kyoto, Japan, 2006.

[14] Chen, B. et al, "The ITC-irst SMT System for IWSLT-2005," In Proc. Of the International Workshop on Spoken Language Translation, Pittsburgh, PA, 2005.

[15] Melamed, D., "Models of Translational Equivalence among Words," In Computational Linguistics, vol. 26, no. 2, pp. 221-249, 2000.

[16] Liang, P., Scar, B., and Klein, D., "Alignment by Agreement," Proceedings of Human Language Technology and North American Association for Computational Linguistics (HLT/NAACL), 2006.

[17] Brown, P., Della Pietra, V., Della Pietra, S. and Mercer, R. "The Mathematics of Statistical Machine Translation: Parameter Estimation," Computational Linguistics 19(2):263–311, 1993.

[18] Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, I.D., Och, F.J., Purdy, D., Smith, N.A., Yarowsky, D., "Statistical machine translation: Final report," In Proceedings of the Summer Workshop on Language Engineering at JHU, Baltimore, MD 1999.

[19] Bo-June (Paul) Hsu and James Glass, "Iterative Language Model Estimation: Efficient Data Structure and Algorithms," In Proc. Interspeech, 2008.

[20] Och, F. J., "Minimum Error Rate Training for Statistical Machine Translation," In ACL 2003: Proc. of the Association for Computational Linguistics, Japan, Sapporo, 2003.

[21] Koehn, P., et al, "Moses: Open Source Toolkit for Statistical Machine Translation," Annual Meeting of the Association for Computational Linguistics (ACL), Prague, Czech Republic, June 2007.

[22] K. Oflazer and I. Kuruoz, "Tagging and morphological disambiguation of Turkish text," In Proceedings of the 4th Conference on Applied Natural Language Processing, Stuttgart, Germany, 1994.

[23] Mermer, C., Kaya, H., and Dogan, M.U. "The TUBITAK-UEKAE Statistical Machine Translation System for IWSLT 2007," In Proc. of IWSLT, 2007.

[24] Matusov, E. and Ueffing, N. and Ney, H., "Computing Consensus Translation from Multiple Machine Translation Systems Using Enhanced Hypotheses Alignment," In Proc. of EACL, 2006.

[25] Fiscus, JG, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," In Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, 1997.

[26] Snover, M. and Dorr, B. and Schwartz, R. and Micciulla, L. and Makhoul, J., "A study of translation edit rate with targeted human annotation," In Proc. of AMTA, 2006.

[27] Rosti, A.V.I. and Matsoukas, S. and Schwartz, R., "Improved Word-Level System Combination for Machine Translation," In Proc. of ACL, 2006.

[28] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki "Online large-margin training for statistical machine translation," In Proc. of EMNLP-CoNLL, 2007.

[29] D. Chiang Y. Marton, and P. Resnik, "Online large-margin training of syntactic and structural translation features," In Proc of EMNLP, 2008.

[30] D. Chiang, K. Knight, W. Wang, "11,001 new features for statistical machine translation," In Proc. NAACL/HLT, 2009.

[31] D. Graff, J. Garofolo, J. Fiscus, W. Fisher, and D. Pallett, "1996 English Broadcast News Speech (HUB4)," *Linguistic Data Consortium*, Philadelphia, 1997. Available: http://www.ldc.upenn.edu

[32] J. Fiscus, J. Garofolo, J. Fiscus, M. Przybocki, W. Fisher, and D. Pallett, "1997 English Broadcast News Speech (HUB4)," *Lingustic Data Consortium*, Philadelphia, 1998. Available: http://www.ldc.upenn.edu

[33] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly Supervised and Unsupervised Acoustic Model Training," *Computer Speech and Language*, vol. 16, pp. 115–129, 2002.

[34] M. Bisani and H. Ney, "Joint-Sequence Models for Grapheme-to-Phoneme Conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, May 2008.

[35] R. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," *Association Computational Linguists 2010 Conference Short Papers*, Uppsala, Sweden.

[36] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Černocký, "Strategies for Training Large Scale Neural Network Language Models," in *Proc. Automatic Speech Recognition and Understanding Workshop*, Hawaii, USA, 2011.

[37] Schwenk, Holger, "Continuous Space Language Models," in *Computer Speech and Language*, vol 21, 492-518, 2007.

[38] S. Mansour *et al.*, "Combining Translation and Language Model Scoring for Domain-Specific Data Filtering," in *Proc. International Workshop on Spoken Language Translation*, San Francisco, USA, 2011.

[39] Mikolov Tom, Karafit Martin, Burget Luk, ernock Jan, Khudanpur Sanjeev, "Recurrent neural network based language model," In: *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, Makuhari, Chiba, JP, 2010.

Minimum Bayes-Risk Decoding Extended with Similar Examples: NAIST-NICT at IWSLT 2012

Hiroaki Shimizu[1,2], Masao Utiyama[2], Eiichiro Sumita[2], Satoshi Nakamura[1]

[1]Nara Institute of Science and Technology (NAIST), Nara, Japan
[2]National Institute of Information and Communication Technology (NICT)
Hikaridai 2-2-2, Keihanna Science City, 619-0288 Kyoto, Japan
hiroaki-sh@is.naist.jp

Abstract

This paper describes our methods used in the NAIST-NICT submission to the International Workshop on Spoken Language Translation (IWSLT) 2012 evaluation campaign. In particular, we propose two extensions to minimum bayes-risk decoding which reduces a expected loss.

1. Introduction

Minimum Bayes-Risk (MBR) decoding has been proposed for statistical machine translation (SMT) to minimize expected loss of translation errors under loss functions that measure translation performance (Kumar and Byrne, 2004). Those loss functions are the inverse of evaluation metrics like BLEU (Papineni et al., 2001) and NIST (Doddington, 2002).

MBR outputs translations that are similar to the other translations in the n-best list, as this reduces the expected loss if one of these other translations is actually the correct answer (see Section. 2 for details).

We extend the MBR decoding with two methods: considering similarity of each translation to the Maximum A Posteriori (MAP) translation and using training sentences pairs that is similar to the input sentence for decoding.

The proposed methods are used in the NAIST-NICT system for the International Workshop on Spoken Language Translation (IWSLT) 2012 evaluation campaign. We participated in the OLYMPICS Task, which is from Chinese to English.

2. Minimum Bayes-Risk decoding

2.1. MAP decision rule

MAP decoding finds the most likely translation $\hat{E}$ from translation candidate E_j given the input sentence F. The MAP translation of F is defined by

$$\hat{E} = \arg\max_{E_j} P(E_j|F) \tag{1}$$

This is the traditional decision rule.

2.2. MBR decision rule

Let F and E be the source and target sentences, the MBR decoding is defined as follows.

$$\hat{E} = \arg\min_{E_i} \sum_{E_j} L(E_j, E_i) P(E_j|F) \tag{2}$$

E_i is the i-th output sentence of the n-best translations. $P(E_j|F)$ is the probability of translation E_j given F. $L(E_j, E_i)$ is the loss function. This loss function will be defined in Section 2.3.

Note that $P(E_j|F)$ can be scaled by

$$P(E_j|F) = \frac{\exp\left(\alpha H(E_j, F)\right)}{\sum_{E_k} \exp\left(\alpha H(E_k, F)\right)} \tag{3}$$

$H(\cdot,\cdot)$ is the weighted overall score. The scaling factor α lies in $[0, \infty)$. If α is smaller, P becomes equal. If α is larger, P becomes uneven.

2.3. BLEU

We use BLEU as the loss function. BLEU is defined by

$$BLEU(E_j, E_i) = \mathrm{BP} \times \exp\left(\frac{1}{4}\sum_{n=1}^{4} \log p_n(E_j, E_i)\right) \tag{4}$$

where p_n is the n-gram precision of E_j given E_i as the reference. BP is the brevity penalty.

The loss function is defined by

$$L(E_j, E_i) = 1 - BLEU(E_j, E_i) \tag{5}$$

We use a sentence-level BLEU score (Papineni et al., 2001). To solve the problem that no matches make the sentence-level BLEU score zero, we add one count to the n-gram hit and total n-gram count for $n > 1$ (C. Lin et al., 2004). We use the sentence-level BLEU score only for MBR decoding and normal BLEU score for the translation results.

By the way, if the loss function is defined as follows, (2) is as same as (1).

$$L(E_j, E_i) = \begin{cases} 1 & E_j = E_i \\ 0 & otherwise \end{cases} \tag{6}$$

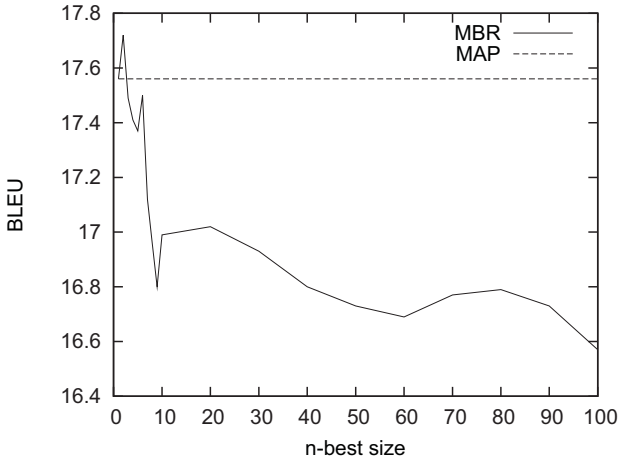


Figure 1: BLEU of MAP and MBR for development test set

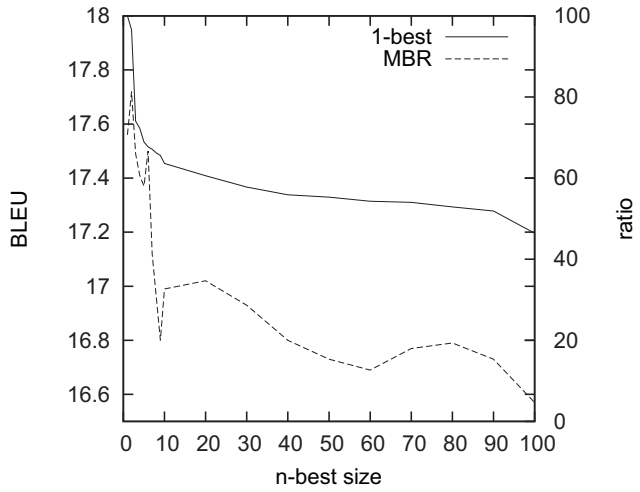


Figure 2: The relation between the percentage of the MAP in MBR and BLEU of MBR. The line "1-best" represents the percentages of MAP translations that used as MBR translations

MAP decoding is a special case of MBR translation where the loss function is defined as the 0-1 loss function formulation.

3. Considering similarity to MAP

In this Section, we introduce our first proposed method.

3.1. Observation

The motivation for the first proposed method is that we have noticed that the BLEU scores of the MBR translations were unstable with regards to different sizes of the n-best output.

Figure 1 shows to what extent the quality of the MBR translation depends on the n-best size. As shown in the figure, the BLEU of MBR translations for $n = 2$ was better than that of MAP translations on the development test set. However, $n \neq 2$ were inferior to those of MAP translations.

This observation made us conjecture that we need some modification to standard MBR decoding.

3.2. Proposed method 1

We conjecture that considering the similarity to the MAP translation is useful to obtain better translations, as the MAP translation is generally better than the other translations. The dotted line in Figure 2 indicates the BLEU scores of MBR translations. The line "1-best" represents the percentages of MAP translations that used as MBR translations. As shown in the figure, both lines gradually decreases as the n-best size increases. This means that when the BLEU scores of MBR translations are high, many of the MAP translations are adopted in MBR translations.

To consider the similarity to the MAP translations, we propose a method that limits the possible translations chosen by MBR to those above a certain similarity to the MAP translation. In other words, we only choose candidates that satisfy the following constraint.

$$BLEU(E_{MAP}, E_i) \geq B_1 \quad (7)$$

where E_{MAP} is the MAP translation and B_1 is a threshold which indicates the similarity to the MAP translation. The sentence-level BLEU score (Papineni et al., 2001) also measures the similarity of MAP translations, because the loss function applied BLEU and the translation results are also measured by BLEU. Note that the n-best size, α and B_1 are decided by the development test set.

4. Using training data for MBR decoding

In this section, we introduce the second proposed method.

4.1. Motivation

Nearest neighbors choose the data which is nearest input data. Nearest neighbors of the source sentences have been used for tuning parameters (Utiyama et al. 2009, Liu et al. 2012). The second method uses nearest neighbors not for tuning but for reranking.

4.2. Proposed method 2

We extended MBR decoding by referencing nearest neighbors. This method is that when we use MBR decoding, we make use of training sentence pairs that are similar to the input sentence. In particular, we use nearest neighbor to improve the probability estimate $P(E_j|F)$. This can be done by

Table 1: Corpus statistics

		Chinese	English
Training	Sentences	75,552	
	words	675,602	739,246
Tuning	Sentences	1,007	
	words	5,973	10,413
Development test	Sentences	1,050	
	words	5,840	10,364

defining the probability of $P(E_j|F)$ in the following form.

$$\begin{aligned} P(E_j|F) &= \sum_{F_k \in \xi} P(E_j, F_k|F) \\ &= \sum_{F_k \in \xi} P(E_j|F_k, F)\ P(F_k|F) \\ &= \sum_{F_k \in \xi} P(E_j|F_k)\ P(F_k|F) \end{aligned} \quad (8)$$

ξ is defined by

$$\xi = \{G : BLEU(G, F) \geq B_2\} \cup (G \in \mathcal{F}) \quad (9)$$

where ξ is a collection of input training sentences above a certain similarity to the input sentence. $\mathcal{F}$ is a collection of input training sentences and input sentence F. B_2 is a threshold which indicates the similarity to the input sentence F. We use sentence-level BLEU.

We interpret the probability of $P(F_k|F)$ as the probability that the input sentence F can be changed to F_k. To use sentence-level BLEU, we define $P(F_k|F)$ as

$$P(F_k|F) = \frac{BLEU(F_k, F)}{\sum_{F_l \in \xi} BLEU(F_l, F)} \quad (10)$$

$P(E_j|F_k)$ is defined by

$$P(E_j|F_k) = \begin{cases} 1 & F_k \neq F\ and\ E_j = F_k \\ 0 & F_k \neq F\ and\ E_j \neq F_k \\ (3) & F_k = F \end{cases} \quad (11)$$

where the probability is as same as normal MBR decoding as (3) in $F_k = F$.

The advantage of this method is making use of more information. As this method uses not only n-best list of input sentence but also training sentences pairs which is similar to input sentence.

This method can be combined with Equation (7) to constrain the candidates. We call it "proposed method 1+2".

5. Results

5.1. Experiment conditions

For building the translation system, we used the phrase-based Moses (Koehn et al., 2007) decoder. We used GIZA++ (Och and Ney, 2003) for word alignment and SRILM (Stolcke, 2002) for 5-gram language model. Minimum Error Rate Training (Och, 2007) was used for tuning. We used the Stanford word segmenter (Tseng et al., 2005) for Chinese segmentation. We used the Peking University (PKU) Stanford models.

We use OLYMPIC Task data: HIT (HIT Olympic Trilingual Corpus) and BTEC (Basic Travel Expression Corpus). For the training data, we used IWSLT_BTEC.train.*, IWSLT12_BTEC.devset* and IWSLT12_HIT.train.*. For the tuning data, we used IWSLT12_HIT.devset2_IWSLT12.*, and we used IWSLT12_HIT.devset1_IWSLT12.* for the development test set data. Statistics computed over these data sets are reported in Table 1.

Table 2: The development and official BLEU score. Moses default n-best size is 200 when we use Moses MBR decoding. n=2 is the development set optimal value.

	Dev	Official
MAP (baseline)	17.56	17.29
MBR (n=200)	16.53	17.72
MBR (n=2)	17.72	17.30
Proposed 1 (non-scale)	17.81	16.96
Proposed 1	17.96	16.79
Proposed 2	17.82	17.45
Proposed 1+2	17.67	17.39

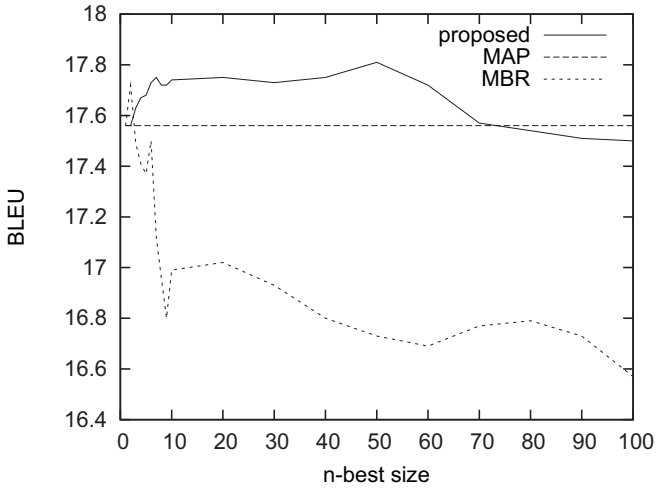


Figure 3: MAP, MBR and proposed BLEU score of development test set

5.2. Development test set

Table 2 shows the development BLEU score. Moses default n-best size is 200 when we use Moses MBR decoding. n=2 is the development set optimal value. When using the MAP similarity threshold over the development test set (proposed

1), we use the following parameters: the MBR scaling factor is 5, the MAP similarity threshold $B_1$is 0.75, and the n-best size is 50.

Figure 3 shows the proposed method is stable and does not depend on the size of the n-best list. When $n = 50$, we can obtain the best BLEU score. If the similarity is less than 0.75, the graph looks like MBR and many translations are normal MBR translations, because similarity to MAP is not considered. If the similarity is more than 0.75, the graph looks like MAP and many of translations are MAP translation. As the limiting condition becomes very severe. The scaling factor 5 is the optimal value for the development test set.

Table 2 shows the BLEU of the nearest neighbor method also uses on the development test set (proposed 2). We use the following parameters: MBR scaling factor is 5, the similarity to the input sentence B_2 is 0.7 and n-best size is 50.

5.3. Official test set

Table 2 also shows the official test set result. The proposed method 1 is not better than the baseline by 0.5. However, the proposed method 2 is better than the baseline by 0.16. The proposed 1+2 is better than the baseline by 0.1.

6. Discussion

First, we discuss about the wrong result of the proposed 1. It can be seen that the results for the MBR decoding for development test set is not good. One of the reasons is that many of MAP translations are good so considering the similarity to MAP translation worked well. However, MBR decoding for official test set is good. We guess that most MAP translations did not have as good quality, and standard MBR translation is stable in the size of the n-best list. So, considering similarity to MAP translation made the result worse. We guess that the proposed method 1 is a valid method for data in which MBR decoding does not have a positive effect. In addition, we use Word Error Rate (WER) for the loss function (5). However the result of WER is worse than that of BLEU.

The proposed method 2 is effective for this official test set. However proposed method 1+2 is not better than method as method 1 is not effective.

7. Conclusion

We participated in the OLYMPICS Task. Our system extended MBR decoding with two methods. The method using training data for MBR decoding improved BLEU scores.

8. Acknowledgement

We thank Graham Neubig for his comments on this paper.

9. References

[1] G. Doddington. 2002. Automatic Evaluation of Machine Translation Quality using N-gram Co-occurrence Statistics. In Proceedings of Human Language Technology Conference.

[2] S. Kumar and W. Byrne. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In Human Language Technologies: North American Chapter of the Association for Computational Linguistics, pages 169–176, Boston, MA, USA.

[3] K. Papineni, S. Roukos, T. Ward and W. Zhu. 2001. BLEU: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176 (W0109–022) , IBM Research Division.

[4] L. Chin-Yew and F. Och. 2004. ORANGE : a Method for Evaluating Automatic Evaluation Metric for Machine Translation. In Proceedings of the 20th International Conference on Computational Linguistics, Geneva, Switzerland.

[5] M. Utiyama, H. Yamamoto and E. Sumita. 2009. Two Methods for Stabilizing MERT: NICT at IWSLT 2009. In proceedings of IWSLT, page 79–82.

[6] L. Liu, H. Cao, T. Watanabe, T. Zhao, M. Yu and C. Zhu. 2012. Locally Training the Log-Linear Model for SMT. In Proceedings of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, page 402–411.

[7] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C.Dyer, O. Bojar, A. Constantin and E. Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, page 177–180.

[8] F. J. Och and H. Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. In Proceedings of Computational Linguistics, vol. 29, No.1 , page 19–51.

[9] A. Stolcke. 2003. SRILM - An Extensible Language Modeling Toolkit. In Proceedings of International Conference on Spoken Language Processing.

[10] F. J. Och. 2007. Minimum Error Rate Training in Statistical Machine Translation. In Proceedings of Association for Computational Linguistics.

[11] H. Tseng, P. Chang, G. Andrew, D. Jurafsky and C. Manning. 2005. A Conditional Random Field Word Segmenter. In Proceedings of Fourth SIGHAN Workshop on Chinese Language Processing.

The NICT Translation System for IWSLT 2012

Andrew Finch† *Ohnmar Htun*‡ *Eiichiro Sumita*†

† Multilingual Translation Group
MASTAR Project
National Institute of Information and
Communications Technology
Kyoto, Japan
andrew.finch,eiichiro.sumita@nict.go.jp

‡ Dept. of Management and
Information System Science
Nagaoka University of Technology
Nagaoka, Japan
s097001@stn.nagaokaut.ac.jp

Abstract

This paper describes NICT's participation in the IWSLT 2012 evaluation campaign for the TED speech translation Russian-English shared-task. Our approach was based on a phrase-based statistical machine translation system that was augmented by using transliteration mining techniques.

The basic premise behind our approach was to try to use sub-word-level alignments to guide the word-level alignment process used to learn the phrase-table. We did this by first mining a corpus of Russian-English transliterations pairs and cognates from a set of interlanguage link titles from Wikipedia. This corpus was then used to build a many-to-many nonparametric Bayesian bilingual alignment model that could be used to identify the occurrence of transliterations and cognates in the training corpus itself. Alignment counts for these mined pairs were increased in the training corpus to increase the likelihood that these pairs would align in training. Our experiments on the test sets from the 2010 and 2011 shared tasks, showed that an improvement in BLEU score can be gained in translation performance by encouraging the alignment of cognates and transliterations during word alignment.

1. Introduction

In the IWSLT 2012 evaluation campaign [1], the NICT team participated in TED [2] speech translation shared-task for Russian-English. This paper describes the machine translation approach adopted for this campaign.

Our overall approach was to take a phrase-based statistical machine translation decoder and increase its performance by improving the word alignment. Typically only word co-occurrence statistics are used in determining the word-to-word alignments during training, however certain classes of words can offer additional features that can be used to assist in the prediction of their alignment: these words are transliterations and cognates. Transliterations are words that have been borrowed from another language; loan words imported into the language while preserving their phonetics as far as possible. So for example, the Italian name 'Donatello' would be transcribed into the Cyrillic alphabet as 'Донателло' (DONATELLO). The upper case form in parentheses is a romanized form of the preceding Russian character sequence, which in this case is exactly the same as the original English word, but in general this is not necessarily the case.

Cognates are words that share a common etymological origin, for example the word 'milk' in English is a cognate of the German word 'milch' and the Russian word 'молоко' (MOLOKO). Transliterations are derived directly from the word in the language from which they are being borrowed, and cognates are both derived from their common root. Our hypothesis is that these relationships can be modeled and thereby detected in bilingual data. Our approach is to model both cases using a generative model, under the assumption that there exists some generative process that can reliably assign a higher generation probability to cognates and transliterations than a model designed to explain random pairs of words. Furthermore, we assume that if two words are assigned a relatively high probability from such a model, then they are likely to be aligned in the data. This assumption is not true in general due to the existence of false cognates; words may appear to be cognates, when in fact there is no genetic relationship between them. Nonetheless, we anticipate that pathological occurrences of this kind will be rare, and that relying on the assumptions mentioned earlier will result an overall benefit.

Due to an unfortunate error in the processing of the phrase-tables of our systems for the final submission to the shared task, the official scores for our system are several BLEU points below what could be expected of the system had there been no error, we therefore do not report the official results for our system on the 2012 test data, but instead rely on experiments based on systems trained on the 2012 training set, and tested on the 2010 and 2011 test sets.

The overall layout of our paper is as follows. In the next section we describe the underlying phrase-based statistical

machine translation system that forms the basis of all of the systems reported in this paper. In the following section we describe the techniques we used to incorporate information from sub-word alignments into the word alignment process. Then we present our experiments comparing our system to a baseline system. Finally we conclude and offer some directions for future research.

2. The Base System

2.1. Decoder

The decoder used in these experiments is an in-house phrase-based statistical machine translation decoder OCTAVIAN than can operate in a similar manner to the publicly available MOSES decoder [3]. The base decoder used a standard set of features that were integrated into a log-linear model using independent exponential weights for each feature. These features consisted of: a language mode; five translation model features; a word penalty; and a lexicalized re-ordering model with monotone, discontinuous, swap features for the current and previous phrase-pairs.

Based on a set of pilot experiments we decoded with a maximum distance of 5 on the distances phrases could be moved in the re-ordering process during decoding.

2.2. Pre-processing

The English data was tokenized by applying a number of regular expressions to separate punctuation, and split contractions such as "it's" and "hasn't" into two separate tokens. We also removed all case information from the English text to help to minimize issues of data sparseness in the models of the translation system. All punctuation was left in both source and target. We took the decision to generate target punctuation directly using the process of translation, rather than as a punctuation restoration step in post processing based on experiments carried out for the 2010 IWSLT shared evaluation [4].

2.3. Post-processing

The output of the translation system was subject to the following post-processing steps which were carried out in the order in which that are listed.

1. Out of vocabulary words (OOVs) were passed through the translation process unchanged, some of these OOVs were Russian and some English. We took the decision to delete only those OOVs containing cyrillic characters not included in the ASCII character set and leave words containing only ASCII characters in the output.

2. The output was de-tokenized using a set of heuristics implemented as regular expressions designed to undo the process of English tokenization. Punctuation was attached to neighboring words and tokens that form split contractions were combined into a single token.

3. The output was re-cased using the re-casing tool supplied with the MOSES [3] toolkit. We trained the re-casing tool on untokenized text from the TED talk training data.

2.4. Training

2.4.1. Data

We trained out translation and language models using only the in-domain TED data supplied for the task. This data consisted of approximately 120k bilingual sentence pairs containing about 2.4 million words of English, and 2 million words of Russian. In addition to this data, we used approximately 600,000 bilingual article title pairs extracted from the interlanguage links of the most recent dump of the Russian Wikipedia database. In the remainder of this section we describe the details of the process of building the machine translation engine used in our experiments. A description of the training and application of the transliteration mining component of our system follows in the next section.

2.4.2. Language Model

The language models were built using the SRI language modeling toolkit [5]. A 5-gram model was built for decoding the development and test data for evaluation, and a 3-gram model was built on the same data for efficient tuning. Pilot experiments indicated that using a lower order language model for tuning did not significantly affect the translation quality of the systems produced by the MERT process. The language models were smoothed using modified Knesser-Ney smoothing.

2.4.3. Translation Model

The translation model for the base system was built in the standard manner using a 2-step process. First the training data was word-aligned using GIZA++. Second, the grow-diag-final-and phrase-extraction heuristics from the MOSES [3, 6] machine translation toolkit were used to extract a set of bilingual phrase-pairs using the alignment produced by GIZA++. However before training the proposed system, mined single-word transliteration/cognate pairs were added to the training data set. In doing this, these word pairs are guaranteed to align, increasing their alignment counts thereby encouraging their alignment where they occur together in the remainder of the corpus. Pilot experiments were run on development data to assess the effect of adding these transliteration/cognate pairs multiple times to the data. We found that adding the pairs a single time was the most effective strategy.

2.4.4. Parameter Tuning

To tune the values for the log-linear weights in our system, we used the standard minimum error-rate training procedure

(MERT) [7]. The weights for the models were tuned using the development data supplied for the task.

3. Using Sub-word Alignment

3.1. Motivation

The use of transliterations to aid the alignment process was first proposed by [8], and has been shown to improve word alignment quality in [9]. The idea is based on the simple principle that for transliterations and cognates there exist similarities at the substring level due to the relationships these words possess, these relationships can be discovered by bilingual alignment at the grapheme level, and may be used as additional alignment evidence during a word alignment process. However this promising idea has received little attention in the literature. Our system is based on a two step process: first a bilingual alignment model is built from noisy data using a transliteration mining process; in the second step the training corpus itself is mined for transliterations/cognates using the model built from the first step. We describe these two steps in more detail in the next two subsections.

3.2. Transliteration Mining

3.2.1. Corpus

To train the mining system we extracted 629,021 bilingual Russian-English interlanguage link title pairs from the most recent (July 2012) Wikipedia database dump. From this data we selected only the single word pairs for training, leaving a corpus of 145,817 noisy word pairs. We expected (based on our experience building transliteration generation models on these languages) that the amount of clean data in this corpus would be sufficient for training the transliteration component of our generative model since the grapheme vocabulary sizes for both languages are not large, and the alignments are often reasonably direct (as can be seen in the set of examples given below). 98,902 pairs were automatically extracted from this corpus as transliteration/cognate pairs.

3.2.2. Methodology

The mining model we used was based on the research of [10] which in turn draws on the work of [11] and [12].

The mining system is capable of simultaneously modeling and clustering the data. It does this by means of a single generative model that is composed of two sub-models: the first models the transliterations/cognates; the second models the noise. The generative story for this model is as follows:

1. Choose whether to generate noise (with probability λ), or a transliteration/cognate pair (probability $1 - \lambda$);
2. Generate the noise pair, or the transliteration pair with the respective sub-model.

The noise and transliteration/cognate sub-models are both unigram joint source-channel models [13]: the joint probability of generating a bilingual word pair is given by the product of the probabilities of a sequence steps each involving the generation of a bilingual grapheme sequence pair. The difference between these models being the types of grapheme sequence pair they are allowed to generate.

As in [10], we have extended the nonparametric Bayesian alignment model of [12] to include null alignments to either single characters or sequences of graphemes up to a maximum specified length. The alignment model is symmetrical with respect to the source and target languages and therefore these null alignments can be to either source or target grapheme sequences, and their probabilities are learned during training in the same manner as the other parameters in the model.

The difference between the noise and transliteration/cognate sub-models was that the noise sub-model was restricted to generate using only null alignments. In other words, the noise sub-model generates the source and target sequences independently. Constraining the noise model in this way allows it to distribute more of its probability mass onto those model parameters that are useful for explaining data where there is no relationship between source and target. The transliteration/cognate sub-model on the other hand is able to learn the many-to-many grapheme substitution operations useful in modeling pairs that can be generated by bilingual grapheme sequence substitution. During the sampling process, both models compete to explain the word pairs in the corpus, thereby naturally clustering them into two sets while learning.

Our Bayesian alignment model is able to perform many-to-many alignment without the overfitting problems commonly encountered when using maximum likelihood training. In the experiments reported here, we arbitrarily limit the maximum source and target sequence lengths to 3 graphemes on each side. This was done to speed up the training process, but was not strictly necessary.

The aligner was trained using block Gibbs sampling using the efficient forward-filter backward-sample dynamic programming approach set out in [14]. The initial alignments were chosen randomly using an initial backward sampling pass with a uniform distribution on the arcs in the alignment graph. The prior probability of the pairs being noise (λ) was set to 0.5 in the first iteration. During the training λ was updated whenever the class (transliteration/cognate or noise) of a bilingual word pair was changed in the sampling process. λ was calculated based on a simple frequency count of the classes assigned to all the word pairs while sampling.

3.3. Mining the Training Set

In order to discover alignments of transliteration/cognate pairs in the training data we again applied a mining approach. We aligned each Russian word to each English word in the same sentence of the training corpus, and then used the approach of [15] to determine whether these pairs were transliterations/cognates. In principle it would be possible to apply

the approach described in the previous section here, however, we chose not to attempt this due to the considerably larger amount of noise in this data, and also because of the size of this corpus. For full details of this method the reader is referred to [15], but in brief the technique mines data by first aligning it using an alignment model similar to the transliteration sub-model described in the previous section. Then features extracted from the alignment are combined with features derived from the characteristics of the word pairs (for example their relative lengths); these features are then used to classify the data. The advantages of this approach over the method described in the previous section are firstly that it utilizes a model already trained on relatively clean data, and so will not be affected by the noise in the corpus being mined, and secondly no iterative learning is required; the process is effectively the same as the backward sampling step and can proceed very rapidly given an already trained model. The mining process yielded a sequence of word pairs that the system considered to be likely candidates for transliterations/cognates. This sequence of pairs was added to the training data used to build the translation model, in doing so these word pairs were forced to align to each other and the counts for their alignments were increased thereby encouraging their alignments in the remainder of the corpus. We ran pilot experiments to determine the effect of increasing the counts further by adding the mined pairs multiple times to the corpus, and although the performance seemed reasonably insensitive to the number of copies of the data we used, the experiments with a single copy of the data gave the highest scores. In future research we would seek to either soften this parameter and then optimize it on the data set (in a similar manner to [11]), or ideally remove it altogether by integrating the mining and alignment processes.

3.4. Examples

Some typical examples of mined transliteration/cognate pairs are given in Table 3.4. Notice that in many of the examples (for example Соционика/Socionics) most of the mapping is possible with simple grapheme-to-grapheme substitutions. In this example, a transformation of the word ending (ика↦ics) is also required. This transformation is quite common in the corpus and the aligner learned this as a model parameter. Furthermore, the grapheme sequence pair was used as a single step in aligning both this word pair and others with analogous endings in the corpus. The mining process was able to learn to be robust to small variations in the data. For example in the pair Посткапитализм/Post-capitalism a hyphen is present on the English side, but not on the Russian side. The aligner learned to delete hyphens in the data by aligning them to null, thereby learning to model its asymmetrical usage in the data.

Russian	**Engish**
Космополитизм (KOSMOPOLITIZM)	Cosmopolitanism
Посткапитализм (POSTKAPITALIZM)	Post-capitalism
Соционика (SOCIONIKA)	Socionics
Физика (FIZIKA)	Physics
Механика MEHANIKA	Mechanics
Парапсихология (PARAPSIHOLOGIJA)	Parapsychology
Хронология (HRONOLOGIJA)	Chronology
Спагетти (SPAGETTI)	Spaghetti
Париж (PARIZH)	Paris

Table 1: Examples of transliteration/cognate pairs discovered by mining Wikipedia interlanguage link titles.

3.5. Experiments

We evaluated the effectiveness of our approach using the the supplied training, development and IWSLT2010 and IWSLT2011 test data sets. The baseline model was trained identically, but without using the mined data. The results are shown in Figure 3.5. Our results show a modest but consistent improvement in translation performance on both test sets, motivating further development of this approach. We analyzed the results to investigate the impact of the approach on the number of OOVs in the test data. Surprisingly on both IWSLT2010 and IWSLT2011 test sets our approach gave rise to a 0.2% increase in number of OOVs. This may indicate our approach is succeeding by improving the overall word alignment, rather than by improving the translation of words with cognates and transliterations in the target language.

Model	**IWSLT2010**	**IWSLT2011**
Baseline	16.23	18.08
Proposed	16.77	18.53

Table 2: The effect on BLEU score of using sub-word alignments to assist word alignment.

4. Conclusions

This paper described NICT's system for the IWSLT 2012 evaluation campaign for the TED speech translation Russian-English shared-task. Our approach was based on a fairly typical phrase-based statistical machine translation system that was augmented using a transliteration mining approach designed to exploit the alignments between transliterations and

cognates to improve the word alignment. Our experimental results on the IWSLT2010 and IWSLT2011 test sets gave improvements of approximately 0.5 BLEU percentage points.

In future work we would like to explore integrate the transliteration/cognate mining techniques more tightly into the word alignment process. We believe it should be possible to simultaneously word align while mining the corpus for sub-word alignments, within a single nonparametric Bayesian alignment process.

5. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. of the International Workshop on Spoken Language Translation*, Hong Kong, HK, December 2012.

[2] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[3] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowa, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *ACL 2007: proceedings of demo and poster sessions*, Prague, Czeck Republic, June 2007, pp. 177–180.

[4] C.-L. Goh, T. Watanabe, M. Paul, A. Finch, and E. Sumita, "The NICT Translation System for IWSLT 2010," in *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, M. Federico, I. Lane, M. Paul, and F. Yvon, Eds., 2010, pp. 139–146.

[5] A. Stolcke, "Srilm - an extensible language model toolkit," 1999. [Online]. Available: http://www.speech.sri.com/projects/srilm

[6] P. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," in *Machine translation: from real users to research: 6th conference of AMTA*, Washington, DC, 2004, pp. 115–124.

[7] F. J. Och, "Minimum error rate training for statistical machine translation," in *Proceedings of the ACL*, 2003.

[8] U. Hermjakob, "Improved word alignment with statistics and linguistic heuristics," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, August 2009, pp. 229–237. [Online]. Available: http://www.aclweb.org/anthology/D/D09/D09-1024

[9] H. Sajjad, A. Fraser, and H. Schmid, "An algorithm for unsupervised transliteration mining with an application to word alignment," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 430–439. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002527

[10] O. Htun, A. Finch, E. Sumita, and Y. Mikami, "Improving transliteration mining by integrating expert knowledge with statistical approaches," *International Journal of Computer Applications*, vol. 58, November 2012.

[11] H. Sajjad, A. Fraser, and H. Schmid, "A statistical model for unsupervised and semi-supervised transliteration mining," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 469–477. [Online]. Available: http://www.aclweb.org/anthology/P12-1049

[12] A. Finch and E. Sumita, "A Bayesian Model of Bilingual Segmentation for Transliteration," in *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, M. Federico, I. Lane, M. Paul, and F. Yvon, Eds., 2010, pp. 259–266.

[13] H. Li, M. Zhang, and J. Su, "A joint source-channel model for machine transliteration," in *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2004, p. 159.

[14] D. Mochihashi, T. Yamada, and N. Ueda, "Bayesian unsupervised word segmentation with nested pitman-yor language modeling," in *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*. Morristown, NJ, USA: Association for Computational Linguistics, 2009, pp. 100–108.

[15] T. Fukunishi, A. Finch, S. Yamamoto, and E. Sumita, "Using features from a bilingual alignment model in transliteration mining," in *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*, 2011, pp. 49–57.

TED Polish-to-English translation system for the IWSLT 2012

Krzysztof Marasek

Multimedia Department
Polish-Japanese Institute of Information Technology, Warsaw, Poland
kmarasek@pjwstk.edu.pl

Abstract

This paper presents efforts in preparation of the Polish-to-English SMT system for the TED lectures domain that is to be evaluated during the IWSLT 2012 Conference. Our attempts cover systems which use stems and morphological information on Polish words (using two different tools) and stems and POS.

1. Introduction

Polish, one of the West-Slavic languages [1], due to its complex inflection and free word order, forms a challenge for statistical machine translation (SMT). Polish grammar is quite complex: seven cases, three genders, animate and inanimate nouns, adjectives agreed with nouns in terms of gender, case and number and a lot of words borrowed from other languages which are often inflected similarly to those of Polish origin. These cause problems in establishing vocabularies of manageable sizes for translation to/from other languages and sparseness of data for statistical model training. Despite of ca. 60 millions of Polish speakers worldwide the number of publicly available resources for the preparation of SMT systems is rather limited, thus progress in that domain is slower than for other languages. In this paper, our efforts in preparation of the Polish-to-English SMT system for the TED task, part of the IWSLT 2012 evaluation campaign, MT optional track, are described.

The remainder of the paper is structured as follows. In section 2 Polish data preparation is described, section 3 deals with English, 4 with training of the translation and language models, and section 5 presents our results. Finally, the paper concludes with a discussion about encountered issues and future perspectives in sections 6 and 7.

2. Polish data preparation

Training, development and evaluation data consists of the Polish translation of TED lectures and its English origin. This has been prepared by FBK [2]. The available data set consists of ca. 2.27 millions of untokenized words on the target side. The transcripts are given as pure text (UTF-8 encoding), one or more sentences per line, and are aligned at language pair level. The organizers also provide a lot of monolingual data (English) and the PL-EN Europarl v.7 parallel corpus.

Some manual preprocessing of training data was necessary.

After extracting the transcripts from the supplied XML files the same number of lines for both languages were obtained, but with some discrepancies in the parallel text. Those differences were caused mostly by repetitions in the Polish text and some additional remarks (like “Applause” or “Thanks”) which were not present in the English text. 28 lines had to be manually corrected for the whole set of 134325 lines. Without trying to judge the TED data translation quality, but as a Polish native speaker, it left an impression that, at least part of the talks were translated by volunteers, making the training material a bit noisy. Moreover, a lot of English proper names are inserted into Polish text.

The vocabulary sizes (extracted using SRILM [3]) were 198622 for Polish and 91479 for English, which exposes the fundamental problem for the translation – the huge difference in the vocabulary sizes.

Tokenization of input data was done using standard tools delivered with Moses [4], with an extension created by FBK for Polish.

Before a translation model was trained, the usual preprocessing was applied, such as removing long sentences (threshold 60) and sentences with length difference exceeding a certain threshold. This was done again using scripts from the Moses toolkit.

The final tokenized, lowercased and cleaned training corpus for Polish and English was 132307 lines long, but with an even greater difference in vocabulary sizes – 47250 for English vs. 123853 for Polish.

This large difference between source and target vocabulary sizes shows the necessity of using additional knowledge sources. Initially, we decided to limit the size of the Polish vocabulary by using stems instead of surface forms. Following that, we tried using morphosyntactic tagging as an additional source of information for the SMT system.

2.1. Stems extraction for Polish

Inspired by the works of Bojar [6], we tried to use stems of Polish words instead of its surface forms with the purpose of reducing the vocabulary size difference. Since the target language is English, it was not necessary to build models which will convert stems to correct grammatical forms – the target was a normal English sentence (surface forms).

For that purpose, a set of freely available tools prepared by the NLP group of the Wrocław Technical University was used. This set of NLP-tools (http://nlp.pwr.wroc.pl) can be used to perform the following tasks:

- Tokenisation — division into tokens and sentences
- Morphosyntactic analysis using the available analysers and dictionaries (including Morfeusz SGJP/SIAT), but also user-supplied dictionaries
- Morphosyntactic tagging
- Shallow parsing (understood as chunking)
- Turning running text into a sequence of feature vectors (using WCCL formalism, useful for further NLP tasks)

From this, two main components were used:

- MACA [8] – a universal framework to join different sources of morphological information, including the existing resources as well as user-provided dictionaries. This framework allows writing simple

configuration files that define tokenisation strategies and the behavior of morphological analysers, including simple tagset conversion.

- WCRFT [7] – morphosyntactic tagger which brings together Conditional Random Fields and tiered tagging (where grammatical information is split into several tiers, usually one tier is used for each of grammatical classes).

The tools, when used in a sequence, form XML-formatted output containing for each token: its surface form, stem and morphosyntactic tag (tags).
If stems are only taken from the Polish TED training data, the vocabulary (for data cleaned as previously) is substantially reduced to only 44102 words.

2.2. Morphosynactic tagging: Wrocław tools

The tagset used by the Wrocław's analyzers could have been changed, but it was most straightforward to use the standard settings, where the IPIC (IPI PAN Corpus, Polish National Corpus [9]) tagset is used. This particular tagset allows for much more fine-grained tagging compared to traditional parts-of-speech. Each tag contains a grammatical class and zero or more values for certain attributes. Each grammatical class defines a set of attributes whose values must be specified. For instance, nouns require that number, gender and case attributes are specified, and adverbs require the degree attribute. This in turn causes specific segmentation of input text, where some words are split into several tokens, thus tokenization differs from the one delivered by standard Moses tools. This causes some problems when building parallel corpora. In order to avoid these problems, additional markers were placed at the end of each input line.
The tagger tries to disambiguate the grammatical forms giving the set of most probable tags. Usually, just one tag is provided and only in really undistinguishable cases all possible tags are given, as in the following example (pl.gen. *man* from sin.nom. *man* or pl.nom *people*):

```
<tok>
<orth>ludzi</orth>
<lex      disamb="1">      <base>człowiek</base>
<ctag>subst:pl:gen:m1</ctag></lex>
<lex       disamb="1">       <base>ludzie</base>
<ctag>subst:pl:gen:m1</ctag></lex>
</tok>
```

In such a case only the first form (first stem) was taken for further processing.

2.3. Morphosynactic tagging: our tools

In several projects related to speech technology a grave demand for text normalization is observed. Text normalization is the process of converting any abbreviations, numbers and special symbols into corresponding word sequences. In particular, normalization is responsible for:
1. expansion of abbreviations in the text into their full form;
2. expansion of any numbers (e.g. Arabic, Roman, fractions) into their appropriate spoken form;
3. expansion of various forms of dates, hours, enumerations and articles in contracts and legal documents into their proper word sequences.

This task, although seemingly simple, is in fact quite complicated – especially in languages like Polish which has 7 cases and 15 gender forms for nouns and adjectives, with additional dimensions for other word classes. That is why most abbreviations have multiple possible expansions and each number notation over a dozen outcomes.
To solve this task we prepared tools [10] which we also try to use for morphosyntactic tagging of Polish texts.
The system consists of a decoder, a language model and a set of expansion rules. The expansion rules are used in the expansion of commonly used abbreviations and written date and number forms. A synchronous Viterbi style decoder that generates a list of hypotheses ordered by the values retrieved from the language model is used. Each time the text contains a word sequence that could be expanded; all the possible expansions are fed into the decoder. Because the expansion of long numbers or some abbreviations expects that several words need to be added at once, hypotheses of varying lengths may end up competing against each other. This is remedied by the normalization of hypotheses' probabilities to their lengths. Such normalization is equivalent to the addition of a heuristic component commonly used in asynchronous decoders like A∗. The language model itself is a combination of three models with a range of n=3 for the individual words, n=5 for word stems and n=7 for grammatical classes. The Evolution Strategy (μ + λ) is used for optimization of model weights, especially:
1. weights of 30 text domain sets (10 parameters for each model),
2. linear interpolation weight for all n-grams in all models. The weights depended on the frequency of occurrence of given n-gram - there were 5 ranges of frequency,
3. linear interpolation weights for the word, stems and grammar classes models (combining the smaller models into one larger), with perplexity of the final model on development set as a quality criterion.

The outcome of the system is also a morphosyntactic tagging of tokens, however no disambiguation is done. Instead, a numerical value describing all possible tags for a given form is stored, eg.:

```
id = 15
features:
adj;acc;sg;m_os;;pos;;
adj;acc;sg;m_zyw;;pos;;
adj;gen;sg;m_nie_zyw;;pos;;
adj;gen;sg;m_os;;pos;;
adj;gen;sg;m_zyw;;pos;;
adj;gen;sg;neu;;pos;;
```

for the surface form "tego" (stem: "ten", eng. *this*).
It should be also noted, that stems are generated only for words from a given vocabulary (for other words OOV symbol is placed) and proper names, foreign words, spellings and abbreviations are recognized and special symbols are inserted instead of stems as in following example:

```
plan|plan|5 był|być|106 w|*letter|0
pełni|pełnia|9 gotowy|gotowy|18 w|*letter|0
dziewięćdziesiątym|dziewięćdziesiąty|255
ósmym|ósmy|255 roku|rok|93  nosił|nosić|106
nazwę|nazwa|10 digital|oov|-2 Millennium|OOV|-
2 Copyright|OOV|-2 act|OOV|-2 .|.|
```

Our tool uses Windows-1250 Eastern Europe character encoding, thus it was necessary to convert data from/to UTF-8 encoding used by all other tools. The decoding procedure showed several UTF-8 special characters used in the original text (like musical notes, etc.) which added some manual work to remove those unnecessary symbols.

3. English data preparation

Preparation of English data was less complicated. For the baseline (surface form) and stems of Polish, only surface forms of English TED data was used. For the factored model, English text was tagged using Stanford CoreNLP tools [11,12]. Stanford CoreNLP integrates all necessary NLP tools, including the parts-of-speech (POS) tagger and provides model files for analysis of English, providing the base forms of words, their parts of speech, recognition of named entities, normalization of dates, times, and numeric quantities, and marks of the structure of sentences in terms of phrases.

4. Training and tuning procedure

Only in-domain data for training of the SMT system was used, mainly because of our lack of experience in translation model adaptation. Also, no other English data for language modeling was used. The supplied Euro-parlament data was from a too distant domain and our attempts to use Google n-grams ended without success (noisy data, tools which we have did not work properly on such huge large data sets). TED talks corpus consists of data which varies significantly with respect to the topics or domain, but has a rather homogeneous presentation style. Moreover, the TED training data perfectly matches the test condition, so we assume that the possible gain from using other data could be limited. It was also our intention to focus our work on researching proper factors combination and configuration of the SMT training.

Thus, TED lectures data [2] was used for training in 4 main modes:

BASE Polish surface form to English surface form

STEM Polish stems to English surface form

FCT1 Polish factors (surface form | stem | extended morphosytactic tag from Wrocław tools) to English factors (surface form | stem | POS from Stanford CoreNLP),

FCT2 Polish factors (surface form | stem | numerical morphosytactic tag from our tool) to English factors (surface form | stem | POS from Stanford CoreNLP).

As development and evaluation data again TED talks are used [2]. The set "iwslt2012-dev2010" consists of 767 lines. Testing of the system was done on "iwslt2012-tst2010" set build of 1564 lines. All development and test data has been prepared for all 4 modes of the SMT training.

All the language models used are 5-gram interpolated language models with Kneser-Ney discounting and were trained with the SRILM toolkit [12]. This includes also language models trained on stems and grammatical tags.

The word alignment of the parallel corpora was generated using the GIZA++-Toolkit [5]. Afterwards, the alignments were combined using the grow-diag-final-and heuristic. The phrases were extracted and scored using the Moses toolkit [4]. For the BASE, FCT1 and FCT2 systems several reordering models were tested. Only marginal improvement on test data was achieved compared to the standard setting "msd-bidirectional-fe".

Tuning was done using MERT Moses' implementation [14] on development data. New weights were then used for testing. A lot of work was spent on finding good composition of factors for translation, generation and decoding steps of the factored models. However, as shown in the next section, we did not find efficient factors yet.

5. Evaluation

For training all the data has been lowercased and tokenized. The evaluation needs data to be recased to its original form. For that, a model was trained using standard Moses tool train-recaser.pl. Evaluation results are presented in Tables 1 and 2.

Table 1: Results of the evaluation, truecase and punctation

TASK	SYSTEM	BLEU	METEOR	WER	PER	TER	GTM	NIST
	BASE	0.2	0.56	0.66	0.52	61.42	0.55	5.64
dev2010	STEM	0.19	0.56	0.66	0.54	62.41	0.53	5.43
	FCT1	0.13	0.47	0.64	0.57	61.88	0.5	4.23
	FCT2	0.1						2.96
	BASE	0.15	0.49	0.74	0.59	69.04	0.49	4.9
tst2010	STEM	0.14	0.49	0.73	0.6	69.21	0.48	4.77
	FCT1	0.11	0.43	0.69	0.6	66.15	0.46	3.92
	FCT2	0.09						2.71
	BASE	0.19	0.54	0.68	0.55	64.19	0.53	5.44
tst2011	STEM	0.17	0.54	0.69	0.57	65.07	0.51	5.2
	FCT1	0.14	0.47	0.64	0.57	61.84	0.49	4.39
	FCT2							
	BASE	0.15	0.48	0.72	0.6	67.96	0.48	4.98
tst2012	STEM	0.14	0.48	0.72	0.6	68.31	0.47	4.78
	FCT1	0.11	0.42	0.69	0.62	66.14	0.45	3.6

Table 2: Results of the evaluation, no casing and no punctation

TASK	SYSTEM	BLEU	METEOR	WER	PER	TER	GTM	NIST
	BASE	0.19	0.53	0.67	0.54	64.46	0.53	5.78
dev2010	STEM	0.17	0.53	0.68	0.56	65.82	0.51	5.5
	FCT1	0.13	0.45	0.66	0.58	64.97	0.48	4.33
	FCT2	0.1						2.88
	BASE	0.14	0.46	0.76	0.62	73.12	0.47	5.05
tst2010	STEM	0.13	0.46	0.76	0.63	73.66	0.45	4.86
	FCT1	0.11	0.41	0.72	0.62	70.05	0.44	4.09
	FCT2	0.08						2.67
	BASE	0.18	0.5	0.7	0.57	67.44	0.51	5.64
tst2011	STEM	0.16	0.5	0.71	0.59	69.19	0.49	5.33
	FCT1	0.13	0.44	0.67	0.59	65.64	0.47	4.48
	FCT2							
	BASE	0.14	0.44	0.74	0.61	71.53	0.46	5.13
tst2012	STEM	0.13	0.44	0.74	0.63	72.52	0.44	4.85
	FCT1	0.1	0.39	0.72	0.64	70.51	0.43	3.61

TASK describes the test set, SYSTEM is one of the systems described in section 4, and BLEU, METEOR, WER, PER, TER, GTM and NIST are appropriate evaluation scores (see en.wikipedia.org/wiki/Evaluation_of_machine_translation for explanation). For the BASE, STEM, FCT1systems the scoring was done by the IWSLT evaluation team [17], for the system FCT2 scoring was done in house using mteval-v12 NIST script for dev2010 and tst2010 datasets only.

6. Discussion

As mentioned in section 4, a lot of work was spent trying to find the best combination of factors for translation, generation and decoding steps within the Moses framework. Unfortunately, a lot of combination ended with decoder errors, with no clear reasons given. This showed that more experience to use those advanced features is definitely needed.

Many researchers claim that word alignment is crucial for good SMT results. The recent study of Wróblewska [15] shows that, in her experiments, best precision of word alignment was achieved if the Polish side of the parallel corpus was lemmatized. This reduces the number of items in the lemma dictionary and approximates the English token dictionary. She does not give an answer to whether lemmatising the English part of the parallel corpus is necessary. Her results somewhat resemble the work presented in this paper.

It also clear that TED talks is a difficult task, at least on the Polish side (huge vocabulary, many long lines). Just for comparison, on the BTEC corpus [16] we obtained better results (NIST=14.27 BLEU=0.89 on development set using mteval-v12 script). It is because BTEC consists of short, clear sentences without any foreign terms (usually inflected in Polish) as it is in the TED talks.

7. Conclusions

The conducted experiments are only a first step towards building the final Polish-to-English SMT system. We tried to use surface forms, stems and two kinds of factors describing grammatical properties of Polish words and surface forms, stems and POS for English. In the near future, we will try to use more data (Europarl) for the SMT preparation and optimize the system for the in-domain data. In further research, we would like to investigate the usage of surface forms and stems simultaneously on the Polish side and look more deeply into works done for other Slavic languages.

8. Acknowledgements

This work is sponsored by the EU-Bridge 7 FR project (grant agreement no. 287658) and statutory works of the PJIIT (ST/MUL/4/2011).

9. References

[1] Jagodziński G., "A Grammar of Polish Language", http://grzegorj.w.interia.pl/gram/en/gram00.html

[2] Cettolo M, Girardi C., Federico M., "WIT3: Web Inventory of Transcribed and Translated Talks". In *Proc. of EAMT*, pp. 261-268, Trento, Italy, 2012

[3] Stolcke A., "SRILM - An Extensible Language Modeling Toolkit", in *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, September 2002

[4] Koehn P., Hoang H., Birch A., Callison-Burch C., Federico M., Bertoldi N., Cowan B., Shen W., Moran C., Zens R., Dyer R., Bojar O., Constantin A., and Herbst E., "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of ACL 2007*, Demonstration Session, Prague, Czech Republic, 2007.

[5] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[6] Bojar O., "Rich Morphology and What Can We Expect from Hybrid Approaches to MT". *Invited talk at International Workshop on Using Linguistic Information for Hybrid Machine Translation* (LIHMT-2011), http://ufal.mff.cuni.cz/~bojar/publications/2011-FILE-bojar_lihmt_2011_pres-PRESENTED.pdf , 2011

[7] Radziszewski A., "A tiered CRF tagger for Polish", in: *Intelligent Tools for Building a Scientific Information Platform: Advanced Architectures and Solutions*, editors: Membenik R., Skonieczny L., Rybiński H., Kryszkiewicz M., Niezgódka M., Springer Verlag, 2013 (to appear)

[8] Radziszewski A., Śniatowski T., "Maca: a configurable tool to integrate Polish morphological data", *Proceedings of the Second International Workshop on Free/Open-Source Rule-Based Machine Translation*, FreeRBMT11, Barcelona, 2011

[9] Przepiórkowski A., Bałko M., Górski R., Lewandowska-Tomaszczyk B., „*Narodowy Korpus Języka Polskiego*", PWN Warszawa, 2012

[10] Brocki Ł., Marasek K., Korzinek D., "Multiple Model Text Normalization for the Polish Language", *The 20th International Symposium on Methodologies for Intelligent Systems ISMIS-2012*, Macau, 4-7 December 2012 (in press)

[11] Toutanova K, Klein D., Manning Ch., and Singer Y., "Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network", in *Proceedings of HLT-NAACL* 2003, pp. 252-259.

[12] Finkel J., Grenager T., and Manning Ch., Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics* (ACL 2005), pp. 363-370.

[13] Stolcke A., "SRILM – An Extensible Language Modeling Toolkit", *International Conference on Spoken Language Processing*, Denver, Colorado, USA, 2002.

[14] Bertoldi N., Haddow B., Fouet J.-B., "Improved Minimum Error Rate Training in Moses", *The Prague Bulletin of Mathematical Linguistics*, February 2009, pp.1-11

[15] Wróblewska A., "Polish-English word alignment: preliminary study", in Ryżko D., Rybiński H., Gawrysiak M., Kryszkiewicz M, editors, *Emerging Intelligent Technologies in Industry, volume 369 of Studies in Computational Intelligence*, pp. 123–132, Springer-Verlag, Berlin, 2011.

[16] Takezawa T., Kikui G., Mizushima M., Sumita E., "Multilingual Spoken Language Corpus Development for Communication Research", *Computational Linguistics and Chinese Language Processing*, Vol. 12, No. 3, September 2007, pp. 303-324

[17] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, S. Stueker, Overview of the IWSLT 2012 Evaluation Campaign, *In Proc. of IWSLT*, Hong Kong, HK, 2012

Forest-to-String Translation using Binarized Dependency Forest for IWSLT 2012 OLYMPICS Task

Hwidong Na and Jong-Hyeok Lee

Department of Computer Science and Engineering
Pohang University of Science and Technology (POSTECH), Republic of Korea
{ leona, jhlee } @postech.ac.kr

Abstract

We participated in the OLYMPICS task in IWSLT 2012 and submitted two formal runs using a forest-to-string translation system. Our primary run achieved better translation quality than our contrastive run, but worse than a phrase-based and a hierarchical system using Moses.

1. Introduction

Syntax-based SMT approaches incorporate tree structures of sentences to the translation rules in the source language [10, 14, 23, 22], the target language [1, 7, 12, 18, 26], or both [2, 3, 28]. Due to the structural constraint, the transducer grammar extracted from parallel corpora tends to be quite large and flat. Hence, the extracted grammar consists of translation rules that appear few times, and it is difficult to apply most translation rules in the decoding stage.

For generalization of transducer grammar, binarization methods of a phrase structure grammar have been suggested [1, 12, 20, 26]. Binarization is a process that transforms an n-ary grammar into a binary grammar. During the transformation, a binarization method introduces the virtual nodes which is not included in the original tree. The virtual nodes in a binarized phrase structure grammar are annotated using the phrasal categories in the original tree. Unfortunately, these approaches are available only for string-to-tree models, because we are not aware of the correct binarization of the source tree at the decoding stage. To take the advantage of binarization in tree-to-string models, a binarized forest of phrase structure trees has been proposed [25]. Since the number of all possible binarized trees are exponentially many, the author encode the binarized trees in a packed forest, which was originally proposed to encode the multiple parse trees [14].

In contrast to previous studies, we propose to use a novel binarized forest of dependency trees for syntax-based SMT. A dependency tree represents the grammatical relations between words as shown in Figure 1. Dependency grammar has that holds the best phrasal cohesion across the languages [6]. We utilize dependency labels for the annotation of the virtual nodes in a binarized dependency tree. To the best of our knowledge, this is the first attempt to binarize the dependency grammar.

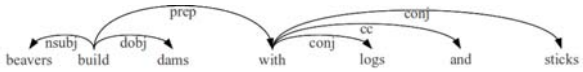


Figure 1: An example dependency tree with dependency labels

2. Binarized Dependency Forest

Forest-to-string translation approaches construct a packed forest for a source sentence, and find the mapping between the source forest and the target sentence. A packed forest is a compact representation of exponentially many trees. Most studies focused on the forest of multiple parse trees in order to reduce the side effect of the parsing error [13, 14, 15, 19, 27, 28]. On the other hand, Zhang et. al. [25] attempted to binarize the best phrase structure tree. A binarization method comprises the conversion of the possibly non-binary tree into a binarized tree. The authors suggested a binarized forest, which is a packed forest that compactly encodes multiple binarized trees. It improves generalization by breaking downs the rules into the smallest possible parts. Thus, a binarized forest that the authors suggested covers non-constituent phrases by introducing a virtual node, for example, “beavers build” or “dams with” in Figure 1.

In this paper, we propose a binarized forest analogous to but two differences. First, we binarize the best *dependency* tree instead of the best phrase structure tree. Because dependency grammar does not have non-terminal symbols, it is not trivial to construct a binarized forest from a dependency tree. Second, we annotate the virtual nodes using the dependency labels instead of the phrase categories.

2.1. Construction of binarized dependency forest

We utilize the concept of the well-formed dependency proposed by Shen et. al. [18]. A well-formed dependency refers to either a connected sub-graph in a dependency tree (treelet) or a floating dependency, i.e., a sequence of treelets that have a common head word. For example, “beavers build” is a treelet and “dams with” is a floating dependency.

Since the number of all possible binarized trees are exponentially many, we encode a binarized forest $\mathcal{F}$ in a chart analogous to Zhange et. al. [25]. Let π be the best dependency tree of a source sentence from w_1 to w_n. π consists of a set of information for each word w_j, i.e. the head word $HEAD(w_j)$ and the dependency label $LABEL(w_j)$. For each word w_j, we initialize the chart with a binary node v. For each span $s^{begin:end}$ that ranges from $w_{begin+1}$ to w_{end}, we check whether the span consists of a well-formed dependency. For each pair of sub-spans $s^{begin:mid}$ and $s^{mid:end}$, which are rooted at v_l and v_r respectively, we add an incoming binary edge e if:

- Sibling (SBL): v_l and v_r consist of a floating dependency, or
- Left dominates right (LDR): v_l has no right child $RIGHT(v_l)$ and v_l dominates v_r, or
- Right dominates left (RDL): v_r has no left child $LEFT(v_r)$ and v_r dominates v_l.

Note that the root node of the SBL case is a virtual node, and we extend the incoming binary edge of v for LDR and RDL cases by attaching v_r and v_l, respectively. For example, $\{dobj, prep\}^{2:4}$ is the root node for the SBL case where v_l is "dams" and v_r is "with", and build$^{0:4}$ is the root node for the LDR case where v_l is build$^{0:2}$ and v_r is $\{dobj, prep\}^{2:4}$.

Algorithm 1 shows the pseudo code, and Figure 2 shows a part of the binarized forest for the example dependency tree in Figure 1. Although the worst time complexity of the construction is $O(n^3)$, the running time is negligible when we extract translation rules and decode the source sentence in practice (less than 1 ms). Because we restrict the combination, a binary node has a constant number of incoming binary edges. Thus, the space complexity is $O(n^2)$.

Algorithm 1: Construct Binarized Dependency Forest

```
1  function Construct(π)
   input  : A dependency tree π for the sentence
            w_1 ... w_J
   output: A binarized forest F stored in chart
2  for col = 1 ... J do
3      create a binary node v for w_col
4      chart[1, col] ← v
5  end
6  for row = 2 ... J do
7      for col = row ... J do
8          if a span s^{col−row:col} consists of a
           well-formed dependency then
9              create a binary node v
10             for i = 1 ... row do
11                 v_l ← chart[i, col − row + i]
12                 v_r ← chart[row − i − 1, col]
13                 if v_l and v_r consist of a
                   floating dependency then
14                     create an incoming binary node
                       e = ⟨v, v_l, v_r⟩
15                 end
16                 else if v_l has no right child
                   and v_l dominates v_r then
17                     create an incoming binary node
                       e = ⟨v_l, LEFT(v_l), v_r⟩
18                 end
19                 else if v_r has no left child
                   and v_r dominates v_l then
20                     create an incoming binary node
                       e = ⟨v_r, v_l, RIGHT(v_r)⟩
21                 end
22                 else
23                     continue // combination is
                           not allowed
24                 end
25                 IN(v) ← IN(v) ∪ {e}
26             end
27             chart[row, col] ← v
28         end
29     end
30 end
```

2.2. Augmentation of phrasal node

We also augment phrasal nodes for word sequences, i.e. phrases in PBSMT. A phrasal node p is a virtual node corresponding to a span $s^{begin:end}$, yet it does not consist of a well-formed dependency. Hence, augmenting phrasal nodes in $\mathcal{F}$ leads to including all word sequences covered in PBSMT. Because phrases capture more specific translation patterns, which are not linguistically justified, we expect that the coverage of the translation rules will increase as we augment phrasal nodes.

We augment phrasal nodes into the chart that we built for the binarized forest. For each span $s^{begin:end}$, we introduce a phrasal node if the chart cell is not defined, i.e. the span does not consist of well-formed dependency. We restrict the maximum length of a span covered by a phrasal node to L. For each pair of sub-spans $s^{begin:mid}$ and $s^{mid:end}$, where they are rooted at v_l and v_r respectively we add an incoming binary edge e if any of v, v_l, or v_r is a phrasal node. Algorithm 2 shows the pseudo code.

2.3. Annotation of virtual node using dependency label

The translation probability of fine-grained translation rules is more accurate than that of a coarse one [21]. It is also beneficial in terms of efficiency because fine-grained translation rules reduce the search space by constraining the applicable rules. Therefore, we annotate the virtual nodes in $\mathcal{F}$ using dependency labels that represent the dependency relation between the head and the dependent word.

An annotation of a virtual node v for a span $s^{begin:end}$ is a set of dependency labels $ANN(v) =$

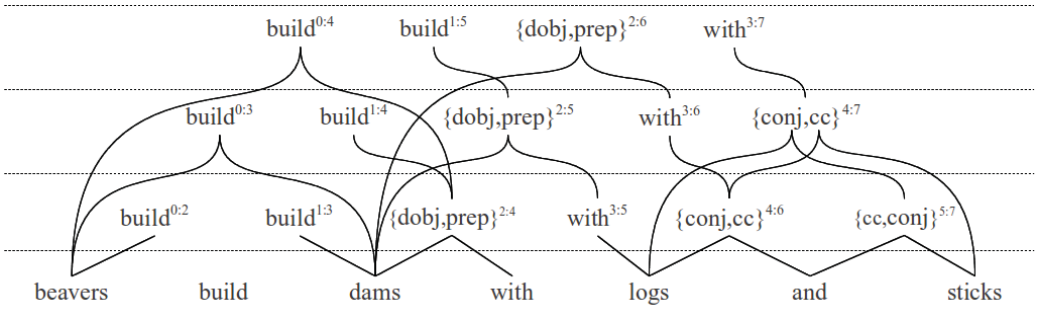


Figure 2: A part of the chart of the binarized dependency forest for the example dependency tree in Figure 1. The dotted lines represent the rows in a chart, and the nodes in a row represent the cells the rooted at these nodes. The solid lines are the incoming binary edges of the binary nodes. For each root node v which covers more than two words, we denote the covered span to $v^{begin:end}$ for clarity. The virtual nodes have annotation using dependency labels as explained in Section 2.3. Note that a binary node can have more than one incoming binary edges, e.g. $\{conj, cc\}^{4:7}$.

Algorithm 2: Augment Phrasal Nodes

function *Augment($\mathcal{F}$, L, n)*
input : A binarized forest $\mathcal{F}$, the maximum phrase length L, the sentence length n
output: A binarized forest $\mathcal{F}'$ with phrasal nodes
for $row = 2 \ldots min(L, n)$ **do**
  for $col = row \ldots n$ **do**
    if $row \leq L$ `and` $chart[row, col]$ `is not defined` **then**
      create a phrasal node v
      $chart[row, col] \leftarrow v$
    end
    else
      $v \leftarrow chart[row, col]$
    end
    for $i = 0 \ldots row$ **do**
      $v_l \leftarrow chart[i, col - row + i]$
      $v_r \leftarrow chart[row - i - 1, col]$
      if `any of` v, v_l, `or` v_r `is a phrasal node` **then**
        create an incoming binary node
        $e = \langle v, v_l, v_r \rangle$
        $IN(v) \leftarrow IN(v) \cup \{e\}$
      end
    end
  end
end

$\bigcup_{j=begin+1}^{end} LABEL(w_j)$. Note that we merge duplicated relations if there are more than two modifiers. Thus it abstracts the dependency relations of the covered words, for example, the modifiers consist of a coordination structure such as "logs and sticks" in the example. When there exist more than two preposition phrases, our proposed method also takes advantage of the abstraction. Since a coordination structure or a the number of preposition phrases can be long arbitrarily, merging duplicated relations minimizes the variation of the annotations, and increases the degree of f the generalization.

2.4. Extraction of translation rule

We extract tree-to-string translation rules from the binarized forest as proposed in [13] after we identify the substitution sites, i.e., frontier nodes. A binary node is a frontier node if a word in the corresponding source span has a consistent word alignment, i.e. there exists at least one alignment to the target and any word in the target span does not aligned to the source word out of the source span. For example, since build$^{0:2}$ has inconsistent word alignment in Figure 3, it is not a frontier node. The identification of the frontier nodes in $\mathcal{F}$ is done by a single post-order traversal.

After we identify the frontier nodes, we extract the minimal rules from each frontier node [8]. Figure 4 shows the minimal rules extracted from the example sentence. For each frontier node v, we expand the tree fragment until it reaches the other frontier nodes. For each tree fragment, we compile the corresponding target words, and substitute the frontier nodes with the labels. If a virtual node is the root of a tree fragment, we do not substitute the frontier nodes that cover length-1 spans. For example, R2, R5, R6, R8 and R9 have length-1 spans that is not substituted. The extraction of the minimal rules takes linear time to the number of the nodes in $\mathcal{F}$, thus the length of the sentence.

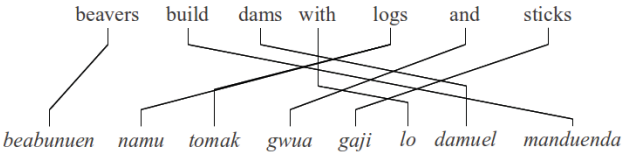


Figure 3: An example of word alignment and target sentence.

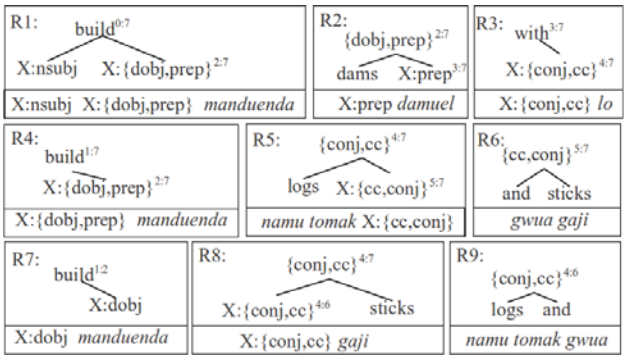


Figure 4: The minimal translation rules. Each box represents the source tree fragment (above) and the corresponding target string (below) with mapping for substitution sites (X).

We also extract composed rules in order to increase the coverage of the extracted translation rules [7]. We believe that the composed rules also prevents the over-generalization of the binarized dependency forest. For each tree fragment in the minimal rules, we extend the the tree fragment beyond the frontier nodes until the size of the tree fragment is larger than a threshold. When we restrict the size, we do not count the non-leaf virtual nodes. We also restrict the number of the extension for each tree fragment in practice. Figure 5 shows two composed rules that extend the tree fragments in R1 and R8, respectively.

3. Experiments

We performed the experiments in the OLYMPICS task in IWSLT 2012. The task provided two parallel corpora, one from the HIT Olypic Trilingual Corpus (HIT) and the other from the Basic Tranvel Expression Corpus (BTEC). We only carried out our experiment with the official condition, i.e. training data limited to supplied data only. As the size of training data sets in the HIT and BTEC is relatively small, we regards the 8 development data sets in the BTEC corpus also as training corpora. Each development corpus in the BTEC corpus has multiple references and we duplicated the source sentences in Chinese for the reference sentences in English. One development set (Dev) was used for tuning the weights in the log-linear model and the other development set (DevTest) was used for testing the translation quality. Finally, the formal runs were submitted by translating the evaluation corpus. Table 1 summarizes the statistics of corpora we used.

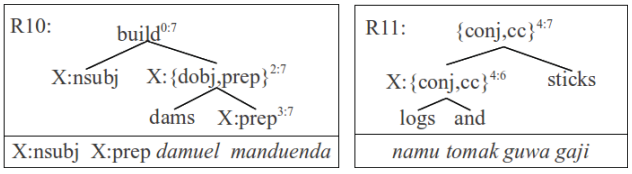


Figure 5: Two composed translation rules.

Table 1: Corpus statistics of the corpora. Sentence column shows the number of sentence pairs, and Source and Target column shows the number of words in Chinese and English, respectively.

	Sentence	Source	Target
Train	111,064	911,925	1,007,611
Dev	1,050	9,499	10,125
DevTest	1,007	9,623	10,083
Test	998	9,902	11,444

Table 2: The official evaluation results of the submitted runs. P is the primary run and C is the contrastive run. M is a phrase-based SMT using Moses with lexicalized reordering and H is Hierarchical phrase-based SMT using Moses-chart.

	BLEU	NIST	TER	GTM	METEOR
P	0.1203	3.7176	0.7999	0.4352	0.3515
C	0.1031	3.4032	0.8627	0.4207	0.3163
M	0.1666	4.3703	0.6892	0.4754	0.4168
H	0.1710	4.4841	0.6817	0.4803	0.4182

We compared the effectiveness of our proposed methods in two different settings. The primary run fully utilized the methods described in Section 2. The contrastive run, on the other hand, skipped the augmentation of phrasal nodes described in Section 2.2. Therefore, the translation rules used in the contrastive run only included tree fragments that satisfies the well-formed dependency. We denoted the contrastive run as the baseline in the next section. We also compared the submitted runs with a phrase-base SMT with lexicalized reordering and a hierarchical phrase-based SMT using Moses. Table 2 shows the evaluation results using various metrics following the instruction provided by the task organizer (README.OLYMPICS.txt). Please refer the details in the overview paper [5].

For both primary and contrastive runs, we implemented a forest-to-string translation system using cube pruning [11] in Java. The implementation of our decoder is based on a log-linear model. The feature functions are similar to hierarchical PBSMT including a penalty for a glue rule, as well as bidirectional translation probabilities, lexical probabilities, and word and rule counts. For the translation probabilities, we applied Good-Turing discounting smoothing in order to prevent over-estimation of sparse rules. We also restricted the maximum size of a tree fragment to 7, and the number of the extension to 10,000.

For an Chinese sentence, we used a CRFTagger to obtain POS tags, and a chart parser to obtain a dependency tree developed in our laboratory. The F-measure of the CRFTagger is 95% and the unlabelled arc score (UAS) of the parser is 87%. We used GIZA++[17] to obtain bidirectional word alignments for each segmented parallel corpus, and applied the grow-diag-final-and heuristics. For tuning the parameter of a log-linear model, we utilized an implementation of min-

imum error rate training [16], Z-MERT [24]. We built the n-gram language model using the IRSTLM toolkit 5.70.03 [4], and converted in binary format using KenLM toolkit [9].

4. Discussion

The augmentation of the phrasal nodes (primary run) outperformed the baseline (contrastive run) in all evaluation metrics. However, both our approaches underperformed any of Moses systems. We suspected the reasons as follows:

- Over-generalization of the dependency structure causes a lot of incorrect reordering, although we annotate the virtual nodes using dependency labels.
- Over-constraint of the tree structure makes a lot of translations impossible that are possible with phrase-based models.
- Parsing error affects the extraction of translation rules and decoding, which are inevitable.

Besides, there are many out-of-vocabulary in all systems due to the relatively small size of the training data. We hope more data in the HIT and BTEC corpora will be available in the future.

5. Conclusion

We participated in the OLYMPICS task in IWSLT 2012 and submitted two formal runs using a forest-to-string translation system. Our primary run achieved better translation quality than our contrastive run, but worse than a phrase-based and a hierarchical system using Moses.

6. Acknowledgements

This work was supported in part by the Korea Ministry of Knowledge Economy (MKE) under Grant No.10041807 and under the "IT Consilience Creative Program" support program supervised by the NIPA(National IT Industry Promotion Agency)" (C1515-1121-0003), in part by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korean government (MEST No. 2012-0004981), in part by the BK 21 Project in 2012.

7. References

[1] DeNero, J., Bansal, M., Pauls, A., and Klein, D. (2009). Efficient parsing for transducer grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 227–235, Boulder, Colorado. Association for Computational Linguistics.

[2] Ding, Y. and Palmer, M. (2005). Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 541–548, Ann Arbor, Michigan. Association for Computational Linguistics.

[3] Eisner, J. (2003). Learning non-isomorphic tree mappings for machine translation. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 205–208, Sapporo, Japan. Association for Computational Linguistics.

[4] Federico, M., Bertoldi, N., and Cettolo, M. (2008). Irstlm: an open source toolkit for handling large scale language models. In *Proceedings of Interspeech*, Brisbane, Australia.

[5] Federico, M., Cettolo, M., Bentivogli, L., Paul, M., and Stüker, S. (2012). Overview of the IWSLT 2012 Evaluation Campaign. Proc. of the International Workshop on Spoken Language Translation.

[6] Fox, H. J. (2002). Phrasal cohesion and statistical machine translation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 304–3111, Stroudsburg, PA, USA. Association for Computational Linguistics.

[7] Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., and Thayer, I. (2006). Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia. Association for Computational Linguistics.

[8] Galley, M., Hopkins, M., Knight, K., and Marcu, D. (2004). What's in a translation rule? In Susan Dumais, D. M. and Roukos, S., editors, *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA. Association for Computational Linguistics.

[9] Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

[10] Huang, L. (2006). Statistical syntax-directed translation with extended domain of locality. In *In Proc. AMTA 2006*, pages 66–73.

[11] Huang, L. and Chiang, D. (2005). Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, Parsing '05, pages 53–64, Stroudsburg, PA, USA. Association for Computational Linguistics.

[12] Huang, L., Zhang, H., Gildea, D., and Knight, K. (2009). Binarization of synchronous context-free grammars. *Comput. Linguist.*, 35(4):559–595.

[13] Mi, H. and Huang, L. (2008). Forest-based translation rule extraction. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 206–214, Honolulu, Hawaii. Association for Computational Linguistics.

[14] Mi, H., Huang, L., and Liu, Q. (2008). Forest-based translation. In *Proceedings of ACL-08: HLT*, pages 192–199, Columbus, Ohio. Association for Computational Linguistics.

[15] Mi, H. and Liu, Q. (2010). Constituency to dependency translation with forests. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1433–1442, Uppsala, Sweden. Association for Computational Linguistics.

[16] Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

[17] Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 440–447, Stroudsburg, PA, USA. Association for Computational Linguistics.

[18] Shen, L., Xu, J., and Weischedel, R. (2008). A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL-08: HLT*, pages 577–585, Columbus, Ohio. Association for Computational Linguistics.

[19] Tu, Z., Liu, Y., Hwang, Y.-S., Liu, Q., and Lin, S. (2010). Dependency forest for statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1092–1100, Beijing, China. Coling 2010 Organizing Committee.

[20] Wang, W., May, J., Knight, K., and Marcu, D. (2010). Re-structuring, re-labeling, and re-aligning for syntax-based machine translation. *Comput. Linguist.*, 36(2):247–277.

[21] Wu, X., Matsuzaki, T., and Tsujii, J. (2010). Fine-grained tree-to-string translation rule extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 325–334, Uppsala, Sweden. Association for Computational Linguistics.

[22] Xie, J., Mi, H., and Liu, Q. (2011). A novel dependency-to-string model for statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 216–226, Edinburgh, Scotland, UK. Association for Computational Linguistics.

[23] Xiong, D., Liu, Q., and Lin, S. (2007). A dependency treelet string correspondence model for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 40–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

[24] Zaidan, O. F. (2009). Z-mert: A fully configurable open source tool for minimum error rate training of machine translation systems. *Prague Bulletin of Mathematical Linguistics*, 91:79–88.

[25] Zhang, H., Fang, L., Xu, P., and Wu, X. (2011). Binarized forest to string translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 835–845, Portland, Oregon, USA. Association for Computational Linguistics.

[26] Zhang, H., Huang, L., Gildea, D., and Knight, K. (2006). Synchronous binarization for machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 256–263, New York City, USA. Association for Computational Linguistics.

[27] Zhang, H., Zhang, M., Li, H., Aw, A., and Tan, C. L. (2009). Forest-based tree sequence to string translation model. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 172–180, Suntec, Singapore. Association for Computational Linguistics.

[28] Zhang, H., Zhang, M., Li, H., and Chng, E. S. (2010). Non-isomorphic forest pair translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 440–450, Cambridge, MA. Association for Computational Linguistics.

Romanian to English Automatic MT Experiments at IWSLT12
(System Description Paper)

Ştefan Daniel Dumitrescu, Radu Ion, Dan Ştefănescu, Tiberiu Boroş, Dan Tufiş

Research Institute for Artificial Intelligence
Romanian Academy, Romania
{sdumitrescu,radu,danstef,tibi,tufis}@racai.ro

Abstract

The paper presents the system developed by RACAI for the ISWLT 2012 competition, TED task, MT track, Romanian to English translation. We describe the starting baseline phrase-based SMT system, the experiments conducted to adapt the language and translation models and our post-translation cascading system designed to improve the translation without external resources. We further present our attempts at creating a better controlled decoder than the open-source Moses system offers.

1. Introduction

This article presents the system developed by RACAI (the Research Institute for Artificial Intelligence of the Romanian Academy) for the ISWLT 2012 competition. We targeted the Machine Translation track of the TED task, Romanian to English translation.

We had access to the following resources:

- In-domain parallel corpus: 142K sentences; 13MB size; TED RO-EN sentences [6].
- Out-of-domain parallel corpus: 550K sentences; 85MB size; Europarl (juridical domain) and SETimes (news domain) RO-EN sentences.
- Out-of-domain monolingual corpus (English): 168M sentences; 26GB size; mostly news domain EN sentences.
- Development set: 1.2K RO-EN sentences (TED tst2010 file)
- Test set: 3K RO only sentences (TED tst2011 and tst2012 files).

Before attempting any translation experiments, the available resources had to be preprocessed. This involves first correcting the Romanian side of the parallel corpora as to obtain the highest possible quality Romanian-side text and then annotate both the Romanian and English sides.

Thus, the first preprocessing step involves automatic text normalization. Historically, due mainly to technical reasons regarding the code-page available in earlier versions of the Windows operating system, the letters *ş* and *ţ* in the Romanian language were initially written as *ş*, *ţ* (with a cedilla underneath – old, incorrect style) and later as *ș*, *ț* (with a comma underneath – correct style). As such, we have several resources with incompatible diacritics for these two letters. All old-style letters have been converted to the new style. The second correction to be made is due to the Romanian orthographic reform from 1993 which re-establish the orthography used until 1953, according to which (among the others) the inner letter "î", has been replaced by "â (ex: p**î**ine is written correctly as p**â**ine). Older texts have been corrected to the current orthography using an internally developed tool that uses a 1.5 million word lexicon of the Romanian language backing-off a rule-based word corrector in case the lexicon might not contain some words.

The third and final necessary correction concerned texts that do not have diacritics. In the provided resources, both in-domain and out-of-domain corpora contain several groups of sentences that have not diacritics. Restoring diacritics is a rather difficult task, as a misplaced or missing diacritic can have dramatic effects starting from change of definiteness of a noun (for example) to changing an entire part-of-speech of a word, yielding sentences that lose their meaning. Using an internally developed tool [19] we were able to carefully restore diacritics where they were missing. Even though the tool is not 100% accurate, it is better to introduce a small amount of error rather than have several words without diacritics that will create more uncertainty in the translation process later on.

The second step of the preprocessing phase is the automatic annotation of both Romanian and English texts. Using also an internally developed tool named TTL [11] we are able to tokenize sentences and annotate each word with its lemma, two types of part-of-speech tags: morpho-syntactic descriptors (MSDs) and a reduced tag set (CTAGs), and different combinations of them. The tags themselves follow the Multext-East lexical standard [8] and the tiered tagging design methodology [20].

As an example, for the English sentence "We can can a can." we obtain the following annotation:

We|we^Pp|we^PPER1|Pp1-pn|PPER1
can|can^Vo|can^VMOD|Voip|VMOD
can|can^Vm|can^VINF|Vmn|VINF
a|a^Ti|a^TS|Ti-s|TS
can|can^Nc|can^NN|Ncns|NN
.|. ^PE|.^PERIOD|PERIOD|PERIOD

The first of the five factors for each word is the word itself (the surface form). The second factor is the lemma of the word, linked by the "^" character, to its first two positions in the MSD tag (grammar category and type). The third factor is the lemma linked to the CTAG, followed by the MSD (fourth factor) and CTAG (fifth factor).

The TTL tool has other advanced features that make it desirable for machine translation. Sometimes it is better for certain phrases to be considered as a single entity. For

example, phrases like "… do something to **the other**, …" are automatically linked together by an underscore and annotated as: "the_other|the_other^Pd|the_other^DMS|Pd3-s|DMS". Other examples of automatically extracted phrases: "in_terms_of", "the_same", "a_little", "a_number_of", |"out_of", "so_as", "amount_of_money", "put_down", "dining_room", etc. The same tokenization, phrase extraction and annotation process is performed for the Romanian language.

The third and last step of the preprocessing phase is true-casing all available resources. True-casing simply means lower-casing the first word in every sentence, where necessary. A model is trained on available data, learning what words should not be lower-cased, as acronyms or proper nouns, and applied back to the data. True-casing benefits automatic machine translation when building both the translation model and the language model by reducing the number of surface forms for each possible word.

2. System description

In this section we present the steps and the experiments performed to create and adapt our MT system to the TED task. We start with a basic phrase-based statistical MT system with default parameters in order to establish a baseline (section 2.1); we then experiment with different adaptations of the language models and the translation tables used (2.2 – 2.4); we perform a parameter setting search to find the combination of parameters that will maximize the translation score (2.5); finally, we apply a technique we call "cascaded translation" [21] to attempt to correct some of the translation errors (section 2.6).

Before describing the steps and experiments performed, we must specify that unless explicitly otherwise stated, the following BLEU scores are all obtained on comparing the English translation of the tst2012 file from the test set to an English reference file we manually created starting from the English subtitles for each respective TED talk. We later obtained access to the English tst2011 file from the same test set, but we did not have enough time to re-run the experiments on this official reference file. We are confident that our tst2012 reference file is very similar to the official file given the correlated scores of our results and those given by the official evaluation as we later present.

2.1. Baseline system

We start with the standard Moses [12] system. We trained the system on the in-domain data (the provided TED RO-EN parallel corpus), as well as building a language model on the English side of the same corpus.

The language model was built using the SRILM toolkit [17]: surface-form, 5-gram, interpolated, using Knesser-Ney's smoothing.

This baseline system yielded a 25.34 BLEU score.

2.2. Direct Language-Model adaptation experiment

The first attempted language model adaptation method is the direct, perplexity-based measure: given the tokenized and true-cased English resources, extract sentences with the lowest perplexity and add them to the in-domain language model.

The procedure first requires that all the English resources (both from the parallel corpora and the monolingual corpora) be merged into a single file. The resulting 27 GB file had around 28 billion tokens contained in almost 168 million sentences. Each sentence was perplexity measured against the in-domain language model. Then, the file was sorted based on sentence perplexity, lowest first.

Starting with the initial in-domain language model that obtained 25.34 BLEU points we added incrementally batches of 1 million sentences, re-translated and noted the score increase/decrease. We observed a non-linear increase up to 10 million added sentences, followed by a rather slow BLEU decrease. We found that the best performing language model constructing using this method contains 10.6 million sentences, 142,000 coming from English side of the in-domain corpus. The score obtained using this method was 28.04, a significant 2.70 point increase from the baseline score of 25.34.

2.3. Indirect Language-Model adaptation experiment

The direct language model adaptation works very well when a specific domain is given and a language model can be built on that domain to provide a perplexity reference for new sentences. If this information is not available, one could try to alleviate the problem in various ways.

Our idea in this indirect language model adaptation is to check whether we could use the information available in the test set to create a better language model.

This, however, presented a problem: while in the test set we are only given the source Romanian sentences that need to be translated, the English language model should be adapted with sentences for which translations are not yet available. Thus, we came up with the following four step procedure to attempt indirect adaptation of the target language model by generating English n-grams from Romanian n-grams:

Step 1: Count the n-grams from the Romanian sentences in the test set. Counting was done up to 5-grams, ignoring functional unigrams (determiners, prepositions, conjunctions, etc.).

Step 2: Having the translation table already created from the base model, attempt to "translate" the n-grams from Romanian to English. Parse the translation table, look up each Romanian n-gram and retain all the equivalents in English. This will increase the number of n-grams several times. At the end of this step we will have a list of English n-grams.

Step 3: Based on the list of English n-grams, iterate over each sentence in the file containing all the English data (27 GB) and count matching n-grams. In order to select the most promising sentences, we have created a few different scoring

methods: **(1)** Standard measure, where if we find a matching n-gram we increase the score of that sentence by n (e.g. if we find four unigrams and two trigrams we increase the score by 4*1+2*3 = 10); **(2)** Standard normalized (Std. Div.) measure, where we divide the standard measure by the length of the sentence in order to compensate for very long sentences likely to have more n-gram matches; **(3)** Square measure, where if we find a matching n-gram we increase the score of the sentence by the square of n (ex: for 4 unigrams and two trigram the score would be $4*1^2+2*3^2=22$); **(4)** Square normalized (Square Div.) measure, dividing the Square measure by the length of the sentence in order to compensate for long sentences. We thus sort in decreasing order each of the English sentences based on our proposed measures, obtaining 4 large English files.

Step 4: From each of the four sorted files, we take incremental batches of sentences and build adapted language models of larger and larger sizes.

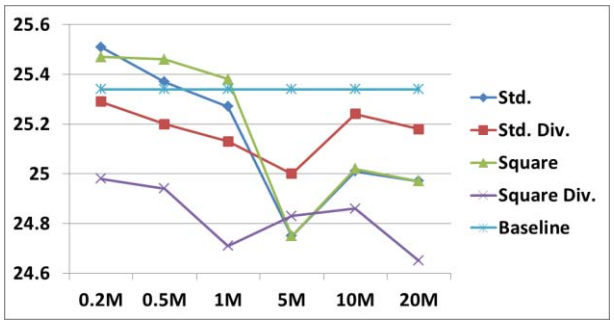


Figure 1: Indirect LM adaptation BLEU scores

Figure 1 presents our experimental results. We manage to obtain just a very slight increase over the baseline of 25.34 when adding just a small number (less than 200,000 sentences in addition to the TED English sentences). This experiment shows that it is possible to adapt a language model starting only from the sentences that need to be translated, but also reveals that there is a fine-grained point over which adding more sentences, using our measures, actually degrades performance. Also, it should be noted that for both direct adaptation using the perplexity measure and the indirect adaptation method, the peak of the graph can be determined only if the target (reference) development set, on which to measure the BLEU score, is available. However, our indirect LM adaptation allows increasing the size of the available development set considering the monolingual test set.

2.4. Translation model adaptation experiment

With the next experiment we attempt to adapt the translation model (TM) using data available from the out-of-domain corpora.

Based on the previous experiments we used perplexity as the similarity measure of choice. We attempted two adaptations based on both the source and the target languages. We built two language models: the first was built on the English side of the TED corpus while the second on the Romanian side. Using each language model in turn, we calculated the perplexity of each corresponding sentence from every translation unit in the out-of-domain parallel corpora. Then we sorted the corpora's translation units according to the perplexity scores of English and Romanian parts. For example, we measured the perplexity of the Romanian side of Europarl & SETimes corpora vs. the language model built on the Romanian side of TED, and then sorted Europarl & SETimes by the ascending perplexity of their Romanian sides (similarly for English).

We made experiments on TM adaptation selecting parallel data according to the similarity with each language model. We took increments of 5% of the sorted parallel corpora and added them to the TED corpus and noted the translation scores. For this experiment we used the development set (tst2010) which had a translation baseline score of 28.82.

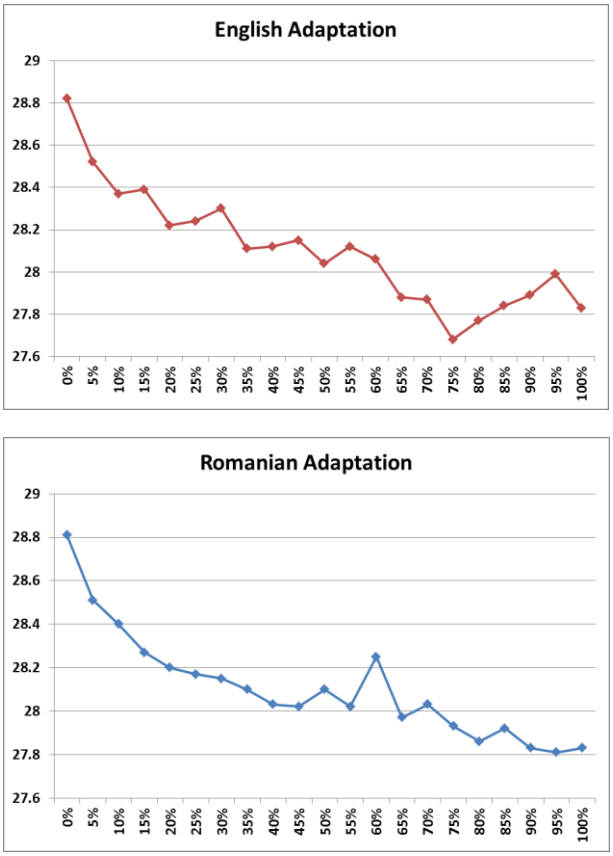


Figure 2: English and Romanian TM adaptation graphs

The experiments show that even adding 5% of the best sentences (based on perplexity) of the Europarl and SETimes corpora decreases the translation score by a significant 0.3 BLEU points. The decrease is rather consistent when trying to adapt the translation model starting from either the Romanian or the English language, clearly stating the conclusion that neither Europarl which is a juridical corpus nor SETimes which is news-oriented do contain parallel sentences that positively contribute to the translation model firmly located in a free-speech domain. After this result it was clear that further attempting to adapt the translation model using the provided out-of-domain corpora was impractical. Using the LEXACC comparable data extraction tool [18] with the TED and Europarl+Setimes corpora as search space supported the

previous observation that the out-of-domain data was too distant from the in-domain-data to be useful in TM adaptation.

2.5. Finding the best translation system

Having experimented with adapting both the language model and the translation model, we started searching for the parameter combination that will maximize the translation score.

The systematic search included the following parameters:

- Translation type
- Alignment model
- Reordering model
- Decoding type and sub-parameters

The translation type refers to which word factors were used and the translation path itself. We started from the simple surface-to-surface translation, gradually using more factors such as part-of-speech (both MSDs and CTAGs, available after using the TTL tool in the corpus preprocessing phase), lemma or different combinations of lemmas and part-of-speech tags. The translation path meant using direct, single-step translation (ex: translation of surface-surface, translation of surface and part-of-speech to surface, etc.) or multiple step translation including generation phases (ex: translation of lemma to lemma then generation of part-of-speech from lemma, then translation of part-of-speech to part-of-speech and finally generation of the surface form from lemma and part-of-speech).

For the alignment and reordering models we also tried using several combinations of word factors.

Finally, for the decoder, we systematically modified the decoding parameters for the default decoder (beam size, stack size) and the decoding model (cube-pruning, minimum-bayes-risk and lattice-minimum-bayes-risk, each with its individual parameters).

After conducting an extended search of about 60 experiments in which parameters were systematically modified we obtained a score of 29.24, again a significant increase from the baseline system with the adapted language model for which we obtained only 28.04. These two figures are unofficial results computed (as mentioned in Section 2) on our hand made reference for tst2012. The best combination of parameters was: a single-step direct translation of surface form to surface form; an alignment model using the "*union*" heuristic; a reordering model using the default "*wbe-msd-bidirectional-fe*" heuristic; the alignment and reordering model based only on the lemma and the reduced MSD, not on the surface forms; a lattice-minimum-bayes-risk decoder with an increased stack size of 1000.

The search was performed using the adapted language model described in section 2.2 and a translation model based only on the TED in-domain corpus.

2.6. Cascaded system translation experiment

Having obtained the optimum parameters so far, we applied a procedure we previously developed [21] to try to further improve the translation score without adding or using any external data. We hypothesize that training a second phrase-based statistical MT system on the data that was output by our initial system, this second system will correct some of the errors the initial system made.

The first step in building the second system of the cascade is based on using the first system to translate the Romanian side of its own RO-EN training corpus. This will yield a translated–EN-EN parallel corpus on which the second system is trained upon. The cascaded system is now ready to be used.

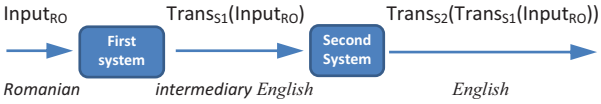


Figure 3: Cascaded system diagram

The diagram shows how the cascading procedure works. The test set is initially translated from Romanian into intermediary English. Next, this intermediary translation is fed to the second system which translates the intermediary English to "final" English. The "final" English is then evaluated against the reference to determine the effect of the cascade: how much improvement was achieved, if any.

We obtained a net increase of 0.36 points bringing the new BLEU score to 29.60 (using our tst2012 manually created reference file). In this particular case the cascade changed 22 percent of the total of 1733 sentences, 12% for the better and 10% for the worse, the rest of the sentences being unaffected.

Table 1: Cascading effect

S1	After system 1	S2	After system 2	Reference
0.57	the microprocessor **. it 's a miracle** the personal computer is a miracle .	1.00	the microprocessor **is a miracle .** the personal computer is a miracle .	the microprocessor is a miracle . the personal computer is a miracle .
0.53	and the reasons **delincvenților** online are very easy to understand .	0.7	and the reasons **online criminals** are very easy to understand .	and the motives of online criminals are very easy to understand .
0.47	and **so let me** begin with an example .	0.31	and **let me try to** begin with an example .	and let me begin with one example .

Table 1 shows some of the effects of cascading. In the first example we see a clear improvement from 0.57 to 1.00 of the translation by correctly placing the comma and transforming "it's a" in "is a". The second example shows that sometimes the cascade can correct initially non-translated words: due to Moses's phrase table pruning mechanism, even though the unigram "delincvenților" is present in the training corpus, it does not appear in the first system's phrase table and thus does not get translated. However, it appears in the second phrase table and is subsequently translated. The third example presents a score decrease from 0.47 to 0.31. However, transforming "so let me" to "let me try to", while from

BLEU's perspective vs. the reference translation is a decrease, from a human perspective, the sentence is still fully comprehensible.

Overall, cascading increases the BLEU score usually from a fraction of a BLEU point up to a few BLEU points [21]. For the official evaluation we have submitted for each test file a cascaded system and a non-cascaded system. The official evaluations showed a small increase of 0.04 BLEU (from 29.92 for the standard, un-cascaded system to 29.96 for the cascaded) for the 2011 test file and an increase of 0.21 BLEU (from 26.81 to 27.02) for the 2012 test file, as presented in Table 2 in Section 4.

3. Alternative translation systems

After performing a host of experiments with Moses with different settings as reported in the previous sections, it became clear that the BLEU barrier of around 30% is not going to be easily (and significantly) broken without additional in-domain, parallel data and because of that, we proceeded to refine our own, in-house developed decoders based on Moses-trained phrase tables and language models. The purpose of this endeavor was to come up with a combination/merging scheme of the outputs of several decoders that, we envisaged, would ensure a superior translation when compared to each of the decoders. In what follows, we briefly give the underlying principles of our in-house developed decoders and present their combined output with the best Moses output (see 2.6).

3.1. The first RACAI decoder (RACAI1)

The first RACAI decoder is based on the Dictionary Lookup or Probability Smoothing (DLOPS) algorithm [4], primarily used for phonetic transcription of out-of-vocabulary (OOV) words. The original algorithm works by adjoining adjacent overlapping sequences of letters that have corresponding transcription equivalents inside a lookup table. The overlapping sequences are selected by finding a single split position (called *pivot*) inside a sequence that will maximize a function called the *fusion score* (described in the original article). The algorithm would recursively produce the phonetic transcriptions of the pivot left and right sequences either by directly returning transcription candidates from the lookup table (if there are any transcription candidates) or by further recursive building the transcriptions. Because of the similarities that arise between the phonetic transcription and MT [13], we thought of adapting DLOPS to perform decoding for MT. There were some limitations of the initial algorithm that needed to be eliminated:

1. We modified the system to use a Berkeley Data Base (BDB) for lookup to be able to cope with large phrase tables;
2. The algorithm looks for the sequence of words with the highest translation score. The indexes of the left-most and right-most words are considered the pivots of the recursions. The DLOPS had to be modified to search for two pivots instead of one;
3. We added word reordering capabilities (this was not an issue in phonetic transcription).

For each sequence of words that has a corresponding entry in the translation table, we retain all possible candidates and, returning from the recursive call, we get the Cartesian product of the translations from the left, center and right source word sequences. Because this translation set usually has a large number of candidates, we score each translation candidate by summing the *S* value for the left, center and the right sub-candidate:

$$S = \theta_1\varphi(f\,|\,e) + \theta_2\varphi(e\,|\,f) + \theta_3\lambda(f\,|\,e) + \theta_4\lambda(e\,|\,f) + \theta_5 LM(e)$$

where $\varphi(f\,|\,e)$ is the Moses-based phrase table inverse phrase translation probability, $\varphi(e\,|\,f)$ is the direct phrase translation probability, $\lambda(f\,|\,e)$ is the inverse lexical similarity score, $\lambda(e\,|\,f)$ is the direct lexical similarity score and $LM(e)$ is the language model score (at word level) of the translation candidate. The weights $\theta_{1,\ldots,5}$ are computed with the Minimum Error Rate Training (MERT) procedure from the Z-MERT package [23].

3.2. The second RACAI decoder (RACAI2)

This first step of this decoder is to collect a set *C* of source sentence non-overlapping segmentations according to the phrase table, giving priority to segmentations formed with the longer spans of adjacent tokens from the input sentence. For the input sentence *S* with *n* tokens, considering at most *k* adjacent tokens (called "a token span") for which we find at least one translation in the phrase table, $k < n$, the total number *N* of non-overlapping segmentations is

$$N_k(n) = \sum_{i=1}^{k} N_k(n-i)$$

For k = 2 this is the well-known Fibonacci series and it is obvious that $N_k(n) > N_2(n)$ for $k > 2$. It can be shown that

$$N_2(n) \geq c\left(\frac{3}{2}\right)^n$$

for some positive constant *c* and this tells us that one cannot simply enumerate all the segmentations of the source sentence according to the phrase table because the space is exponentially large. Thus, our strategy is to choose a segmentation $P = \left\{w_i^j \,|\, 1 \leq i < j \leq n\right\}$, where w_i^j is the token span from the index *i* to index *j* in the source sentence *S* which has at least one translation in the phrase table, such that $|P|$ is minimum.

The second step of the decoder is to choose, for each partial translation h_1^j (up to the current position *j* in *S*) and input token span $w_{j+1}^k \in P$, the best translation h_{j+1}^k from the phrase table such that two criteria are simultaneous optimized:

1. The translation scores of h_{j+1}^k from the Moses phrase table are maximum;
2. The language model (at word form level and POS tag level) score of joining h_1^j with h_{j+1}^k is also maximum.

What we did, was to actually compute an interpolated score as in the case of the previously described decoder with weights tuned with Z-MERT.

The third and final step of the RACAI2 decoder was to correct the raw, statistical translation output to eliminate the translation errors that were observed to be frequent and that violate the English syntactic requirements (mainly due to the inexistence of a reordering mechanism). This is a rule-based module that works only for English. Examples of frequent mistakes include:

- translating the valid sequence "noun, adjective" from Romanian into the same, invalid, sequence in English;
- translating the valid sequence "noun, demonstrative determiner" from Romanian into the same, invalid, sequence in English;
- translating the valid sequence "noun, possessive determiner" from Romanian into the same, invalid, sequence in English.

The astute reader has noticed that the optimization criteria from the second step of this decoder consider local maxima. One immediate improvement is to replace the current optimization step by a Viterbi global optimization [22].

3.3. Combining translations from Moses, RACAI1 and RACAI2

Having three decoders that produce different translations for the same text, it is tempting to consider their combination in order to find a better translation. Generating the best translation for a text (sentence or paragraph), given multiple translation candidates obtained by different translation systems, is an established task in itself. Even the simplest approach of deciding which candidate is the most probable translation has been proven to be difficult [1, 5, 16]. The different solutions described in the literature are focused on re-ranking merged N-best lists of translation candidates, word-level and phrase-level combination methods [2, 6, 8, 14].

Our approach is a phrase-level combination method and exploits the linearity of the candidate translations given by the systems we employed. First, we split the source (i.e. Romanian) sentence into smaller fragments which are considered to be stand-alone expressions that can be translated without additional information from the surrounding context. For considerations regarding speed, this is done by using certain punctuation marks and a list of words (split-markers) that can be considered as fragment boundaries (e.g. certain conjunctions, prepositions, etc.). Every fragment must contain at least two words, out of which one should not be in the above mentioned list of split-markers. For example, the sentence "*s-a făcut de curând un studiu printre directorii executivi în care au fost urmăriți timp de o săptămână.*"[1] is split into 3 fragments: "*s-a făcut de curând un studiu*", "*printre directorii executivi*" and "*în care au fost urmăriți timp de o săptămână.*"

[1] English: "there was also a study done recently with CEOs in which they followed CEOs around for a whole week."

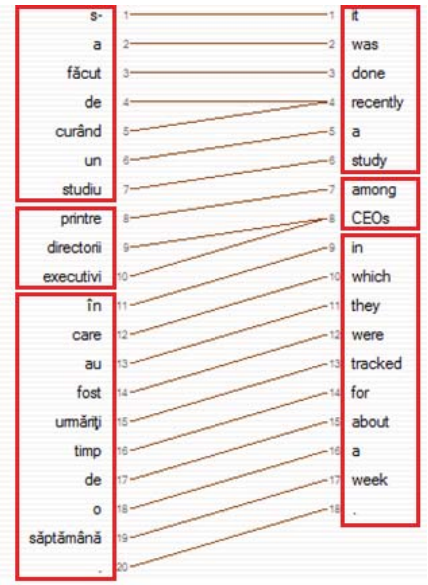


Figure 4: DTW Alignment helps identifying the corresponding translations of the source fragments

In the next step, taking into account the linearity of the translations, we use Dynamic Time Warping (DTW) algorithm [3,15] to align the source sentence with the current translation candidate. The cost function is defined between a source word w_s and a target word w_t as: $c = 1 - te(w_s, w_t)$, where *te* is the translation equivalence score in the existing dictionary. Taking into account the source fragments and the alignments obtained with DTW, we are able to pinpoint the translation for each of fragment. For our example we have the following candidates:

Table 2: Translation candidates for the source fragments

Translation/ system	*s-a făcut de curând un studiu*	*printre directorii executivi*	*în care au fost urmăriți timp de o săptămână.*
Moses	it has recently made a study	**among the CEOs**	in which they were followed for about a week.
RACAI 1	**it was done recently a study**	among CEOs	in which they were tracked for about a week.
RACAI 2	was done recently a study	among execs executives	**in which have been tracked for about a week.**

We modeled the selection process by a HMM. The emission probabilities are given by a translation model learned with Moses, while the transition probabilities are given by a language model learned using SRILM. The combiner uses the Viterbi algorithm [22] to select the best combination of the translation candidates and generate a "better" translation. For our example, the best path found by the Viterbi algorithm passes through the bolded fragments in the above table, yielding the final translation: "it was done recently a study among the CEOs in which have been tracked for about a week.". Yet, this translation is deficient because of the missing

pronoun "they" (existing in Moses and RACAI1 outputs) in the translation for the third fragment.

We have also experimented with combination at the whole-translation (sentence) level (as opposed to phrase-level) and we tried the following:

1. selecting the translation which had the lowest perplexity as measured by the language model of the best Moses setting;
2. selecting the translation which had the largest averaged BLUE score when compared to the other two translations;
3. selecting the translation which had the lowest TERp score when compared to its cascaded version.

The phrase-level combination method outperforms the first sentence-level combination method and it is close (somewhat better) to the other two sentence-level combination methods. We also estimated the maximum gain (an "oracle" selection) from the sentence-level combination by choosing the translation which had the highest BLUE against our reference for tst2012 (see Table 3). We have thus determined the 32.41 BLUE score which is 2.81 points better than the cascaded Moses (29.60).

Even if the phrase-level combination method does not outperform Moses, our analysis shows that the combiner improves about 22% of the Moses translations with an average increase of the BLEU score of 0.088 points per translation while it deteriorates about 27% of them with an average decrease of the BLEU score of 0.098 points per translation, amounting to a global decrease of only 0.69 BLEU points overall (see Table 3; compare S2 with S5). The rest of the translations remained unchanged after the combination.

4. Conclusions

The paper presented RACAI's machine translation experiments for the IWSLT12 TED track, MT task, Romanian to English translation. In the first part we presented our experiments in building a system based on the Moses SMT package. We evaluated different adaptation types for the language and translation model; we then performed a systematic search to determine the best translation parameters (word factors used, alignment and reordering models, decoder type and parameters, etc.); finally, we applied our cascading model to correct some translation errors made by our best single-step translator. This experiment chain yielded our best model, in the official evaluation (Table 2) obtaining 29.96 BLEU points for the tst2011 test set and 27.02 BLEU point for the tst2012.

The second part of the paper presents our experiments in building two prototype decoders and a translation combiner. The decoders (RACAI 1&2) are based on different strategies than Moses (each presented in its own section), in our attempt to go beyond the difficult to reach baseline set by the best Moses-based model. However, even though we could not exceed yet this baseline, we came rather close to it, given that most of the development work was on adapting the Moses model and allowing only around 3 weeks for the development of the alternative decoders.

The following tables show the official results [9] (case and punctuation included) for the entire test set (tst2011&2012), as well as the results obtained on the reference we built for tst2012 (the official reference was not released at the time of this writing). The tables contain the performance figures for our two Moses-based models (S1 being the best direct translation model we found, while S2 being the S1 model with our cascading technique applied), our two prototype decoders (S3 and S4) and our translation combiner (S5).

Because we have not seen the reference for tst2012, our explanation for the differences among the figures in Table 2 and Table 3 is that our evaluations were performed on lower-case version of the data and mainly due to a different tokenization. While the official tokenization is based on space separation, our tokenization is language aware, considering (among others) multiword expressions and splitting clitics.

Table 2: Official systems evaluation results (case+punctuation)

System	tst2011			tst2012		
	BLEU	Meteor	TER	BLEU	Meteor	TER
S1 (Moses, not-cascaded)	29.92	0.6856	46.388	26.81	0.6443	50.891
S2 (Moses, cascaded)	**29.96**	**0.6844**	**46.701**	**27.02**	**0.6446**	**51.093**
S3 RACAI1	25.31	0.6484	48.845	22.56	0.6085	52.964
S4 RACAI2	-	-	-	21.69	0.6009	56.950
S5 Moses + RACAI1 + RACAI2	-	-	-	25.99	0.6378	51.580

Table 3: Local systems evaluation results (language aware tokenization+no case+punctuation)

System	tst2012
	BLEU
S1 = Moses, not-cascaded	29.24
S2 = Moses, cascaded	29.60
S3 = RACAI1	24.50
S4=RACAI2	23.89
S5 = Moses + RACAI1 + RACAI2	28.91
S6 = Oracle Moses + RACAI1 + RACAI2	**32.41**

5. Acknowledgements

The work reported here was funded by the project METANET4U by the European Comission under the Grant Agreement No 270893.

6. References

[1] Akiba, Yasuhrio, Taro Watanabe, and Eiichiro Sumita. 2002. Using Language and Translation Models to Select the Best among Outputs from Multiple MT systems. In Proc. of Coling, pp. 8–14.

[2] Antti-Veikko I. Rosti, Bing Xiang, Spyros Matsoukas, Richard Schwartz, Necip Fazil Ayan, and Bonnie J. Dorr. 2007. Combining outputs from multiple machine translation systems. In Proc. NAACL-HLT 2007, pp. 228–235.

[3] Bellman R. and Kalaba R. 1959. On adaptive control processes, Automatic Control, IRE Transactions on, vol. 4, no. 2, pp. 1-9.

[4] Boroş T., Ştefănescu, D., Ion, R., 2012. Bermuda, a data-driven tool for phonetic transcription of words, in Proceedings of the Natural Language Processing for Improving Textual Accessibility Workshop (NLP4ITA), LREC2012, Istanbul, Turkey, 2012

[5] Callison-Burch, Chris and Raymond S. Flournoy. 2001. A Program for Automatically Selecting the Best Output from Multiple Machine Translation Engines. In Proc. MT Summit, pp. 63–66.

[6] Cettolo, M., Girardi, C., Federico, M., *WIT3: Web Inventory of Transcribed and Translated Talks*. In Proc. of EAMT, pp. 261-268, Trento, Italy, 2012

[7] Matusov E., Ueffing N., and Ney H., 2006. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment, in Proc. EACL, 2006.

[8] Erjavec, T., Monachini, M. 1997. *Specifications and Notation for Lexicon Encoding.* Deliverable D1.1 F. Multext-East Project COP-106. http://nl.ijs.si/ME/CD/docs/mte-d11f/.

[9] Federico, M., Cettolo, M., Bentivogli, L., Paul, M., Stuker, S.,: *Overview of the IWSLT 2012 Evaluation Campaign*, In Proc. of IWSLT, Hong Kong, HK, 2012

[10] Frederking R., Nirenburg S. 1994. Three heads are better than one. In *Proc. ANLP*, pages 95–100.

[11] Ion, R. 2007. *Word Sense Disambiguation Methods Applied to English and Romanian,* PhD thesis (in Romanian). Romanian Academy, Bucharest, 2007.

[12] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E., Moses: Open Source Toolkit for Statistical Machine Translation, in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, demonstration session, Prague, 2007

[13] Laurent Antoine, Deléglise Paul and Meignie, Sylvain. 2009. Grapheme to phoneme conversion using an SMT system. In Proceedings of INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association, pp. 708--711, Brighton, UK.

[14] S. Bangalore, G. Bordel, & G. Riccardi. 2001. Computing consensus translation from multiple machine translation systems, in Proc. ASRU, 2001.

[15] Senin P. 2008. Dynamic time warping algorithm review, University of Hawaii at Manoa, Tech. Rep.

[16] Zwarts S., Dras M., 2008. Choosing the Right Translation: A Syntactically Informed Classification Approach. In Proc. of Coling, pp. 1153-1160.

[17] Stolcke, A., SRILM - An Extensible Language Modeling Toolkit, in *Proc. Intl. Conf. Spoken Language Processing*, Denver, USA, 2002.

[18] Ştefănescu, D., Ion, R., and Hunsicker, S. 2012. *Hybrid Parallel Sentence Mining from Comparable Corpora*. In Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT 2012), pp. 137—144, Trento, Italy, May 28-30, 2012

[19] Tufiş, D. and Ceauşu, A., DIAC+: A Professional Diacritics Recovering System, in *Proceedings of LREC 2008*, May 26 - June 1, Marrakech, Morocco. ELRA - European Language Resources Association, 2008.

[20] Tufiş, D., Tiered Tagging and Combined Classifiers, in F. Jelinek, E. Nöth (eds) Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence 1692, Springer, 1999, pp. 28-33

[21] Tufiş, D. and Dumitrescu, S.D., Cascaded Phrase-Based Statistical Machine Translation Systems, *in Proceedings of the 16th Conference of the European Association for Machine Translation*, Trento, Italy, 2012.

[22] Viterbi, A.J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13 (2): 260–269. doi:10.1109/TIT.1967.1054010. (note: the Viterbi decoding algorithm is described in section IV.)

[23] Zaidan, O.F., 2009. *Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems*. The Prague Bulletin of Mathematical Linguistics, No. 91:79–88.

The TÜBİTAK Statistical Machine Translation System for IWSLT 2012

Coşkun Mermer, Hamza Kaya, İlknur Durgar El-Kahlout, Mehmet Uğur Doğan

TÜBİTAK BİLGEM
Gebze 41470 Kocaeli, Turkey
{coskun.mermer,hamza.kaya,ilknur.durgar,mugur.dogan}@tubitak.gov.tr

Abstract

We describe the TÜBİTAK submission to the IWSLT 2012 Evaluation Campaign. Our system development focused on utilizing Bayesian alignment methods such as variational Bayes and Gibbs sampling in addition to the standard GIZA++ alignments. The submitted tracks are the Arabic-English and Turkish-English TED Talks translation tasks.

1. Introduction

In the 2012 IWSLT Evaluation Campaign [1], we participated in the TED task for the Arabic-English and Turkish-English language pairs. Our major focus this year was improving the word alignment.

Maximum-likelihood (ML) word alignments obtained using GIZA++ [2] can exhibit overfitting, e.g., rare words can have excessively high alignment fertilities [3], also known as "garbage collection" [2, 4]. Furthermore, ML estimation gives a point-estimate of the parameters, which assumes that the unknown parameters are fixed (as opposed to being a random variable). Finally, the expectation-maximization (EM) method used in obtaining the ML-estimates can get stuck in local optima.

As an alternative approach, in our submission we experimented with the Bayesian approach to word alignment. In the Bayesian framework, the parameters are treated as random variables with a prior distribution. By choosing a suitable prior, we can bias the inferred solution towards what we would expect from our prior knowledge and away from unlikely solutions such as garbage collection.

The remainder of this paper is organized as follows. Section 2 summarizes the word alignment methods and their parameter settings used in our systems. Sections 3 and 4 describe the data used and the common aspects of system development in both language tracks. The specifics of the Arabic-English and Turkish-English submissions and the experimental results are described in Sections 5 and 6, respectively, followed by the conclusions.

2. Word alignment methods

In most commonly-used word alignment methods, such as those used in GIZA++ [2], the model parameters are estimated via EM, which is a ML approach. For this evaluation, we experimented with two additional methods that use a Bayesian approach, where the parameters are treated as random variables with a prior and they are integrated over for alignment inference.

The main difference between the ML and Bayesian approaches to word alignment can be summarized as follows [5]. Given a parallel corpus $\{\mathbf{E}, \mathbf{F}\}$, let $\mathbf{A}$ denote the hidden word alignments. The IBM word alignment models [6] assign a probability to each possible alignment through $P(\mathbf{F}, \mathbf{A}|\mathbf{E}, \mathbf{T})$, where $\mathbf{T}$ denotes the (unknown) translation parameters. The ML solution returns the posterior distribution of the alignments $P(\mathbf{A}|\mathbf{E}, \mathbf{F}, \mathbf{T}^*)$, such that:

$$T^* = \arg\max_T P(\mathbf{F}|\mathbf{E}, \mathbf{T}) \quad (1)$$

$$= \arg\max_T \sum_{\mathbf{A}} P(\mathbf{F}, \mathbf{A}|\mathbf{E}, \mathbf{T}). \quad (2)$$

On the other hand, the Bayesian solution returns the posterior $P(\mathbf{A}|\mathbf{E}, \mathbf{F})$, which is obtained from:

$$P(\mathbf{F}, \mathbf{A}|\mathbf{E}) = \int_{\mathbf{T}} P(\mathbf{T}) P(\mathbf{F}, \mathbf{A}|\mathbf{E}, \mathbf{T}). \quad (3)$$

2.1. EM

We used the GIZA++ [2] software to obtain the EM-estimated IBM Model 4 alignments. The default bootstrapping regimen was used, i.e., 5 iterations each of IBM Model 1 and HMM, followed by 3 iterations each of Models 3 and 4, in that order.

2.2. Gibbs sampling

It was shown in [5] that, compared to EM, Bayesian word alignment using Gibbs sampling (GS) reduces overfitting (e.g., high-fertility rare words), induces smaller models, and improves the BLEU score. In our system, we obtained two GS-inferred alignments; one for IBM Model 1 [5] and one for IBM Model 2 [7]. The following settings were common to both samplers:

- *Initialization:* The samplers were initialized with the EM-estimated Model 4 alignments obtained in 2.1.
- *Hyperparameters:* A sparse prior $P(\mathbf{T})$ was imposed on the translation parameters, specifically, a symmetric Dirichlet distribution with $\theta = 0.0001$.

- *Sample collection:* A total of 200 iterations of the sampler was run, with only the last 100 iterations used for Viterbi estimation (i.e., the burn-in period was 100 iterations).

For Bayesian Model 2, we used a uniform prior on the distortion parameters, specifically, a symmetric Dirichlet distribution with $\theta = 1$. We used relative distortion [8] for Model 2 in order to reduce the number of parameters.

2.3. Variational Bayes

Variational Bayes (VB) is a Bayesian inference method sometimes preferred over GS due to its relatively lower computational cost and scalability. However, VB inference approximates the model by assuming independence between the hidden variables and the parameters. Word alignment using Dirichlet priors and VB inference was investigated in [9, 10]. In our experiments, we used the publicly available software[1]. VB training was used in all models of the bootstrapping regimen for training IBM Model 4. As done in [9, 10], we set the Dirichlet hyperparameter $\theta = 0$ (the default setting) and ran 5 iterations of VB for each of IBM Model 1, HMM, Model 3 and Model 4[2].

2.4. Alignment Combination

We used the four different alignment methods explained above (EM with Model 4, GS with Models 1 and 2, and VB with Model 4) and combined the phrases extracted from before extracting phrases and estimating the phrase table probabilities. Our alignment combination method is similar to those previously used by others, e.g., [11]. The only change to the standard Moses training procedure is that we 4-fold replicated the training corpus, ran a different alignment method on each replica, and concatenated the obtained individual alignments. Alignments in each direction were further combined (symmetrized) using the default heuristic in Moses (grow-diag-final-and).

3. Data

Tables 1 and 2 present the main characteristics of the parallel corpora used in our experiments for translation model training. For the Arabic-English task, we utilized only the TED parallel corpus [12], while for the Turkish-English task, we utilized both the TED and SE Times parallel corpora.

We trained three separate language models from the English sides of the following parallel corpora (Table 3): the TED corpus (ted), the News Commentary corpus (nc), and the Gigaword French-English corpus (gigafren). The combination weights of these language models were optimized during the tuning step, together with the other log-linear model features.

[1] http://cs.rochester.edu/∼gildea/mt/giza-vb.tgz

[2] This is achieved by specifying the following options in the Moses training: model1tvb=1,modelhmmtvb=1,model3tvb=1,model4tvb=1.

Table 1: *Statistics of the parallel training data used in the Arabic-English experiments.*

Translation Model	Arabic	English
Sentences	136,729	
Tokens (M)	2.5	2.6
Types (k)	68.5	51.3
Singletons (k)	28.7	21.5

Table 2: *Statistics of the parallel training data used in the Turkish-English experiments.*

	TED		SETimes	
	Turkish	English	Turkish	English
Sentences	124,193		161,408	
Tokens (M)	1.8	2.4	3.9	4.4
Types (k)	153.9	47.3	135.9	66.6
Singletons (k)	87.6	19.6	66.2	29.8

Among the available development corpora, we used dev2010 for tuning and tst2010 for internal testing. We also present the experimental results for the tst2011 dataset, which was made available to the participants after the submission period.

Table 3: *Statistics of the language model training data.*

	ted	nc	gigafren
Tokens (M)	2.8	5.1	672
Unigrams (k)	53	69	2000

4. Common system features

Our submissions for both language pairs feature phrase-based statistical machine translation systems trained using the Moses toolkit [13]. Truecasing models were trained on tokenized training data, and subsequently all models were trained on truecased data. All language models were standard 4-gram models trained with modified Kneser-Ney discounting and interpolation using the SRILM toolkit [14]. The minimum error rate training (MERT) algorithm [15] with lattice sampling [16] and search in random directions [17] was used with BLEU [18] as the metric to be optimized. Evaluation was also performed using BLEU.

5. Arabic-English

5.1. Preprocessing

Arabic data was morphologically decomposed using MADA+TOKAN [19] with BAMA 2.0 (LDC2004L02) [20] and the default tokenization scheme. For English, the default tokenizer in the Moses package was used together with some post-processing. The final tokenization convention can be summarized as follows:

- Map unicode punctuation marks to ASCII.
- Merge and standardize consecutive hyphens and dots.
- Separate hyphens only if both sides are numbers (default in MADA+TOKAN).
- Merge back separated apostrophes.

Moreover, in order to reduce data sparsity in word alignment, all numbers were reduced to their last digits during training. For example, the tokens "60,000" and "2,000" were both replaced with "0".

5.2. Experiments

Table 4 compares the translation performance of the various alignment methods discussed in Section 2. For IBM Models 1 and 2, both Bayesian approaches (VB and GS) outperform EM. However, for Model 4, EM turned out to be better than VB[3]. The alignment combination described in Section 2.4 (last row in Table 4) did not provide the expected improvement, yielding a BLEU score somewhere between the highest and the lowest of the combined individual BLEU scores. Nevertheless, we chose it as our official submission for the Arabic-English track.

Table 4: *Performance of alignment inference schemes and their combination in the Arabic-English experiments.*

	Alignment		BLEU		
	Method	Model	dev10	tst10	tst11
1	EM	1	24.11	22.68	22.34
2	VB	1	24.34	23.21	**22.95**
3	GS	1	**24.59**	**23.22**	22.68
4	EM	2	24.33	22.65	22.37
5	VB	2	25.01	23.64	23.19
6	GS	2	**25.34**	**23.80**	**23.50**
7	EM	4	**25.48**	**23.83**	**23.93**
8	VB	4	25.09	23.71	23.28
9	(3)+(6)+(7)+(8)		25.01	23.58	23.13

Reducing model size was previously proposed as an objective in unsupervised word alignment, e.g., in [21, 22]. To see whether the Bayesian methods indeed achieve smaller models, we analyzed the outputs of each alignment method in terms of the total number of unique word translations in the produced alignments. Table 5 shows that both Bayesian methods induce significantly smaller alignment dictionaries than EM.

A contributing factor for the high dictionary size in ML-estimated alignments is that the rare source words in the training corpus are aligned to excessively many target words, also known as "garbage collection" [3]. To measure the effect of this phenomenon, the average fertility of singletons ($\tilde{\phi}_{sing}$) was used in [23] and [22]. We present $\tilde{\phi}_{sing}$ values in both alignment directions for the different alignment methods in Tables 6 and 7. We see that both Bayesian methods

[3]A Model-4 implementation of GS is not yet available

Table 5: *Number of distinct word translations (unique alignment pairs) induced by the alignment methods in the Arabic-English experiments.*

	Alignment		Dictionary Size (k)		
	Method	Model	en-ar	ar-en	sym.
1	EM	1	508	528	412
2	VB	1	182	187	258
3	GS	1	282	318	321
4	EM	2	558	548	659
5	VB	2	195	199	281
6	GS	2	289	317	395
7	EM	4	496	487	546
8	VB	4	207	218	292
9	(3)+(6)+(7)+(8)		743	771	821

dramatically reduce the average alignment fertility of singletons.

However, $\tilde{\phi}_{sing}$ can sometimes be misleading because a smaller value is not necessarily better. For example, the lowest possible value 0 can be trivially achieved by leaving all singletons unaligned, which is clearly not desirable. Tables 6 and 7 also show the ratio of unaligned singletons ($|sing0|/|sing|$)[4], which reveals that VB for Model 1 leaves nearly half of the singletons unaligned. The rightmost column in the table presents $\tilde{\phi}_{sing+}$, which averages the fertilities only over aligned singletons and has the minimum attainable value of 1.

6. Turkish-English

6.1. Preprocessing

For both languages, the default tokenizer in the Moses package was used, without any morphological processing.

6.2. Experiments

Our first system used a single phrase-table trained on the combined TED+SETimes corpus and used only VB (2.3) as the alignment inference method. Our second system used four different alignment methods as in our Arabic-English submission (Section 5), separately for each of the TED and the SETimes corpora, and then used the resulting two phrase tables in decoding. However, due to a bug at the time of the submission, the internal BLEU scores of this second system were significantly lower than our first system. Therefore, we submitted the first system as our primary submission.

Table 8 compares the BLEU scores of different alignment methods on the Turkish-English TED corpus. As opposed to the Arabic-English case, we observe in Table 8 that alignment combination provides a significant gain over the individual alignments.

[4]We further denote the aligned singletons by "$sing+$" so that $|sing| = |sing0| + |sing+|$.

Table 6: *Singleton alignment performance (en-ar) of the alignment methods in the Arabic-English experiments.*

Method	Model	$\tilde{\phi}_{sing}$	$\|sing0\|/\|sing\|$	$\tilde{\phi}_{sing+}$
EM	1	5.0	0.20	6.2
VB	1	0.8	0.47	1.6
GS	1	1.2	0.26	1.6
EM	2	3.7	0.001	3.7
VB	2	0.9	0.27	1.3
GS	2	1.1	0.23	1.4
EM	4	4.1	0.001	4.1
VB	4	1.3	0.08	1.5

Table 7: *Singleton alignment performance (ar-en) of the alignment methods in the Arabic-English experiments.*

Method	Model	$\tilde{\phi}_{sing}$	$\|sing0\|/\|sing\|$	$\tilde{\phi}_{sing+}$
EM	1	6.0	0.20	7.4
VB	1	0.9	0.46	1.6
GS	1	1.6	0.17	1.9
EM	2	4.4	0.001	4.4
VB	2	1.1	0.23	1.4
GS	2	1.4	0.16	1.7
EM	4	4.8	0.002	4.8
VB	4	1.4	0.08	1.5

7. Conclusion and Future Work

We described our submission to IWSLT 2012. The main innovation tested was using Bayesian word alignment methods (both variational Bayes and Gibbs sampling) in combination with the standard EM. As future work, we plan to apply the same technique on the MultiUN corpus for the Arabic-English task, and other larger corpora for other language pairs.

8. References

[1] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. of the International Workshop on Spoken Language Translation*, Hong Kong, HK, December 2012.

[2] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[3] R. C. Moore, "Improving IBM word alignment Model 1," in *Proc. ACL*, Barcelona, Spain, July 2004, pp. 518–525.

[4] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, M. J. Goldsmith, J. Hajic, R. L. Mercer, and S. Mohanty, "But dictionaries are data too," in *Proc. HLT*, Plainsboro, New Jersey, 1993, pp. 202–205.

Table 8: *Performance of alignment inference schemes and their combination in the TED Turkish-English experiments.*

	Alignment		BLEU	
	Method	Model	dev10	tst10
1	EM	1	10.68	11.43
2	VB	1	**10.80**	11.87
3	GS	1	10.61	**12.10**
4	EM	2	10.21	11.67
5	VB	2	**11.16**	**11.92**
6	GS	2	10.68	11.47
7	EM	4	10.28	11.28
8	VB	4	**10.38**	**11.33**
9	(3)+(6)+(7)+(8)		**11.78**	**12.90**

[5] C. Mermer and M. Saraclar, "Bayesian word alignment for statistical machine translation," in *Proc. ACL-HLT: Short Papers*, Portland, Oregon, June 2011, pp. 182–187.

[6] P. F. Brown, V. J. Della Pietra, S. A. Della Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[7] C. Mermer, M. Saraclar, and R. Sarikaya, "Improving statistical machine translation using Bayesian word alignment and Gibbs sampling," *IEEE Transactions on Audio, Speech and Language Processing (in review)*, 2012.

[8] S. Vogel, H. Ney, and C. Tillmann, "HMM-based word alignment in statistical translation," in *Proc. COLING*, 1996, pp. 836–841.

[9] D. Riley and D. Gildea, "Improving the performance of GIZA++ using variational Bayes," The University of Rochester, Computer Science Department, Tech. Rep. 963, December 2010.

[10] ——, "Improving the IBM alignment models using variational Bayes," in *Proc. ACL: Short Papers*, 2012, pp. 306–310.

[11] W. Shen, B. Delaney, T. Anderson, and R. Slyh, "The MIT-LL/AFRL IWSLT-2007 MT system," in *Proc. IWSLT*, Trento, Italy, 2007.

[12] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[13] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin,

and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proc. ACL: Demo and Poster Sessions*, Prague, Czech Republic, June 2007, pp. 177–180.

[14] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. ICSLP*, vol. 3, 2002.

[15] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. ACL*, Sapporo, Japan, July 2003, pp. 160–167.

[16] S. Chatterjee and N. Cancedda, "Minimum error rate training by sampling the translation lattice," in *Proc. EMNLP*, 2010, pp. 606–615.

[17] D. Cer, D. Jurafsky, and C. D. Manning, "Regularization and search for minimum error rate training," in *Proc. WMT*, 2008, pp. 26–34.

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. ACL*, Philadelphia, Pennsylvania, July 2002, pp. 311–318.

[19] O. R. Nizar Habash and R. Roth, "MADA+TOKAN: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization," in *Proc. Second International Conference on Arabic Language Resources and Tools*, 2009.

[20] T. Buckwalter, "Buckwalter Arabic morphological analyzer version 2.0," *Linguistic Data Consortium*, 2004.

[21] T. Bodrumlu, K. Knight, and S. Ravi, "A new objective function for word alignment," in *Proc. NAACL-HLT Wk. Integer Linear Programming for Natural Language Processing*, Boulder, Colorado, June 2009, pp. 28–35.

[22] A. Vaswani, L. Huang, and D. Chiang, "Smaller alignment models for better translations: Unsupervised word alignment with the l0-norm," in *Proc. ACL*, 2012, pp. 311–319.

[23] C. Dyer, J. H. Clark, A. Lavie, and N. A. Smith, "Unsupervised word alignment with arbitrary features," in *Proc. ACL:HLT*, Portland, Oregon, June 2011, pp. 409–419.

Technical Papers

Active Error Detection and Resolution for Speech-to-Speech Translation

Rohit Prasad, Rohit Kumar, Sankaranarayanan Ananthakrishnan, Wei Chen, Sanjika Hewavitharana, Matthew Roy, Frederick Choi, Aaron Challenner, Enoch Kan, Arvind Neelakantan, Prem Natarajan

Speech, Language, and Multimedia Business Unit, Raytheon BBN Technologies
Cambridge MA, USA
{rprasad,rkumar,sanantha,wchen,shewavit,mroy,fchoi,achallen,ekan,aneelaka,prem}@bbn.com

Abstract

We describe a novel two-way speech-to-speech (S2S) translation system that *actively* detects a wide variety of common error types and resolves them through *user-friendly* dialog with the user(s). We present algorithms for detecting out-of-vocabulary (OOV) named entities and terms, sense ambiguities, homophones, idioms, ill-formed input, etc. and discuss novel, interactive strategies for recovering from such errors. We also describe our approach for prioritizing different error types and an extensible architecture for implementing these decisions. We demonstrate the efficacy of our system by presenting analysis on live interactions in the English-to-Iraqi Arabic direction that are designed to invoke different error types for spoken language translation. Our analysis shows that the system can successfully resolve 47% of the errors, resulting in a dramatic improvement in the transfer of problematic concepts.

1. Introduction

Great strides have been made in Speech-to-Speech (S2S) translation systems that facilitate cross-lingual spoken communication [1][2][3]. While these systems [3][4][5] already fulfill an important role, their widespread adoption requires broad domain coverage and unrestricted dialog capability. To achieve this, S2S systems need to be transformed from *passive conduits* of information to *active participants* in cross-lingual dialogs by detecting key causes of communication failures and recovering from them in a user-friendly manner. Such an active participation by the system will not only maximize translation success, but also improve the user's perception of the system.

The bulk of research exploring S2S systems has focused on maximizing the performance of the constituent automatic speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) components in order to improve the rate of success of cross-lingual information transfer. There have also been several attempts at joint optimization of ASR and MT, as well as MT and TTS [6][7][8]. Comparatively little effort has been invested in the exploration of approaches that attempt to detect errors made by these components, and the interactive resolution of these errors with the goal of improving translation / concept transfer accuracy.

Disclaimer: This paper is based upon work supported by the DARPA BOLT Program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Distribution Statement A (Approved for Public Release, Distribution Unlimited)

Our previous work presented a novel methodology for assessing the severity of various types of errors in our English/Iraqi S2S system [9]. These error types can be broadly categorized into: (1) out-of-vocabulary concepts; (2) sense ambiguities due to homographs, and (3) ASR errors caused by mispronunciations, homophones, etc. Several approaches, including implicit confirmation of ASR output with barge-in and back-translation [10], have been explored for preventing such errors from causing communication failures or stalling the conversation. However, these approaches put the entire burden of error detection, localization, and recovery on the user. In fact, the user is required to infer the potential cause of the error and determine an alternate way to convey the same concept – clearly impractical for the broad population of users.

To address the critical limitation of S2S systems described above, we present novel techniques for: (1) automatically detecting potential error types, (2) localizing the error span(s) in spoken input, and (3) interactively resolving errors by engaging in a clarification dialog with the user. Our system is capable of detecting a variety of error types that impact S2S systems, including out-of-vocabulary (OOV) named entities and terms, word sense ambiguities, homophones, mispronunciations, incomplete input, and idioms.

Another contribution of this paper is the novel strategies for overcoming these errors. For example, we describe an innovative approach for cross-lingual transfer of OOV named entities (NE) by splicing corresponding audio segments from the input utterance into the translation output. For handling word sense ambiguities, we propose a novel constrained MT decoding technique that accounts for the user's intended sense based on the outcome of the clarification dialog.

A key consideration for making the system an active participant is deciding how much the system should talk, i.e. the number of clarification turns allowed to resolve potential errors. With that consideration, we present an effective strategy for prioritizing the different error types for resolution and also describe a flexible architecture for storing, prioritizing, and resolving these error types.

2. Error Types Impacting S2S Translation

We focus on seven types of errors that are known to impact S2S translation. Table 1 shows an example of each of these error types. Out-of-vocabulary names (*OOV-Name*) and Out-of-vocabulary non-name words (*OOV-Word*) are some of the errors introduced by the ASR in S2S systems. OOV words are recognized as phonetically similar words that do not convey the intended concept. *Word sense ambiguities* in the input language can cause errors in translation if a target word/phrase does not correspond to the user's intended sense.

Homophone ambiguities and *mispronunciations* are two other common sources of ASR error that impact translation. *Incomplete utterances* are typically produced if the speaker abruptly stops speaking or due to a false-release of the push-to-talk microphone button. Finally, unseen *idioms* often produce erroneous literal translations due of the lack of appropriate transfer rules in the MT parallel training data.

Table 1: Examples of Types of Errors

Error Type	Example
OOV-Name	**My name is Sergeant Gonzales.** *ASR*: my name is sergeant guns all us
OOV-Word	**The utility prices are extortionate.** *ASR*: the utility prices are extort unit
Word Sense	**Does the town have enough tanks.** *Ambiguity:* armored vehicle | storage unit
Homophone	**Many souls are in need of repair.** *Valid Homophones*: soles, souls
Mispron.	**How many people have been harmed by the water when they wash.** *ASR*: how many people have been harmed by the water when they worse
Incomplete	**Can you tell me what these**
Idiom	**We will go the whole nine yards to help.** *Idiom*: the whole nine yards

3. Approach for Active Error Detection and Resolution

Figure 1 shows the architecture of our two-way English to Iraqi-Arabic S2S translation system. In the English to Iraqi direction, the initial English ASR hypothesis and its corresponding translation are analyzed by a suite of *error detection* modules discussed in detail in Section 3.3. An *Inference Bridge* data structure supports storage of these analyses in an interconnected and retraceable manner. The potential classes of errors and their associated spans in the input are identified and ranked in an order of severity using this data structure. A resolution strategy, discussed in detail in Section 3.4, is executed based on the top ranked error.

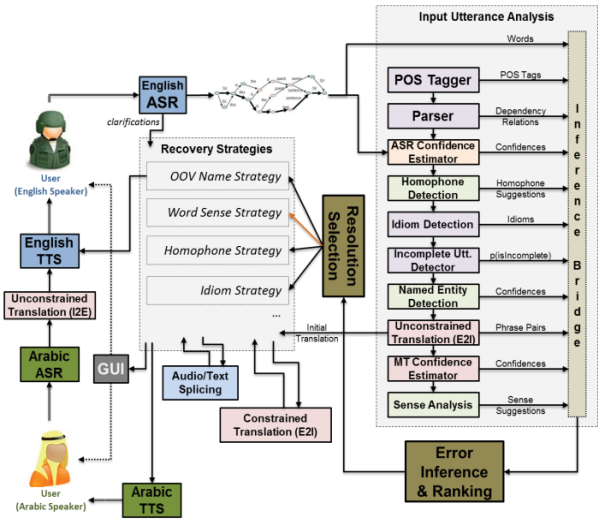


Figure 1: BBN English/Iraqi-Arabic S2S System with Error Recovery in English to Iraqi-Arabic direction

The strategies use a combination of automated and user-mediated interventions to attempt recovery of the concepts associated with the error span. At the end of a strategy, the Arabic speaker may be presented with a translation of the user's input utterance with appropriate corrections; or the English speaker may be informed of the system's inability to translate the sentence along with an explanation of the cause of this failure. With this information, the English speaker can choose to rephrase the input utterance so as to avoid the potential failure. At all times, the English speaker has the option to force the system to proceed with its current translation by issuing the "Go Ahead" command. Our system may be regarded as *high-precision* due to its ability to prevent the transfer of erroneously translated concepts to Arabic speakers. This increased precision comes at the cost of increased effort by the English speaker in terms of performing clarifications and rephrasals. The metrics and results presented in Section 4 study this compromise.

The Arabic to English direction of the system implements a traditional loosely coupled pipeline architecture comprising of the Arabic ASR, Arabic-English MT, and English TTS.

3.1. Baseline ASR System

Speech recognition was based on the BBN Byblos ASR system. The system uses a multi-pass decoding strategy in which models of increasing complexity are used in successive passes in order to refine the recognition hypotheses [11]. In addition to the 1-best and N-best hypotheses, our ASR engine generates word lattices and confusion networks with word posterior probabilities. The latter are used as confidence scores for a variety of error detection components.

The acoustic model was trained on approximately 150 hours of transcribed English speech from the DARPA TRANSTAC corpus. The language model (LM) was trained on 5.8M English sentences (60M words), drawn from both in-domain and out-of-domain sources. LM and decoding parameters were tuned on a held-out development set of 3,534 utterances (45k words). With a dictionary of 38k words, we obtained 11% WER on a held-out test set of 3k utterances.

3.2. Baseline MT System

Our statistical machine translation (SMT) system was trained using a corpus derived from the DARPA TRANSTAC English-Iraqi parallel two-way spoken dialogue collection. The parallel data (773k sentence pairs, 7.3M words) span a variety of scenarios including force protection, medical diagnosis and aid, maintenance and infrastructure, etc.

Table 2: SMT performance for different configurations

System	BLEU	100-TER
Baseline	**16.1**	**35.8**
Boosted	**16.0**	**36.3**
PAC	**16.1**	**36.0**

Phrase translation rules were extracted from bidirectional IBM Model 4 word alignment [12] based on the heuristic approach of [13]. The target LM was trained on Iraqi transcriptions from the parallel corpus and the log-linear model tuned with MERT [14] on a held-out development set (~44.7k words). Table 2 summarizes translation performance on a held-out test set (~38.5k words) of the baseline English

to Iraqi SMT system for vanilla phrase-based, boosted alignment [15], and phrase alignment confidence (PAC) [16] systems. We used the PAC SMT models in our system.

3.3. Input Analysis & Error Detection

3.3.1. Automatic Identification of Translation Errors

In order to automatically detect mistranslated segments of the input, we built a confidence estimation system for SMT (similar to [17]) that learns to predict the probability of error for each hypothesized target word. In conjunction with SMT phrase derivations, these confidence scores can be used to identify input segments that may need to be clarified. The confidence estimator relies on a variety of feature classes:

- *SMT-derived features* include forward and backward phrase translation probability, lexical smoothing probability, target language model probability, etc.
- *Bilingual indicator features* capture word co-occurrences in the generating source phrase and the current target word and are obtained from SMT phrase derivations.
- *Source perplexity* is positively correlated with translation error. We used the average source phrase perplexity as a feature in predicting probability of translation error.
- *Word posterior probability* was computed for each target word in the 1-best hypothesis based on weighted majority voting over SMT-generated *N*-best lists.

Reference labels for target words (*correct* vs. *incorrect*) were obtained through automated TER alignment on held-out partitions of the training set (10-fold jack-knifing). The mapping between above features and reference labels was learned with a maximum-entropy (MaxEnt) model. We also exploited the "bursty" nature of SMT errors by using a joint lexicalized label (n-gram) LM to rescore confusion networks generated by the pointwise MaxEnt predictor. Table 3 summarizes the prediction accuracy of correct and incorrect hypothesized Iraqi words on the MT test set (~38.5k words).

Table 3: Incorrect target word classification performance

Method	Dev set	Test set
Majority (baseline)	51.6%	52.6%
MaxEnt + Lexicalized LM	70.6%	71.1%

3.3.2. OOV Named Entity Detection

Detecting OOV names is difficult because of the unreliable features resulting from tokens misrecognized by ASR in the context of an OOV word. We use a MaxEnt model to identify OOV named-entities (NE) in user input [18]. Our model uses lexical and syntactic features to compute the probability of each input word being a name. We trained this model on Gigaword, Wall Street Journal (WSJ), and TRANSTAC corpora consisting of approximately 250K utterances (4.8M words). This includes 450K occurrences of 35K unique named-entity tokens. On a held-out clean (i.e. no ASR error) test set consisting of only OOV named-entities, this model detects 75.4% named-entities with 2% false alarms.

While the above detector is trained on clean text, our real test cases are noisy due to ASR errors in the region of the OOV name. To address this mismatch, we use word posteriors from ASR in two ways. First, an early fusion technique weighs each feature with the word posterior associated with the word from which the feature is derived. This attenuates unreliable features at runtime. Second, we use a heuristically-determined linear combination of ASR word posteriors and the MaxEnt named-entity posterior to compute a score for each word. This technique helps in further differentiating OOV named-entity words since the ASR word posterior term serves as a strong OOV indicator.

Contiguous words with NE posteriors greater than a specified threshold are considered as candidate OOV names. These spans are filtered through a list of known NEs. If a sizeable span (>0.33 seconds) contains at least one non-stopword unknown name token, it is considered for OOV name resolution.

We evaluated our OOV NE detector on an offline set comprising of 2,800 utterances similar in content to the evaluation scenarios described in Section 4.1. We are able to detect 40.5% OOV NEs with 39.1% precision. Furthermore, an additional 19.9% OOV NEs were identified as error spans using the detector described in the next section.

3.3.3. Error Span Detection

We use a heuristically derived linear combination of ASR and MT confidence for each input word in the source language to identify source words that are likely to result in poor translations. We use this error detector to identify a variety of errors including unknown/unseen translation phrases, OOV Word (non-names), user mispronunciations and ASR errors. All consecutive words (ignoring stop words) identified by this detector are concatenated into a single span.

3.3.4. Improving Translation of Multiple Word Senses

Phrase-based SMT is susceptible to word sense translation errors because it constructs hypotheses based on translation rules with relatively limited context. We address this issue through a combination of (a) constrained SMT decoding driven by sense-specific phrase pair partitions obtained using a novel semi-supervised clustering mechanism, and (b) a supervised classifier-based word sense predictor.

3.3.4.1 Semi-supervised phrase pair clustering

The use of constraints for clustering phrase pairs associated with a given ambiguity class into their senses significantly reduces clustering noise and "bleed" across senses due to lack of sufficient context in the phrase pairs. Constraints are obtained in three different ways.

1. *Key-phrase constraints*: Manually annotated key-phrases are used to establish an initial set of constraints between each pair of translation rules corresponding to a given ambiguity class. Two phrase pairs are related by a *must-link* constraint if their source phrases both contain key-phrases associated with the same sense label; or by a *cannot-link* constraint if they contain key-phrases corresponding to different sense labels.
2. *Instance-based constraints*: The word alignment of a sentence pair often allows extraction of multiple phrase pairs spanning the same ambiguous source word. All of these phrase pairs refer to the same sense of the ambiguous word and must be placed in the same partition. We enforce this by establishing *must-link* constraints between them.
3. *Transitive closure*: The process of *transitive closure*

ensures that the initial set of constraints is propagated across all two-tuples of phrase pairs. This leads to a set of constraints that is far larger than the initial set, leading to well-formed, noise-free clusters. We implemented transitive closure as a modified version of the Floyd-Warshall algorithm. We used the transitive closure over key-phrase and instance-based constraints to partition phrase pairs for a given ambiguity class into their respective senses using constrained *k*-means [19].

3.3.4.2 Constrained SMT decoding

Constrained decoding is a form of dynamic pruning of the hypothesis search space where the source phrase spans an ambiguous word. The decoder must then choose a translation from the partition corresponding to the intended sense. We used the partitioned inventories to tag each phrase pair in the SMT phrase table with its ambiguity class and sense identity.

At run time, the constrained SMT decoder expects each input word in the test sentence to be tagged with its ambiguity class and intended sense identity. Unambiguous words are tagged with a generic class and sense identity. When constructing the search graph over spans with ambiguous words tagged, we ensure that phrase pairs covering such spans match the input sense identity. Thus, the search space is constrained only in the regions of non-generic ambiguity classes, and unconstrained elsewhere. By naturally integrating word sense information within the translation model, we preserve the intended sense and generate fluent translations.

Table 4: Concept transfer for ambiguous words

Method	**Yes**	**No**	**unk**
Unconstrained	95	68	1
Constrained	108	22	34
Improvement	**13.7%**	**66.2%**	**n/a**

We evaluated the constrained decoder on a balanced offline test set of 164 English sentences covering all in-vocabulary senses of 73 ambiguity classes that appeared in multiple senses in our training data. Each test sentence contains exactly one ambiguous word. We presented each input sentence and its translation to a bilingual judge, with the ambiguous source word and the target word(s) due to it both highlighted. The judge passes a binary judgment; *yes*, implying that the sense of the source word is preserved, or *no*, indicating an incorrect sense substitution. Non-dominant senses of an ambiguity class may not be translatable if the corresponding partition does not possess sufficient contextual coverage. We count the number of untranslatable ambiguous source concepts separately from correct or incorrect sense transfer. Table 4 summarizes these results.

3.3.4.3 Supervised word sense disambiguation

Complementary to the above framework is a supervised word sense disambiguation system that uses MaxEnt classification to predict the sense of an ambiguous word. Sense predictions by this component are integrated with user input in our mixed-initiative interactive system to identify the appropriate phrase pair partitions for constrained decoding.

We selected up to 250 representative sentences for each ambiguity class from the training corpus and had human annotators (a) assign an identity and description for up to five different senses, and (b) label each instance with the appropriate sense identity. Based on these annotations, we trained separate maximum entropy classifiers for each ambiguity class, with sense identities as target labels. Classifiers were trained for 110 ambiguity classes using *contextual* (window-based), *dependency* (parent/child of ambiguous word), and corresponding *part-of-speech* features.

We performed an offline evaluation of the sense classifiers by using them to predict the sense of the ambiguity classes in held out test sentences. The most frequent sense of an ambiguity class in the training data served as a baseline (chance level) for that class. The baseline word sense predication accuracy rate over 110 ambiguity classes covering 2,324 sentences containing ambiguous words was 73.7%. This improved to 88.1% using the MaxEnt sense classifiers.

3.3.5. Homophone Detection and Correction

A common problem with ASR is the substitution of a different word that sounds identical to the spoken word (e.g. “role” vs. “roll”). To alleviate this problem, we developed a state-of-the-art automatic homophone detection and correction module based on MaxEnt classification. We induced a set of homophone classes from the ASR lexicon such that the words in each class had identical phonetic pronunciation. For each homophone class, we identified training examples containing the constituent words. A separate classifier was trained for each homophone class with the correct variants as the target labels. This component essentially functions as a strong, local, discriminative language model. The features used for the homophone corrector are identical to those used for supervised word sense disambiguation (Section 3.3.4.3).

We evaluated this component by simulating, on a held-out test set for each homophone class, 100% ASR error by randomly substituting a different variant for each homophone constituent in these sentences. We then used the classifier to predict the word variant for any slot corresponding to a homophone class constituent. The overall correction rate over 223 homophone classes covering 174.6k test sentences containing homophone classes was 95.8%. Similarly, the false correction rate (simulated by retaining the correct homophone variant in the test set) was determined to be 1.3%.

3.3.6. Idiom Detection

Idioms unseen in SMT training usually generate incomprehensible literal translations. To detect and pre-empt translation errors originated from idioms, we harvested a large list of English idioms from public domain sources to use in a simple string matching front-end. However, the harvested idioms are usually in a single canonical form, e.g. “give *him* a piece of my mind”. Thus, simple string match would not catch the idiom “give *her* a piece of my mind”. We used two approaches to expand coverage of the idiom detector.

1. *Rule-based idiom expansion:* We created rules for pronoun expansion (e.g. “his” → “her”, “their”, etc.) and verb expansion (e.g. “give her a piece of my mind” → “gave her a piece of my mind”), being conservative to avoid explosion and creation of nonsense variants.
2. *Statistical idiom detector:* We trained a binary MaxEnt classifier that predicts whether any input n-gram is an idiom. We used 3.2k gold standard canonical idioms as positive samples and all 15M non-idiom n-grams in our data as negative samples. On a balanced set containing

unseen idiom variants and non-idioms, this classifier gave us a detection rate of 33.2% at 1.8% false alarm.

3.3.7. *Incomplete Utterance Detection*

In order to detect user errors such as intentional aborts after mis-speaking, or unintentional pushing or releasing of the "record" button, we built an incomplete utterance detector (based on a MaxEnt classifier) that identifies fragments with ungrammatical structure in recognized transcriptions. Training data for incomplete utterances were automatically generated using an error simulator that randomly removed words from the beginning and/or end of a clean, fully-formed sentence. A number of lexical and syntactic features were used to train and evaluate the incomplete utterance classifier.

We trained a binary classifier on approximately 771k fully formed sentences and varied the number of automatically generated incomplete utterances. We evaluated the classifier on a balanced test set of 1,000 sentences with 516 auto-generated sentences that were verified by hand to be positive examples of incomplete sentences. At a false alarm rate of 5%, the incomplete utterance detector demonstrated a detection rate of 41%. Syntactic and part-of-speech features were particularly powerful at identifying this error type.

3.4. Error Resolution Strategies

Our implementation of error resolution strategies follows a multi-expert architecture along the lines of Jaspis [20] and Rime [21]. Each strategy has been manually designed to resolve one or more types of errors discussed in Section 2.

Figure 2 illustrates 9 interaction strategies used by our system. Each strategy is comprised of a sequence of steps which include actions such as TTS output, user input processing, translation (unconstrained or constrained) and other error type specific operations.

The *OOV Name* and *ASR Error* strategies are designed to interactively resolve errors associated with OOV entities (names and non-names), ASR errors and MT errors. When a span of words is identified as an OOV named-entity, the user is asked to confirm whether the audio segment spanning those words actually corresponds to a name (Excerpt A), following which the segment is spliced in place of the target phrases corresponding to that span. In the case where a (non-name) error span is detected by the detector described in Section 3.3.3, the user is asked to rephrase the utterance. This strategy

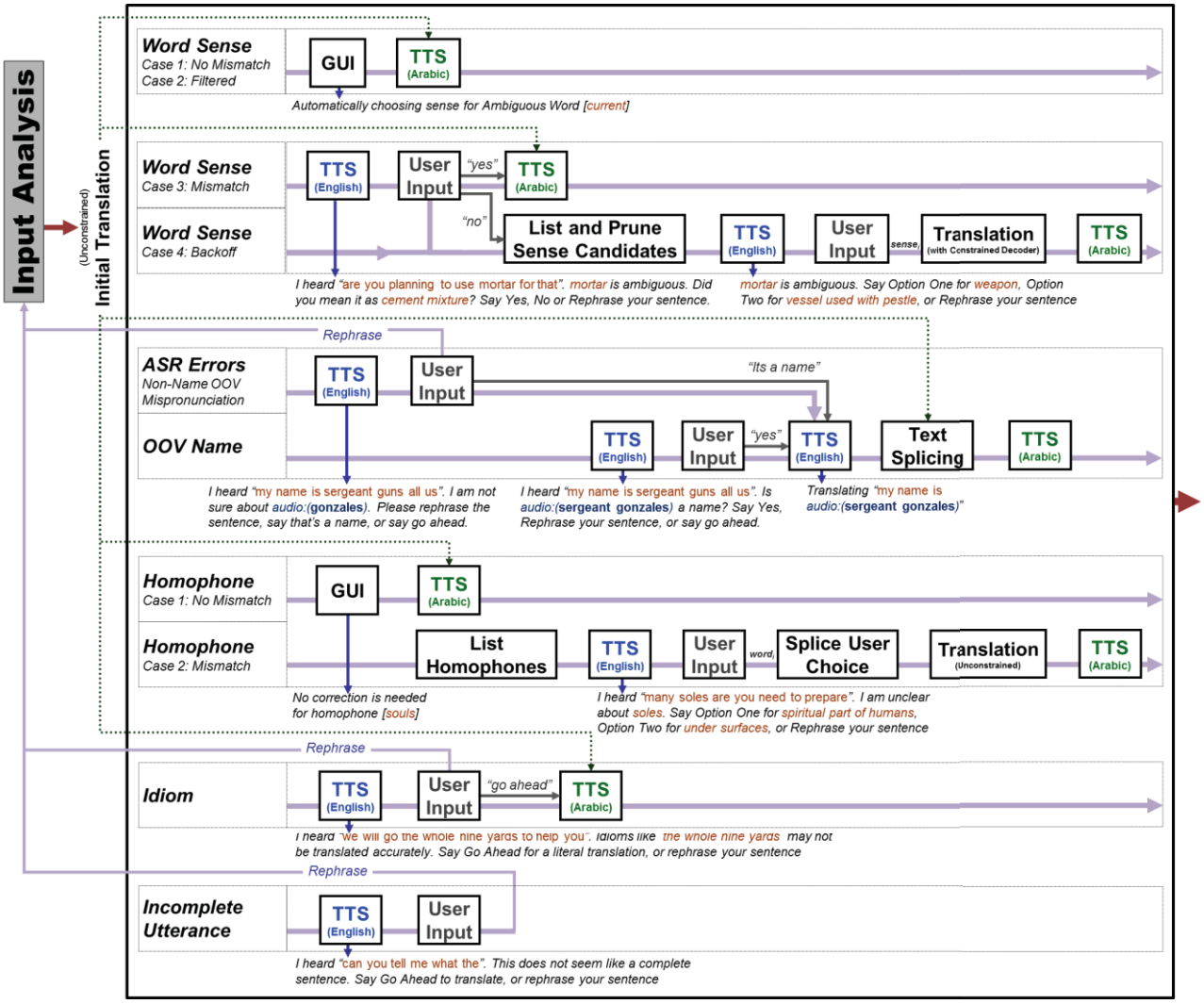


Figure 2. Interaction Strategies for Error Resolution

Table 5: Examples of Error Recovery Strategies

A. Out of Vocabulary Name *User Utt.:* My name is Sergeant Gonzalez ASR: **my name is sergeant guns all us** Sys.: I heard "my name is sergeant guns all us". Is (audio:Sergeant Gonzalez) a name? Say Yes or Rephrase your sentence, or Go Ahead. ASR: **yes** Sys.: Translating "my name is (audio: Sergeant Gonzalez)".
B. OOVs and other ASR/MT error *User Utt.:* Are your wages enough to subsist on? ASR: **are your wages enough to sub system** Sys.: I heard "are your wages enough to sub system". I am not sure about (audio: subsist). Please rephrase the sentence, say that's a name, or say Go Ahead. User: **do you make enough money** Sys.: Translating "do you make enough money"
C. Word Sense *User Utt.:* I am looking for the man in charge. ASR: **i am looking for the man in charge** Sys.: I heard "i am looking for the man in charge". Charge is ambiguous. Say Option One for in command of, Option Two for payment, Option Three for accusation, or Rephrase your sentence. User: **option one** Sys.: Translating "i am looking for the man in charge" using charge as in command of.
D. Homophone *User Utt.*: How much do you pay for the fare? ASR: **how much do you pay for the fair** Sys.: I heard "how much do you pay for the fair". I am unclear about fair. Say Option One for price of transportation, Option Two for free from bias, Or Rephrase your sentence ASR: **option one** Sys.: Translating "how much do you pay for the fare"
E. Improper Input/Fragment *User Utt.:* I am going for the- ASR: **i am going for the** Sys.: I heard "i am going for the". This does not seem like a complete sentence. Say Go Ahead to translate or Rephrase your sentence. ASR: **i am going to the village** Sys.: Translating "i am going to the village"
F. Idiom *User Utt.:* What can we do about this dead space here? ASR: **what can we do about this dead space here** Sys.: I heard "what can we do about this dead space here". Idioms like dead space may not be translated accurately. Say Go Ahead for literal translation, or Rephrase your sentence. ASR: **go ahead** Sys.: Translating "what can we do about this dead space here"

is suitable for handling multiple error types including OOVs, mispronunciations, and ASR/MT errors. Additionally, the *ASR Errors* strategy has been designed to capture a large fraction of the OOV name false negatives (i.e. missed detections) by allowing the user to indicate if the error span is a name (Excerpt B). Because of the similar nature of the errors handled by these two strategies, we have found it beneficial to maintain reciprocity between them to resolve all the errors handled by these strategies.

The four *Word Sense* (WS) disambiguation strategies resolve sense ambiguity errors. The underlying principle behind the strategies is that the sense of an ambiguous word must be confirmed by at least two of four possible independent sources. These four sources include (a) the translation system (sense lookup corresponding to phrase pair associated with the ambiguous word), (b) sense-inventory that lists source phrase keywords, (c) sense predicted by supervised model for sense-class and (d) sense specified by the user. Some of these sources may not be available for certain words. *Case 2: Filtered* strategy corresponds to the case where (a) and (b) agree. In this case, the user is shown a message using the GUI and the system proceeds to present the translation to the Arabic speaker. Similarly, *Case 1: No Mismatch* strategy correspond to the case where (a) and (c) agree. If these three sources are unable to resolve the sense of the word, the user is asked to confirm the sense identified by source (a) following the *Case 3: Mismatch* strategy. If the user rejects that sense, a list of senses is presented to the user (*Case 4: Backoff* strategy). The user-specified sense drives constrained decoding to obtain an accurate translation which is then presented to the Arabic speaker. An example of this case is shown in Excerpt C of Table 5.

Albeit simpler, the two homophone (HP) resolution strategies mimic the WS strategies in principle and design. The observed homophone variant produced by the ASR must be confirmed either by the MaxEnt model (*Case 1: No Mistmatch*) of the corresponding homophone class or by the user (*Case 2: Mismatch*) as shown in Excerpt D. The input utterance is modified (if needed) by substituting the resolved homophone variant in the ASR output which is then translated and presented to the Arabic speaker.

Strategies for resolving errors associated with idioms and incomplete utterances (Excerpts E and F) primarily rely on informing the user about the detection of these errors. The user is expected to rephrase the utterance to avoid these errors. For idioms, the user is also given the choice to force a literal translation when appropriate.

At all times, the user has the ability to rephrase the initial utterance as well as to force the system to proceed with the current translation. This allows the user to override system false alarms whenever suitable. The interface also allows the user to repeat the last system message which is helpful for comprehension of long prompts presented by the system.

4. Experimental Results

In this section, we present results from a preliminary evaluation for measuring the benefit of active error detection and resolution capability in S2S systems. Note that this evaluation does not contrast the various design choices involved in our implementation. Instead, we focus on a holistic evaluation of the system.

4.1. Evaluation Approach and Metrics

Multiple English speaking human subjects interacted with the system to communicate 20 scenarios to an Arabic speaker. Each scenario consists of 5 "starting" utterances. The subject speaks one English starting utterance at a time and is allowed to freely respond to any interactive recovery dialog initiated by the system. Interaction corresponding to each starting utterance comes to an end when the system presents an Arabic

translation. Each starting utterance has been designed to pose exactly one of the seven error types discussed in Section 2. This is often compounded by unexpected ASR errors.

Prior to the start of the experiment, each speaker was trained using five scenarios (25 starting utterances) to allow the speakers to familiarize themselves with the system prompts. In all, we were able to collect interactions corresponding to 103 starting utterance for this evaluation. The primary measure of success of a S2S system is its ability to accurately communicate concepts across the language pair. High Level Concept Transfer (HLCT) [22] has been used in the past for multi-site S2S system evaluations under the DARPA TRANSTAC program.

In this paper, we adapt HLCT to focus on the concept associated with the erroneous span (word/phrase) in each starting utterance. We consider only the span associated with the intended error. Each erroneous concept is considered as transferred if it is conveyed accurately in the translation. The benefit of using active error detection and recovery is measured as the improvement in HLCT between the initial translation (i.e. before recovery) and final translation (i.e. after recovery). This is demonstrated in Table 6. In addition to improvement in concept transfer, we also present error detection accuracy metrics as well as analysis of number of clarification turns.

Table 6: Example of HLCT for Erroneous Concept

User Utt: i have heard that the utility prices are extortionate
Before Clarification
ASR: **i have heard that the utility prices are extort unit**
MT: آنيسمعتإنهالخدماتالأسعاروحدة
Gloss: I heard that services all prices are same
Concept Transferred? **No ✗**
After Clarification
ASR: **the price for utilities seems very high**
MT: السعرالخدماتمبينكلشعالية
Gloss: the price of services seem to be very high
Concept Transferred? **Yes ✓**

4.2. Results

Table 7: HLCT for Erroneous Spans
(#: count of utterances transferred, %: percentage transferred)

Intended Error	Count	Initial Transfer		Final Transfer		Change
		#	%	#	%	%
OOV-Name	12	1	8.33	5	41.67	33.33
OOV-Word	46	3	6.52	20	43.48	36.96
Word Sense	18	4	22.22	10	55.56	33.33
Homophone	15	4	26.67	5	33.33	6.67
Mispronunciation	5	1	20.00	2	40.00	20.00
Idiom	2	0	0.00	1	50.00	50.00
Incomplete	5	0	0.00	5	100.00	100.00
All	**103**	**13**	**12.62**	**48**	**46.60**	**33.98**

ASR WER for the utterances used in this evaluation was 23%. Table 7 shows the initial, final and change (improvement) in HLCT for the erroneous span for each of the error types. Overall, our S2S system equipped with active error detection and recovery is able to improve the transfer of erroneous concepts by 33.98%. This improvement is more prominent in the case of certain types of errors such as OOVs.

Table 8 shows the detection accuracy within our evaluation set for each type of error. Two different detection accuracy metrics are shown. First, %correct is the fraction of errors that were identified as the intended error. Second, %recoverable is the fraction of errors that were identified as an error whose strategy supports recovery from the intended error. For example, an OOV-Name incorrectly identified as an error span is still recoverable because the strategy allows the user to inform the system that the span is a name. Note that %recoverable is always greater than or equal to %correct because correctly identified errors is considered recoverable in this analysis. Overall, 33% of errors are identified correctly and 59.2% are identified as a potentially recoverable error. Of these, as shown in Table 7, 46.6% errors are actually recovered by our recovery strategies. On average, the recovery strategies require 1.4 clarification turns.

Table 8: Error Detection Accuracy
(*Intended and Actual Errors may differ)

Intended Error	%Correct	%Recoverable
OOV-Name	41.7	75.0
OOV-Word	37.8	75.6
Word Sense*	16.7	16.7
Homophone*	31.3	50.0
Mispronunciation	60.0	60.0
Idiom	0.0	0.0
Incomplete	20.0	80.0
All	**33.0**	**59.2**

5. Discussion and Future Work

Error recovery strategies have been shown to be effective at improving task success in several applications [23][24]. However, their application to S2S systems has been limited [10][25]. In [25], the authors developed a wide range of repair strategies for narrow domain S2S. However, this implementation did not have any active error detection. Instead, it was delegated to the user who was asked to highlight erroneous words resulting from ASR errors.

The active error detection and interactive recovery strategies described in this paper go well beyond user confirmation [10] and repair strategies of [25]. As seen in the results presented in Section 4, well-designed error-specific recovery strategies can significantly improve (34%) the communication of erroneous concepts despite moderate error detection capabilities (33%). We also note that this state-of-the-art implementation is able to recover only about 46.6% erroneous concepts. This suggests a significant scope for improvement of S2S systems in this line of investigation.

While our current system has demonstrated an effective approach for enhancing eyes-free S2S systems with active error detection and recovery, this system implements these capabilities in only one direction (English to Arabic). Developing similar capabilities in both directions of S2S presents exciting challenges. In particular, the participation of the foreign language speaker in the error recovery activity offers both opportunities for developing novel interaction

strategies as well as challenges such as addressee detection, speaker diarization and prompt targeting in addition to addressing increased computational needs for bi-directional error detection.

In addition to extending our system to a 2-way implementation, further scientific inquiry to evaluate the effectiveness of error recovery in S2S systems is necessary. Specifically, evaluation presented in this paper has two shortcomings. First, each utterance in the evaluation scenarios is designed to have one of the 7 expected errors. This was necessary in these preliminary evaluations to gather a representative sample of each of types of error within a reasonable number of utterances collectable with a small number of human subjects. However, in practice, many utterances may have none or multiple expected errors. While our current system is capable of dealing with these situations, the evaluation presented here does not measure system performance under such conditions.

Second, in a practical S2S system, often the two speakers are able to perform limited amount of error recovery. While this form of error recovery is often expensive in terms of user time and effort, a thorough evaluation should compare this form of recovery to automated error recovery.

6. References

[1] Wahlster, W., "Verbmobil: translation of face-to-face dialogs", *Proc. of European Conf. on Speech Comm. And Tech., 1993, p. 29-38*

[2] Nakamura, S., Markov, K., Nakaiwa, H., Kikui, G., Kawai, H., Jitsuhiro, T., Zhang, J.S., Yamamoto, H., Sumita, E., and Yamamoto, S. "The ATR multilingual speech-to-speech translation system," *IEEE Trans. on Audio, Speech, and Language Processing, 14.2 p. 365-376, 2006*

[3] Stallard, D., Prasad, R., Natarajan, P., Choi, F., Saleem, S., Meermeier, R., Krstovski, K., Ananthakrishnan, S., and Devlin, J. "The BBN TransTalk Speech-to-Speech Translation System", *Speech and Language Technologies, InTech, 2011, p. 31-52*

[4] Google Translate, http://translate.google.com/

[5] Eck, M., Lane, I., Zhang, Y., and Waibel, A. "Jibbigo: Speech-to-speech translation on mobile devices," *IEEE Wksp. on SLT, 2010, p.165-166*

[6] Zhang, R., Kikui, G., Yamamoto, H., Watanabe, T., Soong, F., and Lo, W. K. "A unified approach in speech-to-speech translation: integrating features of speech recognition and machine translation", *Proc. of 20th COLING, Stroudsburg, PA, USA, 2004*

[7] He, X. and Deng, L. "Optimization in Speech-Centric Information Processing: Criteria and techniques", *Proc. of ICASSP, 2012, p. 5241-5244*

[8] Matsoukas, S., Bulyko, I., Xiang, B., Nguyen, K., Schwartz, R. and Makhoul, J. "Integrating Speech Recognition and Machine Translation," *Proc. of ICASSP, 2007, p. 1281- 1284*

[9] Stallard, D., Kao, C. Krstovski, K., Liu, D., Natarajan, P., Prasad, R., Saleem, S., and Subramanian, K., "Recent improvements and performance analysis of ASR and MT in a speech-to-speech translation system", *Proc. of ICASSP, 2008, p. 4973-4976*

[10] Prasad, R., Natarajan, P., Stallard, D., Saleem, S., Ananthakrishnan, S., Tsakalidis, S., Kao, C.-L., Choi, F., Meermeier, R., Rawls, M., Devlin, J., Krstovski, K., Challenner, A. "BBN TransTalk: Robust multilingual two-way speech-to-speech translation for mobile platforms," *Computer Speech & Language, 2011*

[11] Nguyen L., and Schwartz, R. "Efficient 2-pass N-best decoder," *Proc. of Eurospeech, Rhodes, Greece, 1997, p. 167–170.*

[12] Brown, P. E., Della Pietra, V. J., Della Pietra, S. A., and Mercer, R. L. "The Mathematics of Statistical Machine Translation: Parameter Estimation", *Computational Linguistics, 19, 1993, p. 263–311*

[13] Koehn, P., Och, F. J., and Marcu, D. "Statistical Phrase-based Translation", *NAACL-HLT, 2003, p. 48–54*

[14] Och, F. J., "Minimum Error Rate Training in Statistical Machine Translation", *Proc. of 41st ACL, Stroudsburg, PA, USA, 2003, pp. 160-167*

[15] Ananthakrishnan, S., Prasad, R., and Natarajan, P. "An Unsupervised Boosting Technique for Refining Word Alignment", *Proc. of IEEE Wksp. on SLT, Berkeley, CA, 2010*

[16] Ananthakrishnan, S., Prasad, R., and Natarajan, P., "Phrase Alignment Confidence for Statistical Machine Translation", *Proc. of Interspeech, Makuhari, Japan, 2010, p. 2878-2881*

[17] Bach, N., Huang F. and Al-Onaizan, Y. "Goodness: A Method for Measuring Machine Translation Confidence", *Proc. of 49th ACL-HLT, 2011, Portland, OR, USA*

[18] Kumar, R., Prasad, R., Ananthakrishnan, S., Vembu, A. N., Stallard, D., Tsakalidis, S., Natarajan, P. "Detecting OOV Named-Entities in Conversational Speech", *Proc. of Interspeech, 2012, Portland, OR, USA*

[19] Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. "Constrained k-means Clustering with Background Knowledge", *Proc. of 18th ICML, 2001, San Francisco, CA, USA, p. 577–584*

[20] Turunen, M. and Hakulinen, J. "Jaspis - An Architecture for Supporting Distributed Spoken Dialogues", *Proc. of Eurospeech, Geneva, Switzerland, 2003*

[21] Nakano, M., Funakoshi, K., Hasegawa, Y., Tsujino, H., "A Framework for Building Conversational Agents Based on a Multi-Expert Model", *Proc. of 9th SigDial Workshop on Discourse and Dialog, Columbus, Ohio, 2008*

[22] Weiss, B. A., Schlenoff, C.I., Sanders, G. A., Steves, M. P., Condon, S., Phillips, J. and Parvaz, D. "Performance Evaluation of Speech Translation Systems", *Proc. of 6th LREC, 2008*

[23] Turunen, M. and Hakulinen, J. "Agent-based Error Handling in Spoken Dialogue Systems", *Proc. of Eurospeech, 2001*

[24] Bohus, D. and Rudnicky, A. I. "Sorry, i didn't catch that!- an investigation of non-understanding errors and recovery strategies", *Proc. of SIGDial, 2005, p. 128-143*

[25] Suhm, B., Myers, B. and Waibel, A. "Interactive recovery from speech recognition errors in speech user interfaces," *Proc. of 4th ICSLP, 1996. p.865-868*

A Method for Translation of Paralinguistic Information

Takatomo Kano, Sakriani Sakti, Shinnosuke Takamichi, Graham Neubig, Tomoki Toda, Satoshi Nakamura

Graduate School of Information Science, Nara Institute of Science and Technology

Abstract

This paper is concerned with speech-to-speech translation that is sensitive to paralinguistic information. From the many different possible paralinguistic features to handle, in this paper we chose duration and power as a first step, proposing a method that can translate these features from input speech to the output speech in continuous space. This is done in a simple and language-independent fashion by training a regression model that maps source language duration and power information into the target language. We evaluate the proposed method on a digit translation task and show that paralinguistic information in input speech appears in output speech, and that this information can be used by target language speakers to detect emphasis.

1. Introduction

In human communication, speakers use many different varieties of information to convey their thoughts and emotions. For example, great speakers enthrall their listeners by not only the contents of the speech but also their zealous voice and confident looks. This paralinguistic information is not a factor in written communication, but in spoken communication it has great importance. These acoustic and visual cues transmit additional information that cannot be expressed in words. Even if the context is the same, if the intonation and facial expression are different an utterance can take an entirely different meaning [1, 2].

However, the most commonly used speech translation model is the cascaded approach, which treats Automatic Speech Recognition (ASR), Machine Translation (MT) and Text-to-Speech (TTS) as black boxes, and uses words as the basic unit for information sharing between these three components. There are several major limitations of this approach.

For example, it is widely known that errors in the ASR stage can propagate throughout the translation process, and considering several hypotheses during the MT stage can improve accuracy of the system as a whole [3]. Another less noted limitation, which is the focus of this paper, is that the input of ASR contains rich prosody information, but the words output by ASR have lost all prosody information. Thus, information sharing between the ASR, MT, and TTS modules is weak, and after ASR source-side acoustic details are lost (for example: speech rhythm, emphasis, or emotion).

In our research we explore a speech-to-speech translation system that not only translates linguistic information, but also paralinguistic speech information between source and target utterances. Our final goal is to allow the user to speak a foreign language like a native speaker by recognizing the input acoustic features (F0, duration, power, spectrum etc.) so that we can adequately reconstruct these details in the target language.

From the many different possible paralinguistic features to handle, in this paper we chose duration and power. We propose a method that can translate these paralinguistic features from the input speech to the output speech in continuous space. In this method, we extract features at the level of Hidden Markov Model (HMM) states, and use linear regression to translate them to the duration and power of HMM states of the output speech. We perform experiments that use this technique to translate paralinguistic features and reconstruct the input speech's paralinguistic information, particularly emphasis, in output speech.

We evaluate the proposed method by recording parallel emphasized utterances and using this corpus to train and test our paralinguistic translation model. We measure the emphasis recognition rate and intensity by objective and subjective assessment, and find that the proposed paralinguistic translation method is effective in translating this paralinguistic information.

2. Conventional Speech-to-Speech Translation

Conventionally, speech to speech translation is composed of ASR, MT, and TTS. First, ASR finds the best source language sentence $\mathbf{E}$ given the speech signal S,

$$\hat{\mathbf{E}} = \arg\max_{\mathbf{E}} P(\mathbf{E}|S). \tag{1}$$

Second, MT finds the best target language sentence $\mathbf{J}$ given the sentence $\mathbf{E}$,

$$\hat{\mathbf{J}} = \arg\max_{\mathbf{J}} P(\mathbf{J}|\hat{\mathbf{E}}). \tag{2}$$

Finally, TTS finds finds the best target language speech parameter vector sequence $\mathbf{C}$ given the sentence $\hat{\mathbf{J}}$,

$$\hat{\mathbf{C}} = \arg\max_{\mathbf{C}} P(\mathbf{O}|\hat{\mathbf{J}}) \tag{3}$$

$$\text{subject to } \mathbf{O} = \mathbf{MC}, \tag{4}$$

where $\mathbf{O}$ is a joint static and dynamic feature vector sequence of the target speech parameters and $\mathbf{M}$ is a transformation matrix from the static feature vector sequence into the joint static and dynamic feature vector sequence.

It should be noted that in the ASR step here we are translating speech S, which is full of rich acoustic and prosodic cues, into a simple discrete string of words $\mathbf{E}$. As a result, in conventional systems all of the acoustic features of speech are lost during recognition, as shown in Figure 1. These features include the gender of the speaker, emotion, emphasis, and rhythm. In the TTS stage, acoustic parameters are generated from the target sentence and training speech only, which indicates that they will reflect no feature of the input speech.

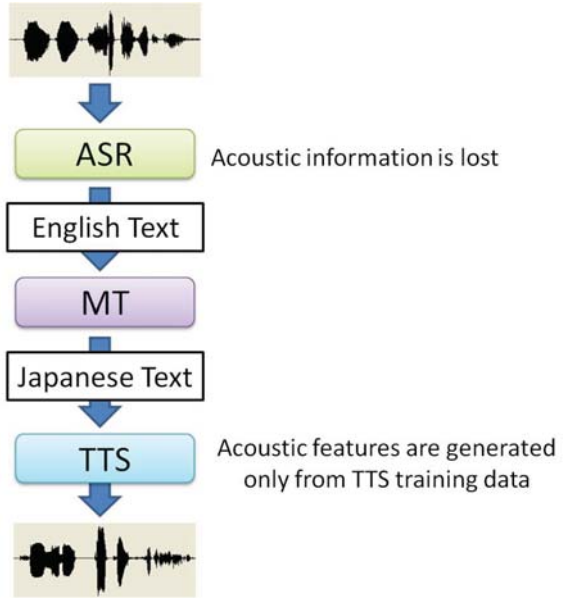


Figure 1: Conventional speech to speech translation model

3. Acoustic Feature Translation Model

In order to resolve this problem of lost acoustic information, we propose a method to translate paralinguistic features of the source speech into the target language. Our proposed method consists of three parts: word recognition and feature extraction with ASR, lexical and paralinguistic translation with MT and linear regression respectively, and speech synthesis with TTS. While this is the same general architecture as traditional speech translation systems, we add an additional model to translate not only lexical information but also two types of paralinguistic information: duration and power. In this paper, in order to focus specifically on paralinguistic translation we chose a simple, small-vocabulary lexical MT task: number-to-number translation.

3.1. Speech Recognition

The first step of the process uses ASR to recognize the lexical and paralinguistic features of the input speech. This can be represented formally as

$$\hat{\mathbf{E}}, \hat{\mathbf{X}} = \arg\max_{\mathbf{E},\mathbf{X}} P(\mathbf{E}, \mathbf{X}|S), \tag{5}$$

where S indicates the input speech, $\mathbf{E}$ indicates the words included in the utterance and $\mathbf{X}$ indicates paralinguistic features of the words in $\mathbf{E}$.

In order to recognize this information, we construct a word-based HMM acoustic model. The acoustic model is trained with audio recordings of speech and the corresponding transcriptions $\mathbf{E}$ using the standard Baum-Welch algorithm. Once we have created our model, we perform simple speech recognition using the HMM acoustic model and a language model that assigns a uniform probability to all digits. Viterbi decoding can be used to find $\mathbf{E}$.

Finally we can decide the duration and power vector $\boldsymbol{x_i}$ of each word e_i. The duration component of the vector is chosen based on the time spent in each state of the HMM acoustic model in the path found by the Viterbi algorithm. For example, if word e_i is represented by the acoustic model A, the duration component will be a vector with length equal to the number of HMM states representing e_i in A, with each element being an integer representing the number of frames emitted by each state. The power component of the vector is chosen in the same way, and we take the mean value of each feature over frames that are aligned to the same state of the acoustic model. We express power as $[power, \Delta power, \Delta\Delta power]$ and join these features together as a super vector to control power in the translation step.

3.2. Lexical Translation

Lexical translation is defined as finding the best translation $\mathbf{J}$ of sentence $\mathbf{E}$.

$$\hat{\mathbf{J}} = \arg\max_{\mathbf{J}} P(\mathbf{J}|\mathbf{E}), \tag{6}$$

where $\mathbf{J}$ indicates the target language sentence and $\mathbf{E}$ indicates the recognized source language sentence. Generally we can use a statistical machine translation tool like Moses [4], to obtain this translation in standard translation tasks. However in this paper we have chosen a simple number-to-number translation task so we can simply write one-to-one lexical translation rules with no loss in accuracy.

3.3. Paralinguistic Translation

Paralinguistic translation converts the source-side duration and mean power vector $\mathbf{X}$ into the target-side duration and mean power vector $\mathbf{Y}$ according to the following equation

$$\hat{\mathbf{Y}} = \arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}). \tag{7}$$

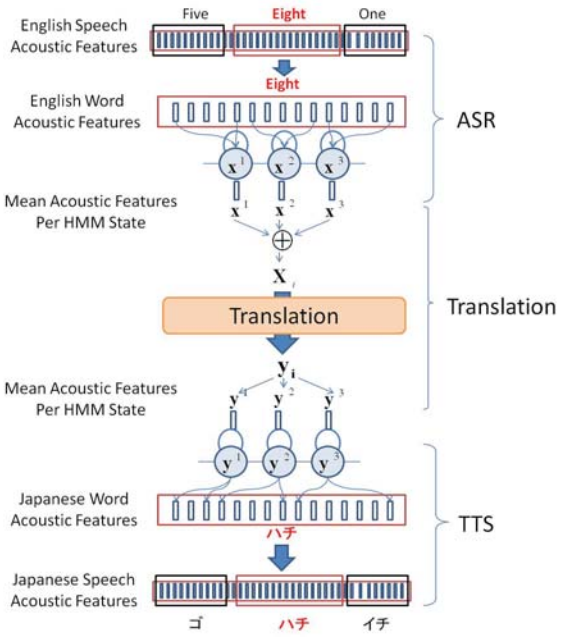


Figure 2: Overview of paralinguistic translation

In particular, we control duration and power of each word using a source-side duration and power super vector $\boldsymbol{x_i} = [\boldsymbol{x_1}, \cdots, \boldsymbol{x_{N_x}}]^\top$ and a target-side duration and power super vector $\boldsymbol{y_i} = [\boldsymbol{y_1}, \cdots, \boldsymbol{y_{N_y}}]^\top$. In these vectors N_x represents the number of HMM states on the source side and N_y represents the number of HMM states on the target side. $^\top$ indicates transposition. The sentence duration and power vector consists of the concatenation of the word duration and power vectors such that $\mathbf{Y} = [\boldsymbol{y_1}, \cdots, \boldsymbol{y_i}, \cdots, \boldsymbol{y_I}]$ where I is the length of the sentence. In this work, to simplify our translation task, we assume that duration and power translation of each word pair is independent from that of other words, allowing us to find the optimal Y using the following equation:

$$\hat{\mathbf{Y}} = \arg\max_{\mathbf{Y}} \prod_i P(\boldsymbol{y_i}|\boldsymbol{x_i}). \tag{8}$$

The word-to-word acoustic translation probability $P(\boldsymbol{y_i}|\boldsymbol{x_i})$ can be defined with any function, but in this work we choose to use linear regression, which indicates that $\boldsymbol{y_i}$ is distributed according to a normal distribution

$$P(\boldsymbol{y_i}|\boldsymbol{x_i}) = N(\boldsymbol{y_i}; \mathbf{W}_{e_i,j_i}\boldsymbol{x'_i}, S) \tag{9}$$

where x' is $\left[1x^\top\right]^\top$ and $\mathbf{W}_{e_i,j_i}$ is a regression matrix (including a bias) defining a linear transformation expressing the relationship in duration and power between e_i and j_i. An important point here is how to construct regression matrices for each of the words we want to translate. In order to do so, we optimize each regression matrix on the translation model training data by minimize root mean squared error (RMSE) with a regularization term

$$\hat{\mathbf{W}}_{e,j} = \arg\min_{\mathbf{W}_{e_i,j_i}} \sum_{n=1}^{N} ||\boldsymbol{y^*}_n - \boldsymbol{y}_n||^2 + \alpha||\mathbf{W}_{e_i,j_i}||^2, \tag{10}$$

where N is the number of training samples, n is the id of each training sample, $\boldsymbol{y^*}$ is target language reference word duration and power vector, and α is a hyper-parameter for the regularization term to prevent over-fitting.[1] This maximization can be solved efficiently in closed form using simple matrix operations.

3.4. Speech Synthesis

In the TTS part of the system we use an HMM-based speech synthesis system [5], and reflect the duration and power information of the target word paralinguistic information vector onto the output speech. The output speech parameter vector sequence $\mathbf{C} = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_T]^\top$ is determined by maximizing the target HMM likelihood function given the target word duration and power vector $\hat{\mathbf{Y}}$ and the target language sentence $\hat{\mathbf{J}}$ as follows:

$$\hat{\mathbf{C}} = \arg\max_{\mathbf{C}} P(\mathbf{O}|\hat{\mathbf{J}}, \hat{\mathbf{Y}}) \tag{11}$$

$$\text{subject to } \mathbf{O} = \mathbf{MC}, \tag{12}$$

where $\mathbf{O}$ is a joint static and dynamic feature vector sequence of the target speech parameters and $\mathbf{M}$ is a transformation matrix from the static feature vector sequence into the joint static and dynamic feature vector sequence.

While TTS generally uses phoneme-based HMM models, we instead used a word based HMM to maintain the consistency of feature extraction and translation. In this task the vocabulary is small, so we construct an independent context model.

4. Evaluation

4.1. Experimental Setting

We examine the effectiveness of the proposed method through English-Japanese speech-to-speech translation experiments. In these experiments we assume the use of speech-to-speech translation in a situation where the speaker is attempting to reserve a ticket by phone in a different language. When the listener accidentally makes a mistake when listening to the ticket number, the speaker re-speaks, emphasizing the place where the listener has made the mistake. In this situation, if we can translate the paralinguistic information, particularly emphasis, this will provide useful information to the listener about where the mistake is. This information will not be present with linguistic information only.

[1] We chose α to be 10 based on preliminary tests but the value had little effect on subjective results.

In order to simulate this situation, we recorded a bilingual speech corpus where an English-Japanese bilingual speaker emphasizes one word during speech in a string of digits. The lexical content to be spoken was 500 sentences from the AURORA2 data set, chosen to be word balanced by greedy search [6]. The training set is 445 utterances and the test set is 55 utterances, graded by 3 evaluators. We plan to make this data freely available by the publication of this paper.

Before the experiments, we analyzed the recorded speech's emphasis. We found several inclinations of emphasized segments such as shifts in duration and power. For example there are often long silences before or after emphasized words, and the emphasized word itself becomes longer and louder.

We further used this data to build an English-Japanese speech translation system that include our proposed paralinguistic translation model. We used the AURORA2 8440 utterance bilingual speech corpus to train the ASR module. Speech signals were sampled at 8kHz with utterances from 55 males and 55 females. We set the number of HMM states per word in the ASR acoustic model to 16, the shift length to 5ms, and other various settings for ASR to follow [7]. For the translation model we use 445 utterances of speech from our recorded corpus for training and hold out the remainder for testing. As the recognition and translation tasks are simple are simple , the ASR and MT models achieved 100% accuracy on every sentence in the test set. For TTS, we use the same 445 utterances for training an independent context synthesis model. In this case, the speech signals were sampled at 16kHz. The shift length and HMM states are identical to the setting for ASR.

In the evaluation, we compare the baseline and two proposed models shown below:

Baseline: traditional lexical translation model only

Duration: Paralinguistic translation of duration only

Duration + Power: Paralinguistic translation of duration and power

The word translation result is the same between both models, but the proposed model has more information than the baseline model with regards to duration and power. In addition, we use naturally spoken speech as an oracle output. We evaluate both varieties of output speech with respect to how well they represent emphasis.

4.2. Experimental Results

We first perform an objective assessment of the translation accuracy of duration and power, the results of which are found in Figure 3 and Figure 4. For each of the nine digits plus "oh" and "zero," we compared the difference between the proposed and baseline duration and power and the reference speech duration and power in terms of RMSE. From these results, we can see that the target speech duration and power output by the proposed method is more similar to the reference than the baseline over all eleven categories, indicating the proposed method is objectively more accurate in translating duration and power.

Training sentences	8440
Word error rate	0
HMM states	16

Table 1: Setting of ASR

Training utterances	445
Test utterances	55
Regularization term	10

Table 2: Setting of paralinguistic translation

Training utterances	445
HMM states	16

Table 3: Setting of TTS

As a subjective evaluation we asked native speakers of Japanese to evaluate how well emphasis was translated into the target language. The first experiment asked the evaluators to attempt to recognize the identities and positions of the emphasized words in the output speech. The overview of the result for the word and emphasis recognition rates is shown in Figure 5. We can see that both of the proposed systems show a clear improvement in the emphasis recognition rate over the baseline. Subjectively the evaluators found that there is a clear difference in the duration and power of the words. In the proposed model where only duration was translated, many testers said emphasis was possible to recognize, but sometime it was not so clear and they were confused. When we also translate power, emphasis became more clear and some examples of emphasis that only depended on power were also able to be recognized. When we examined the remaining errors, we noticed that even when mistakes were made, mistakenly recognized positions tended to be directly before or after the correct word, instead of being in an entirely different part of the utterance.

The second experiment asked the evaluators to subjectively judge the strength of emphasis, graded with the following three degrees.

1: not emphasized

2: slightly emphasized

3: emphasized

The overview of the experiment regarding the strength of emphasis is shown in Figure 6. This figure shows that there

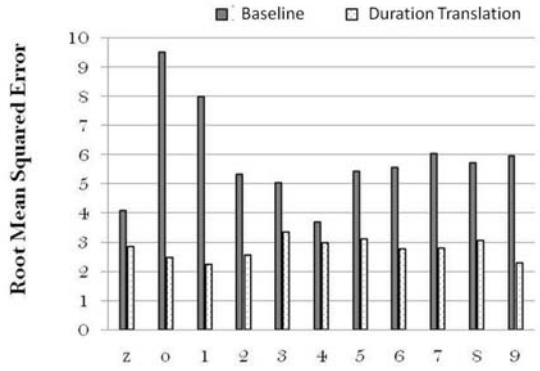


Figure 3: Root mean squared error rate (RMSE) between the reference target duration and the system output for each digit

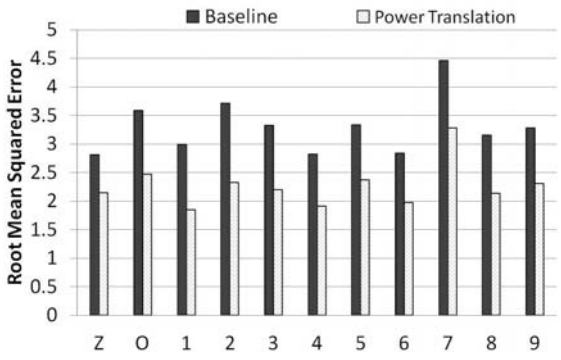


Figure 4: Root mean squared error rate (RMSE) between the reference target power and the system output for each digit

is a significant improvement in the subjective perception of strength of emphasis as well. Particularly, when we analyzed the result we found two interesting trends between duration translation and duration and power translation. Particularly, the former method was often labeled with a score of 2 indicating that the duration is not sufficient to represent emphasis clearly. However, duration+power almost always scored 3 and can be recognized as the position of emphasis. This means that in English-Japanese speech translation, speech's power is an important factor to convey emphasis.

5. Related Works

There have been several studies demonstrating improved speech translation performance by utilizing paralinguistic information of source side speech. For example, [8] focuses on using the input speech's acoustic information to improve translation accuracy. They try to explore a tight coupling of ASR and MT for speech translation, sharing information on the phone level to boost translation accuracy as measured by BLEU score. Other related works focus on using speech intonation to reduce translation ambiguity on the target side

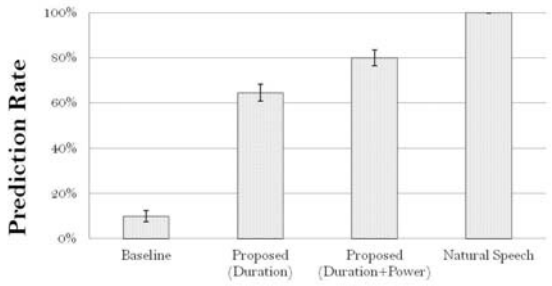


Figure 5: Prediction rate

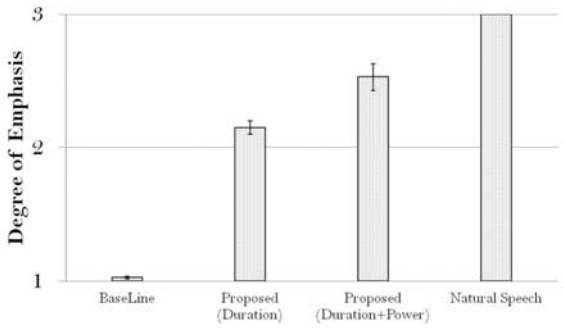


Figure 6: Degree of emphasis

[9, 10].

While the above methods consider paralinguistic information to boost translation accuracy, as we mentioned before, there is more to speech translation than just the accuracy of the target sentence. It is also necessary to consider other features such as the speaker's facial and prosodic expressions to fully convey all of the information included in natural speech. There is some research that considers translating these expressions and improves speech translation quality in other ways that cannot be measured by BLEU. For example some work focuses on mouth shape and uses this information to translate speaker emotion from source to target [1, 11]. On the other hand, [2] focus on the input speech's prosody, extracting F0 from the source speech at the sentence level and clustering accent groups. These are then translated into target side accent groups. V. Kumar et al consider the prosody in encoded as factors in the Moses translation engine to convey prosody from source to target [12].

In our work, we also focus on source speech paralinguistic features, but unlike previous work we extract them and translate to target paralinguistic features directly and in continuous space. In this framework, we need two translation models. One for word-to-word lexical translation, and another for paralinguistic translation. We train a paralinguistic translation model with linear regression for each word pair. This allows for relatively simple, language-independent implementation and is more appropriate for continuous features such as duration and power.

6. Conclusion

In this paper we proposed a method to translate duration and power information for speech-to-speech translation. Experimental results showed that duration and power information in input speech appears in output speech, and that this information can be used by target language speakers to detect emphasis.

In future work we plan to expand beyond the easy lexical translation task in the current paper to a more general translation task. Our next step is to expand our method to work with phrase-based machine translation. Phrase-based SMT handles non-monotonicity, insertions, and deletions naturally, and we are currently in the process devising methods to deal with the expand vocabulary in paralinguistic translation. In addition, traditional speech-to-speech translation, the ASR and TTS systems generally use phoneme-based HMM acoustic models. And it will be necessary to change our word-based ASR and TTS to phoneme-based systems to improve their performance on open-domain tasks. Finally, while we limited our study to duration and power, we plan to expand to other acoustic features such as F0, which play an important part in other language pairs, and also paralinguistic features other than emphasis.

7. Acknowledgment

Part of this work was supported by JSPS KAKENHI Grant Number 24240032.

8. References

[1] S. Ogata, T. Misawa, S. Nakamura, and S. Morishima, "Multi-modal translation system by using automatic facial image tracking and model- based lip synchronization," in *ACM SIGGRAPH2001 Conference Abstracts and Applications,Sketch and Applications*. Siggraph, 2001.

[2] P. D. Agero, J. Adell, and A. Bonafonte, "Prosody generation for speech-to-speech translation," in *In Proceedings of ICASSP*. ICASSP, 2006.

[3] H. Ney, "Speech translation: coupling of recognition and translation," in *Proceedings of Acoustics, Speech, and Signal Processing.* IEEE Int. Conf, 1999.

[4] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of ACL*, 2007.

[5] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis." Speech Communication, 2009.

[6] J. Zhang and S. Nakamura, "An efficient algorithm to search for a minimum sentence set for collecting speech database," in *Proceedings of the 15th International Congress of Phonetic Sciences*. ICPhS, 2003.

[7] H. G. Hirsh and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions," in *ISCA ITRW ASR2000*, 2000.

[8] J. Jiang, Z. Ahmed, J. Carson-Berndsen, P. Cahill, and A. Way, "Phonetic representation- based speech translation," in *Proceedings of Machine Translation Summit 13*, 2011.

[9] T. Takezawa, T. Morimoto, Y. Sagisaka, N. Campbell, H. Iida, F. Sugaya, A. Yokoo, and S. Yamamoto, "A Japanese-to-English speech translation system: ATR-MATRIX," in *In Proceedings of the 5th International Conference on Spoken Language Processing*. ICSLP, 1998.

[10] W. Wahlster, "Robust translation of spontaneous speech: a multi-engine approach," in *IJCAI'01 Proceedings of the 17th international joint conference on Artificial intelligence - Vol 2*. IJCAI, 2001.

[11] S. Morishima and S. Nakamura, "Multimodal translation system using texture mapped Lip-Sync images for video mail and automatic dubbing applications." EURASIP, 2004.

[12] V. Kumar, S. Bangalore, and S. Narayanan, "Enriching machine-mediated speech-to-speech translation using contextual information," 2011.

Continuous Space Language Models using Restricted Boltzmann Machines

Jan Niehues and Alex Waibel

International Center for Advanced Communication Technologies - InterACT
Institute for Anthropomatics
Karlsruhe Institute of Technology, Germany
firstname.lastname@kit.edu

Abstract

We present a novel approach for continuous space language models in statistical machine translation by using Restricted Boltzmann Machines (RBMs). The probability of an n-gram is calculated by the free energy of the RBM instead of a feed-forward neural net. Therefore, the calculation is much faster and can be integrated into the translation process instead of using the language model only in a re-ranking step.

Furthermore, it is straightforward to introduce additional word factors into the language model. We observed a faster convergence in training if we include automatically generated word classes as an additional word factor.

We evaluated the RBM-based language model on the German to English and English to French translation task of TED lectures. Instead of replacing the conventional n-gram-based language model, we trained the RBM-based language model on the more important but smaller in-domain data and combined them in a log-linear way. With this approach we could show improvements of about half a BLEU point on the translation task.

1. Introduction

Language models are very important in many tasks of natural language processing like, for example, machine translation or speech recognition. In most of these tasks, n-gram-based language models are successfully used. In this model the probability of a sentence is described as a product of the probabilities of the words given the previous words. For the conditional word probability a maximum likelihood estimation is used in combination with different smoothing techniques. Although this is often a very rough estimation, especially for rarely seen words, it can be trained very fast. This enables us to make use of huge corpora which are available for many language pairs.

But there are also several tasks where we need to build the best possible language model from a small corpus. When using a machine translation system, in many real-world scenarios we do not want to have a general purpose translation system, but a specific translation system performing well on one task, e.g. like translation of talks. For these cases, it has been shown that the translation quality can be improved significantly by adapting the system to the task. This has successfully been done by using an additional in-domain language model in the log-linear model used in statistical machine translation (SMT).

When adapting an MT system, we need to train a good language model on small amounts of in-domain data. Then the conventional n-gram-based language models often need to back-off to smaller contexts and the models do no longer perform as well. In contrast, continuous space language models (CSLMs) use always the same context size. Furthermore, the longer training time of CSLMs is no problem for small training corpora.

In contrast to most other continuous space language models, which use feed-forward neuronal nets, the probability in a Restricted Boltzmann Machine (RBM) can be calculated very efficiently. This enables us to use the language models during the decoding of the source sentence and not only in a re-scoring step.

The remaining paper is structured as follows: First we will review related work. Afterwards a brief overview of Restricted Boltzmann Machines will be given before we describe the RBM-based language model. In Section 5 we describe the results on different translation tasks. Afterwards, we will give a conclusion.

2. Related Work

A first approach to predict word categories using neural networks was presented in [1]. Later, [2] used neuronal networks for statistical language modelling. They described in detail an approach based on multi-layer perceptrons and could show that this reduces the perplexity on a test set compared to n-gram-based and class-based language models. In addition, they gave a short outlook to energy minimization networks.

An approach using multi-layer perceptrons has successfully been applied to speech recognition by [3], [4] and [5]. One main problem of continuous space language models is the size of the output vocabulary in large vocabulary continuous speech recognition. A first way to overcome this is to use a short list. Recently, [6] presented a structured output layer neural network which is able to handle large output vocabularies by using automatic word classes to group the output vocabulary.

A different approach also using Restricted Boltzmann Machines was presented in [7]. In contrast to our work, no approximation was performed and therefore, the calculation was more computation intensive. This approach and the beforementioned ones based on feed-forward networks were compared by Le et al. in [8].

Motivated by the improvements in speech recognition accuracy as well as in translation quality, authors tried to use the neural networks also for the translation model in a statistical machine translation system. In [9] as well as in [10] the authors modified the n-gram-based translation approach to use the neural networks to model the translation probabilities.

Restricted Boltzmann machines have already been successfully used for different tasks like user rating of movies [11] and images [12].

3. Restricted Boltzmann Machines

In this section we will give a brief overview on Restricted Boltzmann Machines (RBM). We will concentrate only on the points that are important for our RBM-based language model, which will be described in detail in the next section.

RBMs are a generative model that have already been used successfully in many machine learning applications. We use the following definition of RBMs as given in [13].

3.1. Layout

The RBM is a neural network consisting of two layers. One layer is the visible input layer, whose values are set to the current event. In the case of the RBM-based language model the n-gram will be represented by the states of the input layer. The second layer consists of the hidden units. In most cases those units are binary units, which can have two states. For the RBM-based language model we use "softmax" units instead of binary units for the input layer. The softmax units can have K different states instead of only two. They can be modeled as K different binary states with the restriction that exactly one binary unit is in state 1 while all others are in state 0.

In an RBM there are weighted connections between the two layers, but no connections within the layer. The layers are fully connected to each other.

3.2. Probability

The network defines a probability for a given set of states of the input and hidden units by using the energy function. Let v be the vector of all the states of the input units and h be the vector of states of the hidden units. Then the probability is defined as:

$$p(v,h) = \frac{1}{Z} e^{-E(v,h)} \tag{1}$$

using the energy function

$$E(v,h) = -\sum_{i \in visible} a_i v_i - \sum_{j \in hidden} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{2}$$

and the partition function

$$Z = \sum_{v,h} e^{-E(v,h)} \tag{3}$$

In these formulas a_i is the bias of the visible units, while b_j is the bias of the hidden units. w_{ij} is the weight of the connection between the visible unit v_i and the hidden unit h_j.

If we want to assign the probability to a word sequence, we only have the input vector, but not have the hidden value. Therefore, we would like to have the probability of this word sequence with any given hidden value. Therefore, the probability of a visible vector is defined as:

$$p(v) = \frac{1}{Z} \sum_{h} e^{-E(v,h)} \tag{4}$$

The problem of this definition is that it is exponential in the number of hidden units. A better way to calculate this probability is to use the free energy of the visible vector $F(v)$:

$$e^{-F(v)} = \sum_{h} e^{-E(v,h)} \tag{5}$$

The free energy can by calculated as:

$$F(v) = -\sum_{i} v_i a_i - \sum_{j} log(1 + e^{x_j}) \tag{6}$$

In this definition x_j is defined as $b_j + \sum_i v_i w_{ij}$. Using this definition, we are still not able to calculate the probability $p(v)$ efficiently because of Z. However, we can calculate $e^{F(v)}$ efficiently, which is proportional to the probability $p(v)$, since Z is constant for all input vectors.

3.3. Training

In most cases RBMs are trained using Contrastive Divergence [14]. The aim during training is to increase the probability of the seen training example. In order to do this, we need to calculate the derivation of probability of the example given the weights:

$$\frac{\delta log p(v)}{\delta w_{ij}} = < v_i h_j >_{data} - < v_i h_j >_{model} \tag{7}$$

where <> indicates the expectation of the value between the brackets given the distribution indicated after the brackets. The first term can be calculated easily, since there are no interconnections between the hidden units.

For the second term we use the expected value under a reconstructed distribution instead of the model distribution. This leads to a very rough approximation of the gradient, but in several experiments it was shown that it performs very well.

4. RBMs for Language modeling

After giving a general overview of RBMs we will now describe the RBM that is used for language modeling in detail. Furthermore, we will describe how we derive the sentence probability from the probabilities calculated by the RBM and how we integrate the RBM into the translation process.

4.1. Layout

The layout of the RBM used for language modeling is shown in Figure 1. The input layer of the n-gram language model consists of N blocks of input units for every word of the n-gram. Each of these blocks consists of a softmax unit, which can assume V different states representing the words of the vocabulary, where V is the vocabulary size. These softmax units are modeled by V binary units, where always exactly one unit has the value 1 and the other units have the value 0. The vocabulary consists of all the words of the text as well as the sentence end and beginning mark ($< s >$,$< /s >$) and the unknown word $< unk >$.

The hidden layer consists of H hidden units, where H is a free parameter, which will be set.

Using this setup, we need to train $N * V * H$ weights connecting the hidden and visible units as well as $N * V + H$ bias values.

4.1.1. Word Factors

For some tasks, it is interesting to not only use the surface form of the word, but consider different word factors. We can, for example, use also the part-of-speech (POS) tags of the words or we can use automatically generated word clusters. Such abstract word classes have the advantage, that they are seen more often and therefore, their weights can be trained more reliably. In this case, the additional word factor can be seen as a kind of smoothing.

The layout described before can be easily extended to also use different word factors. In that case, each of the N blocks consists of W sub-blocks, where W is the number of word factors that are used. These sub-blocks are then softmax units with different sizes depending on the vocabulary size of the factor. Like it is in the original layout, all the softmax units are then fully connected to all hidden units. The remaining layout of the framework stays the same.

4.2. Training

As it is done in most RBMs we train our model using contrastive divergence. In a first step, we collect all n-grams of the training corpus and shuffl them randomly. We then split the training examples into chunks of m examples to calculate the weight updates. This is done by calculating the difference between the products mentioned in Equation 7. The first term of the equation is straightforward to calculated. The second term is approximated using Gibbs sampling as suggested in [13]. Therefore, first the values of the hidden values are calculated given the input. Then the values of the visible units given the hidden values is calculated. And finally, a second forward calculation is used. In our experiments we only used one iterations of Gibbs sampling. In our experiments we use a value of 10 for m.

After calculating the updates, we average over all examples and then update the weights using a learning rate of 0.1. As described in [13], by averaging over the examples the size of the update is independent of m and therefore the learning rate does not need to be changed depending on the batch size. Unless stated otherwise, we perform this training for one iteration on the whole corpus.

4.3. Sentence Probabilty

Using the network described before we are able to calculate $e^{F(v)}$ efficiently, which is proportional to the probability of the n-gram $P(w_1 \dots w_N)$.

If we want to use the language model as part of a translation system, we are not interested in the probability of an n-gram, but the probability of a sentence $S =< s > w_1 \dots w_L < /s >$. In an n-gram-based language model this is done by defining the probability as a product of the word probabilities given its history $P(S) = \prod_{i=1}^{L+1} P(w_i|h_i)$, where we use $w_i =< s >$ for $i \leq 0$ and $w_i =< /s >$ for $i > L$. In an n-gram-based approach $P(w_i|h_i)$ is approximated by $P(w_i|w_{i-N+1} \dots w_{i-1})$.

In our approach we are able to calculate a score proportional to $P(w_1 \dots w_N)$ efficiently, but for the conditional probability we would need to sum over the whole vocabulary as shown in Equation 8, which would no longer be efficient.

$$P(w_i|w_{i-N+1} \dots w_{i-1}) = \frac{P(w_{i-N+1} \dots w_i)}{\sum_{w' \in V} P(w_{i-N+1} \dots w_{i-1} w')} \quad (8)$$

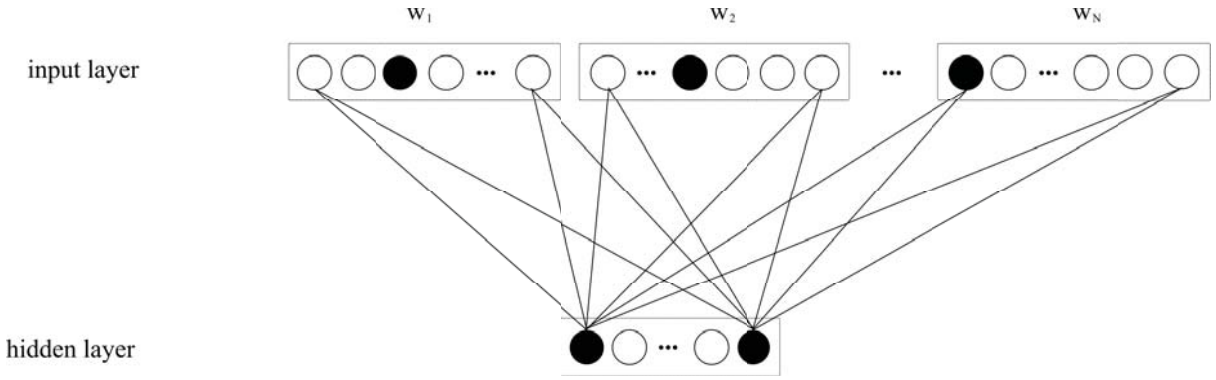
One technique often used for n-gram-based language models is to interpolate the probabilities of different history lengths. If we use the geometric mean of all n-gram probabilities up to the length N in our model we get the following definition for the conditional probability:

$$P'_{GM}(w_i|h_i) = \sqrt[N]{\prod_{j=1}^{N} P_j(w_i|w_{i-j+1} \dots w_{i-1})} \quad (9)$$

$$P_{GM}(w_i|h_i) = \frac{1}{Z_{h_i}} P'_{GM}(w_i|h_i) \quad (10)$$

where $Z_{h_i} = \sum_{w'} P'_{GM}(w'|h_i)$. Using this definition we can express the sentence probability $P_{RBM}(S)$ of our RBM-

Figure 1: RBM for Language model



based language model as:

$$
\begin{aligned}
P_{RBM}(S) &= \prod_{i=1}^{L+1} P_{GM}(w_i|h_i) \qquad (11)\\
&= \prod_{i=1}^{L+1} \frac{1}{Z_{h_i}} * \sqrt[N]{P'_{RBM}(S)}\\
P'_{RBM}(S) &= \prod_{i=1}^{L+1}\prod_{j=1}^{N} P(w_i|w_{i-j+1}\dots w_{i-1})\\
&\overset{(\dagger)}{=} \prod_{i=1}^{L} P(w_{i-N+1}\dots w_i)\\
&* \prod_{j=2}^{N} \frac{P(w_{L-j+2}\dots w_L < /s >)}{P(< s >)} * P(< /s >)\\
&= \frac{1}{Z_S}\prod_{j=1}^{L+N-1} \frac{1}{Z_M} e^{F(w_{j-N+1}\dots w_j)}
\end{aligned}
$$

In (†) we used the fact that $P(w_i|w_{i-j+1}\dots w_{i-1}) = P(w_{i-j+1}\dots w_{i-1}w_i)/P(w_{i-j+1}\dots w_{i-1})$. Then, except for the beginning and the end, all n-gram probabilties for $n < N$ cancel out. In the last line, Z_M is the partition function of the RBM and $Z_S = P(< /s >)/P(< s >)^{N-1}$.

To use the probability in the log-linear model we get:

$$
\begin{aligned}
log(P_{RBM}(S)) = &- \frac{1}{N}(log(Z_S) + (N-1) * log(Z_M))\\
&- \frac{L}{N} * log(Z_M) \qquad (12)\\
&- \sum_{i=1}^{L+1} log(Z_{h_i})\\
&+ \frac{1}{N}\sum_{j\in L+N-1} F(w_{j-N+1}\dots w_j)
\end{aligned}
$$

Here the first term is constant for all sentences, so we do not need to consider it in the log-linear model. Furthermore, the second term only depends on the length of the sentence. This is already modeled by the word count model in most phrase-based translation system. We cannot calculate the third term efficiently. If we ignore this term, it means that we approximate all n-gram probabilities by the unigram probabilities in this term, because in this case Z_{h_i} is zero. By using this approximation, we can use the last term as a good feature to describe the language model probability in our log-linear model. As described before, this part can be calculated efficiently.

The integration to the decoding process is very similar to the one used in n-gram-based language models. If we extend one translation hypothesis by a word, we have to add the additional n-gram probability to the current feature value as it is also done in the standard approach. We also have to save the context of $N-1$ words to calculate the probability. The only difference is that we add at the end of the sentence not only one n-gram ending with $< /s >$, but all the ones containing $< /s >$.

5. Evalutation

We evaluated the RBM-based language model on different tasks. We will first give a brief description of our SMT system. Then we will describe in detail our experiments on the German to English translation task. Afterwards, we will describe some more experiments on the English to French translation task.

5.1. System description

The translation system was trained on the European Parliament corpus, News Commentary corpus, the BTEC corpus and TED talks[1] . The data was preprocessed and compound splitting was applied for German. Afterwards the discriminative word alignment approach as described in [15] was applied to generate the alignments between source and target words. The phrase table was built using the scripts from the

[1] http://www.ted.com

Moses package [16]. A 4-gram language model was trained on the target side of the parallel data using the SRILM toolkit [17]. In addition we used a bilingual language model as described in [18].

Reordering was performed as a preprocessing step using POS information generated by the TreeTagger [19]. We used the reordering approach described in [20] and the extensions presented in [21] to cover long-range reorderings, which are typical when translating between German and English.

An in-house phrase-based decoder was used to generate the translation hypotheses and the optimization was performed using MERT[22].

We optimized the weights of the log-linear model on a separate set of TED talks and also used TED talks for testing. The development set consist of 1.7K segments containing 16K words. As test set we used 3.5K segments containing 31K words.

5.2. German to English

The results for translating German TED lectures into English are shown in Table 1. The baseline system uses a 4-gram language model trained on the target side of all parallel data. If we add a 4-gram RBM-based language model trained only on the TED data for 1 iteration using 32 hidden units we can improve the translation quality on the test data by 0.8 BLEU points (RBMLM H32 1Iter). We can gain additional 0.6 BLEU points by carrying out 10 instead of only 1 iteration of contrastive divergence.

If we use a factored language model trained on the surface word forms and the automatic clusters generated by the MKCLS algorithm [23] (FRBMLM H32 1Iter), we can get an improvement of 1.1 BLEU points already after the first iteration. We grouped the words into 50 word classes by the MKCLS algorithm.

If we add an n-gram-based language model trained only on the in-domain data (Baseline+NGRAM), we can improve by 1 BLEU point over the baseline system. So the factored RBM-based language model as well as the one trained for 10 iteration can outperform the second n-gram-based language model.

We can get further improvements by combining the n-gram-based in-domain language model and the RBM-based language model. In this case we use 3 different language models in our system. As shown in the lower part of Table 1, additional improvements of 0.3 to 0.4 BLEUs points can be achieved compared to the system not using any RBM-based language model. Furthermore, it is no longer as important to perform 10 iteration of training. The difference between one and 10 training iterations is quite small. The factored version of the language model still performs slightly better than the language model trained only on words.

Table 1: *Experiments on German to English*

Iterations	BLEU Score	
	Dev	Test
Baseline	26.31	23.02
+ RBMLM H32 1Iter	27.39	23.82
+ RBMLM H32 10Iter	27.61	24.47
+ FRBMLM H32 1Iter	27.54	24.15
Baseline+NGRAM	27.45	24.06
+ RBMLM H32 1Iter	27.64	24.33
+ RBMLM H32 10Iter	27.95	24.38
+ FRBMLM H32 1Iter	27.80	24.40

5.3. Network layout

We carried out more experiments on this task to analyse the influence of the network layout on the translation quality. Therefore, we used a smaller system only using the n-gram-based or RBM-based in-domain language model trained on the target side of the TED corpus. The results of these experiments are summarised in Table 2. The first system uses an n-gram-based language model trained on the TED corpus. The other systems use all an RBM-based language model trained for one iteration on the same corpus.

When comparing the BLEU scores on the development and test data, we see that we can improve the translation quality by increasing the number of hidden units to up to 32 hidden states. If we use less hidden states, the network is not able to store the probabilities of the n-grams properly. If we increase the number of hidden units further, the performance in translation quality decreases again. One reason for this might be that we have too many parameters to train given the size of the training data.

Table 2: *Experiments using different number of hidden units*

System	Hidden Units	BLEU Score	
		Dev	Test
NGRAM		27.09	23.80
RBMLM	8	25.65	23.16
	16	25.67	23.07
	32	26.40	23.41
	64	26.12	23.18

5.4. Training iterations

One critical point of the continuous space language model is the training time. While an n-gram-based language model can be trained very fast on a small corpus like the TED corpus without any parallelization, the training of the continuous space language model takes a lot longer. In our case the corpus consists of 942K words and the vocabulary size is 28K. We trained the RBM-based language model using 10 cores in parallel and it took 8 hours to train the language model for

one iteration.

Therefore, we analysed in detail the influence of the number of iterations on the translation performance. The experiments were again performed on the smaller system using no large n-gram-based language model mentioned before (No Large LM) and the system using a large n-gram language model trained on all data mentioned in the beginning (Large LM). They are summarized in Table 3. In the first line we show the performance of the system using a n-gram-based language model trained only on the TED corpus for comparison.

In these experiments, we see that the performance increases up to 10 iterations of the training data. Using 10 instead of one iteration, we can increase the translation quality by up to 0.5 BLEU points on the development data as well as on the test data. Using the large language model we could outperform the small n-gram-based language model by the RBM-based language model trained for 10 iterations. Performing more than 10 iterations does not lead to further improvements. The translation quality even decreases again. The reason for this might be that we are facing over-fitting after the 10th iteration. In the smaller setup, using the RBM language model cannot help to outperform the n-gram-based language model.

Table 3: *Experiments using different number of training iterations*

System	Iterations	No Large LM		Large LM	
		Dev	Test	Dev	Test
NGRAM		27.09	23.80	27.45	24.06
RBMLM	1	26.40	23.41	27.39	23.82
	5	26.72	23.38	27.40	23.98
	10	26.90	23.51	27.61	24.47
	15	26.57	23.47	27.63	24.22
	20	26.16	23.20	27.49	24.30

5.5. RBMLM for English-French

We also tested the RBM-based language model on the English to French translation task of TED lectures. We trained and tested the system on the data provided for the official IWSLT Evaluation Campaign 2012. The system is similar to the one used on the German to English tasks, but uses language model and phrase table adaptation to the target domain. The results for this task are shown in Table 4.

The difference between the Baseline system and the systems using RBM-based language models is smaller than in the last experiments, since the baseline system uses already several n-gram-based language models. On the development set both the RBM-based language model as well as the factored RBM-based language model using also automatic word classes could improve by 0.1 BLEU points. For the test set only the factored version can improve the translation quality by 0.1 BLEU points.

Table 4: *Experiments on English to French*

Iterations	BLEU Score	
	Dev	Test
Baseline	28.93	31.90
RBMLM	28.99	31.76
FRBMLM	29.02	32.03

6. Conclusions

In this work we presented a novel approach for continuous space language models. We used a Restricted Boltzmann Machine instead of a feed-forward neuronal net. Since this network is less complex, we were able to integrate it directly into the decoding process. Using this approach, the run-time for the calculation of the probability no longer depends on the vocabulary size, but only on the number of hidden units.

The layout of the network allows an easy integration of different word factors. We were able to improve the quality of the language model by using automatically determined word classes as an additional word factor.

As shown in the experiments, this type of language model works especially well for quite small corpora as they are typically used in the domain adaptation scenario. Therefore, the longer training time of a continuous space language model does not matter as much as for language models trained on huge amounts of data.

By integrating this language model into our statistical machine translation system, we could improve the translation quality by up to 0.4 BLEU points compared to a baseline system using already an in-domain n-gram-based language model.

7. Acknowledgements

This work was partly achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

8. References

[1] M. Nakamura, K. Maruyama, T. Kawabata, and K. Shikano, "Neural network approach to word category prediction for english texts," in *Proceedings of the 13th conference on Computational linguistics - Volume 3*, ser. COLING '90, 1990.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, Mar. 2003.

[3] H. Schwenk and J.-L. Gauvain, "Connectionist language modeling for large vocabulary continuous speech

recognition," in *In International Conference on Acoustics, Speech and Signal Processing*, 2002.

[4] H. Schwenk, "Continuous space language models," *Comput. Speech Lang.*, vol. 21, no. 3, Jul. 2007.

[5] T. Mikolov, M. Karafit, L. Burget, J. ernock, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH2010)*, vol. 2010, no. 9. International Speech Communication Association, 2010.

[6] H. S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language model," in *ICASSP*. IEEE, 2011.

[7] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[8] H. S. Le, A. Allauzen, G. Wisniewski, and F. Yvon, "Training continuous space language models: Some practical issues," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, 2010.

[9] H. Schwenk, M. R. Costa-jussa, and J. A. R. Fonollosa, "Smooth bilingual n-gram translation," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007.

[10] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous Space Translation Models with Neural Networks," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, Jun. 2012.

[11] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*, ser. ICML '07. New York, NY, USA: ACM, 2007.

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, Jul. 2006.

[13] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," Tech. Rep., 2010.

[14] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[15] J. Niehues and S. Vogel, "Discriminative Word Alignment via Alignment Matrix Modeling." in *Proc. of Third ACL Workshop on Statistical Machine Translation*, Columbus, USA, 2008.

[16] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *ACL 2007, Demonstration Session*, Prague, Czech Republic, 2007.

[17] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *Proc. of ICSLP*, Denver, Colorado, USA, 2002.

[18] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, Edinburgh, UK, 2011.

[19] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.

[20] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *TMI*, Skövde, Sweden, 2007.

[21] J. Niehues and M. Kolss, "A POS-Based Model for Long-Range Reorderings in SMT," in *Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece, 2009.

[22] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *Workshop on Data-drive Machine Translation and Beyond (WPT-05)*, Ann Arbor, MI, 2005.

[23] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes." in *EACL'99*, 1999.

Focusing Language Models For Automatic Speech Recognition

Daniele Falavigna, Roberto Gretter

HLT research unit, FBK, 38123 Povo (TN), Italy

(falavi,gretter)@fbk.eu

Abstract

This paper describes a method for selecting text data from a corpus with the aim of training auxiliary Language Models (LMs) for an Automatic Speech Recognition (ASR) system. A novel similarity score function is proposed, which allows to score each document belonging to the corpus in order to select those with the highest scores for training auxiliary LMs which are linearly interpolated with the baseline one. The similarity score function makes use of "similarity models" built from the automatic transcriptions furnished by earlier stages of the ASR system, while the documents selected for training auxiliary LMs are drawn from the same set of data used to train the baseline LM used in the ASR system. In this way, the resulting interpolated LMs are "focused" towards the output of the recognizer itself.

The approach allows to improve word error rate, measured on a task of spontaneous speech, of about 3% relative. It is important to note that a similar improvement has been obtained using an "in-domain" set of texts data not contained in the sources used to train the baseline LM.

In addition, we compared the proposed similarity score function with two other ones based on perplexity (PP) and on TFxIDF (Term Frequency x Inverse Document Frequency) vector space model. The proposed approach provides about the same performance as that based on TFxIDF model but requires both lower computation and occupation memory.

1. Introduction

Automatic speech recognition systems can significantly take advantage from training language models on large text corpora that represent well the application domain. Since, generally, a limited amount of in-domain text data is available for a given application, from which a corresponding LM is trained, the acquisition of more domain specific text data often becomes a crucial task.

It is a common practise among ASR specialists to try to automatically obtain texts relevant for the given application from large publicly available corpora and to use the collected corpora to train auxiliary LMs to be combined with the in-domain LM.

In the literature several methods are proposed for selecting text data matching an in-domain LM. In general, the approaches consist in using a function that gives a similarity score to each possible candidate text (sentences or entire documents) to select and to retain only those whose scores are higher than a predefined threshold.

In [1] the similarity function used to score documents is simply the perplexity computed using the given in-domain LM.

The work reported in [2] utilizes two unigram LMs, both trained on the general corpus to select: the first LM is trained on all texts of the corpus, the second LM is trained on all texts except the document to score. The difference in the log-likelihood of the in-domain text data given by the two LMs is used as scoring function.

The work in [3] proposes a method based on cross-entropy difference between the in-domain LM and a LM trained on a random sample of the general text data to select. The authors of this paper demonstrated significant reduction of perplexity using this method, with respect to [1] and [2], on a corpus used for automatic Machine Translation (MT).

In [4] three data selection techniques are proposed. The first one is based on a vector space model that uses TFxIDF (Term Frequency x Inverse Document Frequency) feature coefficients. A centroid similarity measure, defined as scalar product between a vector representing in-domain data and a vector representing the document to score, is employed. The second and the third methods are based on an "ngram-ratio" similarity measure and on ranking the documents of the general text corpus through resampling of in-domain data, respectively. The paper shows improvements both in perplexities and BLEU scores using all of the three selection methods. In addition, the paper demonstrates that the automatic selection approaches work well even if the set of in-domain text data, on which similarity models are estimated (both LMs or TFxIDF vectors), is replaced by texts coming from the output of the MT decoder.

More recently, some approaches have been proposed for adapting LMs using data extracted from the Web. The authors of [5] compare the usage of both manually and automatically generated texts for selecting auxiliary data for LM adaptation in a ASR task. In [6] a strategy is proposed for automatic closed-captioning of video that uses a LM adapted to the topic of the video itself. A classification is first performed to determine the topic of a given video and a large set of topic-specific LMs is trained using documents downloaded from the Web.

This work has been partially funded by the European project EU-BRIDGE, under the contract FP7-287658.

Similarly to [4] and [5] we use automatically generated documents (i.e. the documents obtained from the automatic transcriptions of the audio) to select text data from a huge general text corpus. Given an automatically transcribed document (the query document), the purpose of the selection procedure is to detect and retain from the general corpus only the documents that are most similar to a given query. Then, an auxiliary LM is trained using the automatically (query dependent) selected data. However, differently from [4], [5] and [6] we select documents for training the auxiliary LMs from the same set used to train the baseline LM employed in the ASR system, i.e. no additional documents are required to train auxiliary LMs. Finally, baseline and auxiliary LMs are linearly interpolated, as will be explained below.

This procedure allows to train LMs focused on the query document, i.e. on the ASR output. We prefer to use the term "LM focusing", instead of LM adaptation, to underline the fact that we are not using new data to train auxiliary LMs but, on the contrary, a subset of existing text data is somehow enhanced in order to better match the linguistic content of the audio to transcribe. To be more precise, we are proposing to "frequently" adapt the LM according to a given (or automatically detected) segmentation of the audio stream to transcribe. Since to do this it is necessary to train auxiliary LMs through data selection over large corpora of text data we developed an approach, similar to to TFxIDF based one, that employs a vector space model to represent documents to compare. However, the employed features, the way adopted for storing them and the similarity metrics used, has allowed to improve both computation and memory efficiency with respect to TFxIDF. In section 3 the detailed description of the proposed method and comparisons with both TFxIDF method and an approach based on perplexity minimization will be given.

The source used for LMs training is "google-news", an aggregator of news, provided and operated by Google, that collects news from many different sources, in different languages, and that groups articles having similar contents. We download daily news from this site, filter-out unuseful tags and collect texts. Therefore, a "google-news" corpus has become available for training both baseline LM and auxiliary ones.

To measure the performance of our automatic selection approach we carried out a set of experiments on the evaluation sets delivered for IWSLT 2011 Evaluation Campaign[1]. Task of this campaign is the automatic transcription/translation of TED talks, a global set of conferences whose audio/video recordings are available through the Internet (see http://www.ted.com/talks).

The simplest way for combining LMs trained on different sources is to compute the probability of a word w, given its past history h, as:

$$P[w \mid h] = \sum_{j=1}^{j=J} \lambda_j P_j[w \mid h] \quad (1)$$

where $P_j[w \mid h]$ are LM probabilities trained on the j^{th} source, λ_j are weights estimated with the aim of minimizing the overall perplexity on a development set and J is the total number of LMs to combine. More complex approaches [7] are based on linear interpolation of log-probabilities using discriminative training of λ_j (a comparison among different LM combination techniques can be found in [8]).

According to what previously seen, equation 1 is used to combine two LMs: the baseline LM (LM_{base}) and an auxiliary, i^{th} "talk-specific" LM (LM^i_{aux}), trained on auxiliary data, automatically selected. In particular, a preliminary automatic transcription of the given i^{th} TED talk is used both to select the data to train LM^i_{aux} and to estimate interpolation weights, λ^i_{base} and λ^i_{aux}, to be used with equation 1. Then, a rescoring ASR step is carried out, as explained in section 2.4, using focused, talk-specific LM probabilities given by equation 1.

We measured on IWSLT 2011 evaluation sets a relative improvement of about 3% in Word Error Rate (WER) after ASR hypotheses rescoring using auxiliary LMs trained on data selected with the proposed approach. The same improvement has been measured using TFxIDF based method for selecting auxiliary texts but, as previously mentioned, the latter method is more expensive both in terms of computation and memory requirements. Finally, a relative lower WER improvement has been achieved using an automatic selection procedure based on perplexity minimization.

2. Automatic Transcription System

The automatic transcription system used in this work is the one described in [9, 10]. It is based on two decoding passes followed by a third linguistic rescoring step.

For IWSLT 2011 evaluation campaign speech segments to transcribe have been manually detected and labelled in terms of speaker names. Then, audio recordings with manual segments to transcribe have been furnished to participants, hence no automatic speaker diarization procedure has been applied.

In both first and second decoding passes the system uses continuous density Hidden Markov Models (HMMs) and a static network embedding the probabilities of the baseline LM. A frame synchronous Viterbi beam-search is used to find the most likely word sequence corresponding to each speech segment to recognize. In addition, in the second decoding pass the system generates for each speech segment a word graph (see below for the details). The best word sequences generated in the second decoding pass are used to evaluate the baseline performance, as well as for selecting auxiliary documents. The corresponding word graphs are rescored in the third decoding pass using the focused LMs. Note that in this latter decoding step acoustic model probabil-

[1] visit http://www.iwslt2011.org/ for details of the IWSLT 2011 evaluation campaign

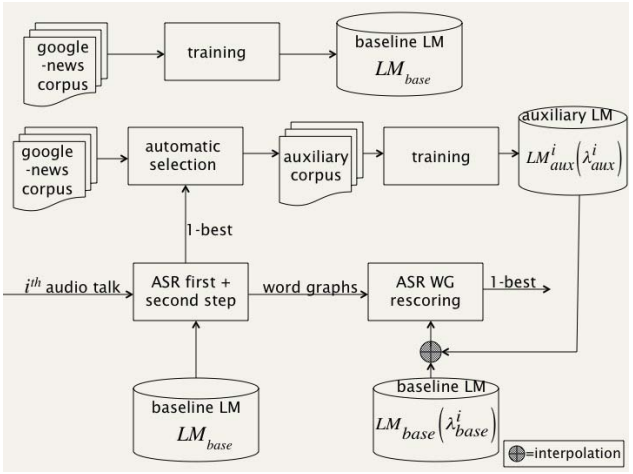


Figure 1: *Block diagram of the ASR system.*

ities associated to arcs of word graphs remain unchanged, i.e. the third decoding step implements a pure linguistic rescoring. Figure 1 shows a block diagram of the ASR system with the main modules involved, emphasizing both the procedure for selecting auxiliary documents and the rescoring pass using interpolated LM probabilities given by equation 1. More details related to each module are reported below.

2.1. Acoustic data selection for training

For acoustic model (AM) training, domain specific acoustic data were exploited. Recordings of TED talks released before the cut-off date, 31 December 2010, were downloaded with the corresponding subtitles which are content-only transcriptions of the speech. In content-only transcriptions anything irrelevant to the content is ignored, including most non-verbal sounds, false starts, repetitions, incomplete or revised sentences and superfluous speech by the speaker. The collected data consisted in 820 talks, for a total duration of ≈216 hours, with ≈166 hours of actual speech. The provided subtitles are not a verbatim transcription of the speeches, hence a lightly supervised training procedure was applied to extract segments that can be deemed reliable. The approach is that of selecting only those portion in which the human transcription and an automatic transcription agree (see [9, 11] for the details). This procedure has allowed to make available 87% of the training speech, and this amount was considered satisfactory.

2.2. Acoustic model

13 Mel-frequency cepstral coefficients, including the zero order coefficient, are computed every 10ms using a Hamming window of 20ms length. First, second and third order time derivatives are computed, after segment-based cepstral mean subtraction, to form 52-dimensional feature vectors. Acoustic features are normalized and HLDA projected to obtain 39-dimensional feature vectors as described below.

AMs were trained exploiting a variant of the speaker adaptive training method based on Constrained Maximum Likelihood Linear Regression (CMLLR) [12]. In our training variant [13, 10] there are two sets of AMs, the target models and the recognition models. The training procedure makes use of an affine transformation to normalize acoustic features on a cluster by cluster basis (a cluster contains all of the speech segments belonging to a same speaker, according to the given manual segmentation) with respect to the target models. For each cluster of speech segments, an affine transformation is estimated through CMLLR [12] with the aim of minimizing the mismatch between the cluster data and the target models. Once estimated, the affine transformation is applied to cluster data. Recognition models are then trained on normalized data. Leveraging on the possibility that the structure of the target and recognition models can be determined independently, a Gaussian Mixture Model (GMM) can be adopted as target model for training AMs used in the first decoding pass [13]. This has the advantage that, at recognition time, word transcriptions of test utterances are not required for estimating feature transformations. Instead, target models for training recognition models used in the second decoding pass are usually triphones with a single Gaussian per state.

In the current version of the system, a projection of the acoustic feature space, based on Heteroscedastic Linear Discriminant Analysis (HLDA), is embedded in the feature extraction process as follows. A GMM with 1024 Gaussian components is first trained on an extended acoustic feature set consisting of static acoustic features plus their first, second and third order time derivatives. Acoustic observations in each, automatically determined, cluster of speech segments, are then normalized by applying a CMLLR transformation estimated w.r.t. the GMM. After normalization of training data, an HLDA transformation is estimated w.r.t. a set of state-tied, cross-word, gender-independent triphone HMMs with a single Gaussian per state, trained on the extended set of normalized features. The HLDA transformation is then applied to project the extended set of normalized features in a lower dimensional feature space, that is a 39-dimensional feature space.

Recognition models used in the first and second decoding passes are trained from scratch on normalized, HLDA projected, features. HMMs for the first decoding pass are trained through a conventional maximum likelihood procedure. Recognition models used in the second decoding pass are speaker adaptively trained exploiting, as seen above, as target-models triphone HMMs with a single Gaussian density per state.

2.3. Baseline LM

As previously mentioned the text data used for training the baseline LM are extracted from ”google-news” web corpus. These data are grouped into 7 broad domains (economy, sports, science and technology, etc) and, after cleaning, re-

moving double lines and application of a text normalization procedure, the corpus results into about 5.7M of documents, for a total of about 1.6G of words. The average number of words per document is 272.

On this data we trained a 4-gram backoff LM using the modified shift beta smoothing method as supplied by the IRSTLM toolkit [14]. The LM results into about 1.6M unigrams, 73M bigrams, 120M 3-grams and 195M 4-grams.

As seen above the LM is used twice: the first time to compile a static Finite State Network (FSN) which includes LM probabilities and lexicon for the first two decoding passes. The LM employed for building this FSN is pruned in order to obtain a network of manageable size, resulting in a recognition vocabulary of 200K words, 37M bigrams, 34M 3-grams, 38M 4-grams. The non-pruned LM is instead combined (through equation 1) with the auxiliary LMs and used in the third decoding step to rescore word graphs.

2.4. Word graphs generation and rescoring

Word graphs (WGs) are generated in the second decoding step. To do this, all of the word hypotheses that survive inside the trellis during the Viterbi beam search are saved in a word lattice containing the following information: initial word state in the trellis, final word state in the trellis, related time instants and word log-likelihood. From this data structure and given the LM used in the recognition steps, WGs are built with separate acoustic likelihood and LM probabilities associated to word transitions. To increase the recombination of paths inside the trellis and consequently the densities of the WGs, the so called word pair approximation [15] is applied. In this way the resulting graph error rate was estimated to be around $\frac{1}{3}$ of the corresponding WER.

As shown if figure 1, for each given i^{th} talk an auxiliary LM (LM^i_{aux}) is trained using data selected automatically from a huge corpus (i.e. "google-news") with one of the methods described in section 3. The i^{th} query document used to score the corpus consists of the 1-best output of the second ASR decoding step, as depicted in Figure 1. Then, the original (baseline) LM probability on each arc of each WG is substituted with the interpolated probability given by equation 1. The interpolation weights, λ^i_{base} and λ^i_{aux}, associated to the two LMs (LM_{base} and LM^i_{aux}) are estimated so as to minimize the overall LM perplexity on the 1-best output (the same used to build the i^{th} query document), of the second ASR decoding step. For clarity reasons this latter procedure is not explicitly shown in Figure 1.

Finally, the rescored 1-best word sequences are used for evaluating the performance.

3. Auxiliary Data Selection

In this section we describe the processes for selecting documents (rows in "google-news" corpus, each one containing a news article) which are semantically similar to a given automatically transcribed document. In the following, N is the number of total rows of the corpus (5.7M for this work) and D is the total number of unique words in the corpus.

The result of this process is to obtain a sorted version of the whole "google-news" corpus according to similarity scores. The most similar documents will be used to build talk-dependent auxiliary LMs, trained on different amount of data.

3.1. TFxIDF based method

We are given a dictionary of terms $t_1, \ldots, t_D$ derived from the corpus to select (i.e. "google-news").

From the sequence of automatically recognized words $W^i = w^i_1, \ldots, w^i_{\texttt{len}(W^i)}$ of the given i^{th} query document (i.e. the i^{th} automatically transcribed talk) the TFxIDF coefficients $c^i[t_d]$ are evaluated for each dictionary term t_d as follows [16]:

$$c^i[t_d] = (1 + \log(tf^i_d)) \times \log(\frac{D}{df_d}) \qquad 1 \leq d \leq D \quad (2)$$

where tf^i_d is the frequency of term t_d inside document W^i and df_d is the number of documents in the corpus to select that contain the term t_d.

The TFxIDF coefficients of the n^{th} row (document) in the "google-news" corpus $r^n[t_d], 1 \leq n \leq N$ are computed in the same way (where N is the total number of rows). Then, the two vectors $\mathbf{C}^i = c^i[t_1], \ldots, c^i[t_D]$ and $\mathbf{R}^n = r^n[t_1], \ldots, r^n[t_D]$ are used to estimate a similarity score for the n^{th} document via scalar product:

$$s(\mathbf{C}^i, \mathbf{R}^n) = \frac{\mathbf{C}^i \cdot \mathbf{R}^n}{\mid \mathbf{C}^i \mid\mid \mathbf{R}^n \mid} \quad (3)$$

The approach requires to evaluate N scalar products for each automatically transcribed talk. Each scalar product wants, according to equation above, to essentially compute Q^i_n sums plus Q^i_n multiplications, where Q^i_n is the number of common terms in W^i and in the n^{th} document, $W^n_{\texttt{google-news}}$. Hence, the total number of arithmetic operations required for scoring the whole corpus is proportional to $O(2 \times N \times \texttt{E}[Q^i_n])$, where $\texttt{E}[]$ denotes expectation. Concerning memory occupation, the method basically requires to load into memory of the computer the IDF coefficients, i.e. the term $\log(\frac{D}{df_d})$ in equation 2, of all words in the dictionary, plus the $r^n[t_d]$ coefficients, for a total of $D + N \times \texttt{E}[Q^i_n]$ float values. Then, TFxIDF coefficients of the query document are estimated through equation 2, while TFxIDF coefficients of each row of "google-news" are conveniently computed in a preliminary step and stored in a file. In our implementation, access to coefficients entering the scalar product of equation 3 is done using associative arrays. Note that we don't consider this contribution in the complexity evaluation of the approach.

Note also that sorting the whole corpus according to the resulting TFxIDF scores, to find out the most similar doc-

uments to the given query document talk, may be computationally expensive. Hence, we discard documents of the corpus whose TFxIDF scores are below a threshold and perform sorting only on the remaining set of documents. The latter threshold is determined through preliminary analyses of TFxIDF values, taking advantage from the fact that TFxIDF coefficients are normalized within the interval $[0-1]$.

3.2. New proposed approach

3.2.1. Preprocessing stage

First, we build a table containing all the different words found in the "google-news" corpus, each one with an associated counter of the related number of occurrences in the corpus itself. The words are sorted in descending order with respect to the counter and a list is built that includes only the most frequent D' words (in our case a choice of $D' = 200773$ allows to retain words having more than 34 occurrences). Then, from the resulting list the most frequent $D" = 100$ words are removed, allowing to create an index table, where each index is associated to a word in a dictionary $\mathcal{V}$ (lower indices correspond to words having higher counters). Finally, every word in the corpus is replaced with its corresponding index in $\mathcal{V}$. Words outside $\mathcal{V}$ are discarded. Indices of each row are then sorted to allow quick comparison (this point will be discussed later).

The rationale behind this approach is the following:

- very common words, i.e. those with low indices, only carry syntactic information, therefore they are useless if the purpose is to find semantically similar sentences;
- very uncommon words will be used rarely so they will just slow down the search process.

The choice for the reported values of D' and D'' has been done on the basis of preliminary experiments carried out on a development data set (see section 4) and resulted not to be critical. With the chosen values about half of the words of the corpus were discarded: currently there are 5.7 millions rows, corresponding in total to 1561.1 millions words, 864.5 millions survived indices. We keep alignment between the original corpus and its indexed version.

3.2.2. Searching stage

We apply to the given i^{th} talk the same procedure as before, obtaining a sequence of numerically sorted word indices. Hence, as for the TFxIDF method, both the i^{th} talk and the n^{th} "google-news" document are represented by two vectors (containing integer indices in this case): $\mathbf{C}'^i$ and $\mathbf{R}'^n$, respectively. The similarity score is in this case:

$$s'(\mathbf{C}'^i, \mathbf{R}'^n) = \frac{e(\mathbf{C}'^i, \mathbf{R}'^n)}{\mathtt{dim}(\mathbf{C}'^i) + \mathtt{dim}(\mathbf{R}'^n)} \quad (4)$$

where $e(\mathbf{C}'^i, \mathbf{R}'^n)$ is the number of common indices between the two vectors $\mathbf{C}'^i$ and $\mathbf{R}'^n$.

Note that, differently from TFxIDF approach, where both vectors $\mathbf{C}^i$ and $\mathbf{R}^n$ can be assumed to have dimension equal to D (the size of the dictionary), in this case the normalization term for the similarity measure is given by the denominator of equation 4. The two vectors $\mathbf{C}'^i$ and $\mathbf{R}'^n$ have dimensions exactly equal to the number of the corresponding indexed words survived after pruning of dictionary, as explained above.

Note also that, while TFxIDF method allows to compare two documents by weighting same words both with their frequencies and with their relevance in the documents to select, the proposed approach is essentially a method to count the number of same words in the documents (word counters are not used in the similarity metric). However, since components of index vectors are numerically ordered, the computation of the similarity score $s'(\mathbf{C}'^i, \mathbf{R}'^n)$ results very efficient. This is essential given the large number of documents in the corpus to score.

Each of the N score computation, according to equation 4, essentially needs Q'^i_n comparisons (in this case no sums or multiplications are executed) to be executed, with $Q'^i_n \leq Q^i_n$, due to dictionary pruning. Since, we can assume $\mathtt{E}[Q'^i_n] \simeq \frac{1}{2}\mathtt{E}[Q^i_n]$ (due to halving of indices), the total number of comparisons required for scoring the whole corpus is proportional to $O(\frac{N}{2} \times \mathtt{E}[Q^i_n])$, i.e. $\frac{1}{4}$ with respect to TFxIDF based method. In addition, differently from the latter one, the proposed approach doesn't require to load into memory of the computer any parameter related to the whole dictionary, instead only the sequence of indices (i.e. one sequence of integer values for each row of "google-news") entering equation 4 is needed. In our implementation the latter indices are conveniently stored and read from a file. Therefore, the memory requirements of the proposed approach are negligible. Furthermore, since the resulting document scores are not normalized, the estimate of the threshold to be used for selecting the subset of the documents to sort from the whole corpus is based on a preliminary computation of a histogram of scores.

Finally, in order to measure the complexities of proposed method and TFxIDF based one, we led three different selection runs using ASR output of a predefined TED talk. For processing the whole "google-news" corpus the proposed method took on average about $16min$, with a memory occupation of about 10MB, while the TFxIDF based method took on average about $114min$, with a memory occupation of about 650MB. These runs were carried out on the same Intel/Xeon E5420 machine, free from other computation loads.

3.3. Perplexity based method

A 3-gram LM is trained with the automatic transcription of the given i^{th} TED talk. Then, the perplexity of each document in the "google-news" corpus is estimated using this latter LM and the resulting perplexity values are used to find out the most similar documents to the given talk. Also in this case an histogram of perplexity scores is es-

timated to determine the optimal selection threshold before sorting documents. Basically, each of the N perplexity values (one for each "google-news" document) requires to compute $\texttt{len}(W^n_{\texttt{google-news}})$ log-probabilities (through LM look-up table and LM backoff smoothing) and $\texttt{len}(W^n_{\texttt{google-news}})$ sums.

4. Experiments and results

As previously mentioned experiments have been carried out on the evaluation sets of IWSLT 2011 evaluation campaign. In total, these latter ones include 27 talks, which have been divided into a development set and a test set. Table 1 reports some statistics derived from evaluation sets.

Table 1: *Statistics related to the dev/test sets of IWSLT 2011 evaluation campaign: total number of running words, minimum, maximum and mean number of words per talk.*

	dev-set (19 talks)	**test-set (8 talks)**
#words	44505	12431
(min,max,mean)	(591,4509,2342)	(484,2855,1553)

Note the quite small number of words available for each talk to build the similarity models to be used in the automatic selection process, especially for the test set. Despite this fact, significant performance improvement has been achieved on this task.

We evaluated performance, both in terms of PP and WER.

The overall perplexity PP_{dev} on the dev set is computed summing the LM log-probabilities of each reference talk and dividing by the total number of words, according to the following equation:

$$PP_{dev} = 10^{\frac{\sum_{i=1}^{i=19} -\log_{10}(P^i_{LM}[W_i])}{NW}} \quad (5)$$

where $P^i_{LM}[W_i]$ is the probability of the reference word sequence in the i^{th} talk, computed using the i^{th} talk-dependent interpolated LM, and NW is the total number of words in the dev set. The overall perplexity on the test set is computed in a similar way.

Performance, as a function of the number of words used to train the auxiliary LMs, are reported in Figures 2 to 5, for both dev set and test set.

In the figures the point corresponding to 0 words on the abscissa indicates performance obtained using the baseline, talk independent, LM (i.e. no interpolation with auxiliary LMs has been made).

As can be observed all of automatic selection methods allow to improve both in terms of perplexity and WER. Looking at curves of perplexity (figures 2 and 4), we note that an optimal value for the number of words that should be used for training auxiliary LM is clearly reached with both TFxIDF and new proposed selection approach (the related curves exhibit clear minimal points). Instead, this trend is not exhibited by PP based curves, where the minimal perplexity value seems will be reached with a quite high number of auxiliary words (we deserve to extend the curves with future experiments). This is probably due to the fact that proposed and TFxIDF selection methods give more weight to content words than the PP based one, where also functional words can significantly contribute to form the scores of documents to select.

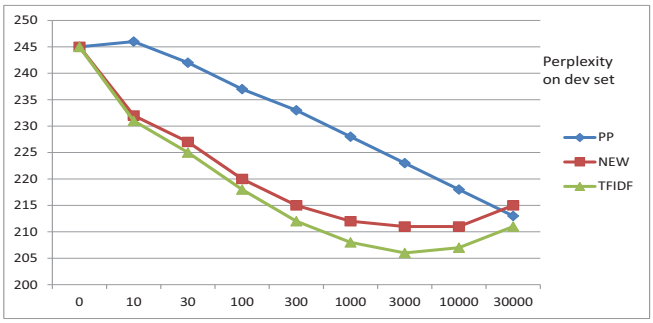


Figure 2: *Perplexity on dev set of PP-based selection method, NEW proposed method and TFxIDF based method as a function of the number of words, shown on a logarithmic scale, used to train the auxiliary LMs.*

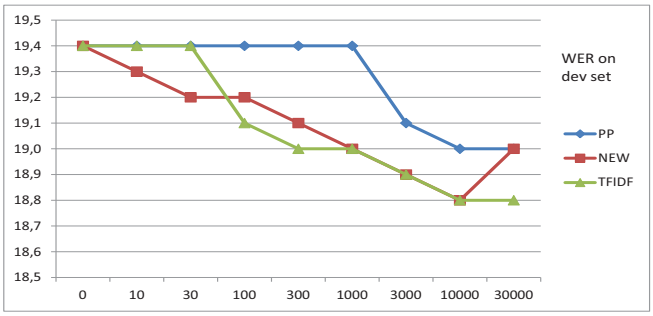


Figure 3: *%WER on dev set for the various selection methods.*

A different trend is instead observed looking at curves related to WERs (see figure 3 and 5), specifically, they do not exhibit clear minimal values. Actually, while perplexity values depend only on LM probabilities (i.e. on models derived only from text data, including the selected ones), WER values are obtained through Maximum a Posteriori (MAP) decoding, combining LM probability scores and AM likelihood scores, giving rise to more irregularities in the related curves, as well as to local minima. In any case, it is important to note that the usage of focused LMs allow always to decrease WER. In particular, both new and TFxIDF approaches allow to achieve about 3% WER reduction on both

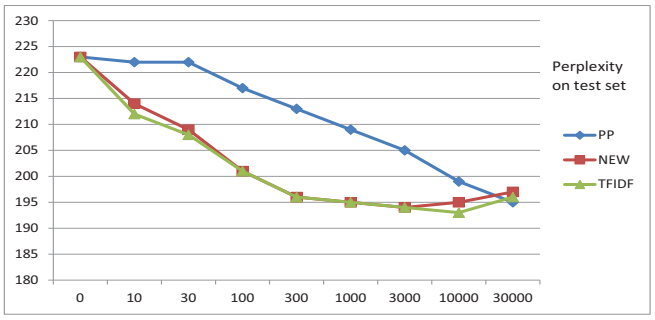


Figure 4: *Perplexity on test set for the various selection methods.*

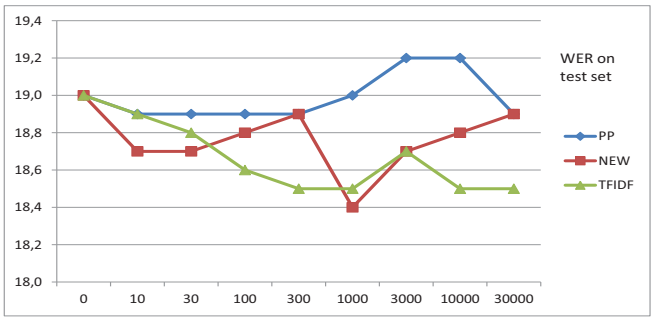


Figure 5: *%WER on test set for the various selection methods.*

dev and test sets, while a lower improvement (around 2% relative WER reduction) is obtained with the PP based selection method.

Finally, for comparison purposes we trained a domain specific LM using subtitles of TED talks that have been downloaded from the internet by the organizer of IWSLT 2011 evaluation campaign, before the cut-off date (December 31, 2010), and distributed to the participants. The latter domain specific corpus contains around 2M words and the resulting LM (LM_{ted}) contains about: 40K unigrams, 540K bigrams, 1.6M 3-grams and 1M 4-grams. Then, we have linearly interpolated LM_{ted} with the baseline LM (LM_{base}) and, as for the automatic selection methods, we have rescored the WGs generated in the second ASR decoding pass with the "adapted" LM (i.e. $LM_{base} \oplus LM_{ted}$, where symbol $\oplus$ denotes interpolation according to equation 1). Note that also in this case the linear interpolation weights have been estimated using the automatic transcriptions of the second ASR decoding pass. Table 2 reports the performance, both in terms of WER and PP, for the "focused" LMs (where LM_{pp}, $LM_{tf \cdot idf}$ and LM_{new} have been trained on automatically selected text corpora of 3M words) and for the domain adapted LM.

Table 2: *Results obtained using "focused" LMs and domain adapted LM.*

	dev-set		test-set	
	PP	**%WER**	**PP**	**%WER**
$LM_{base} \oplus LM_{pp}$	223	19.0	205	18.9
$LM_{base} \oplus LM_{new}$	210	18.8	194	18.4
$LM_{base} \oplus LM_{tf \cdot idf}$	206	18.8	194	18.5
$LM_{base} \oplus LM_{ted}$	158	18.7	142	18.4

As can be seen from the Table, although PP values for the domain adapted LM ($LM_{base} \oplus LM_{ted}$) are significantly lower with respect to the other LMs, the corresponding WER values are similar to those obtained with focused LMs. The proposed selection approach (row $LM_{base} \oplus LM_{new}$), gives 0.1% difference on the dev set and 0% on test set, respectively.

4.1. Experiments with IWSLT2012 data

To further check the effectiveness of LM focusing approaches described so far, we carried out additional experiments using the sets of English text corpora distributed for IWSLT 2012 Evaluation Campaign. These latter consist of: news commentaries and news crawls, proceedings of European Parliament sessions and the newswire text corpus Gigaword (fifth edition), as distributed by the LDC consortium (see LDC catalog http://www.ldc.upenn.edu/Catalog/ for more details about this corpus). In addition an in-domain text corpus containing transcriptions of TED talks has been provided.

With these data we built 3 LMs:

- LM_{W12}, trained on news commentaries/crawls and European Parliament proceedings (about 830M words);
- LM_{G5}, trained on Gigaword, fifth edition (about 4G words);
- LM_{T12}, trained on in-domain TED data (about 2.7M words).

Similarly to what reported in Table 2 we measured performance (both PP and WER) using talk-specific linearly interpolated LMs. In particular, we compared performance using different combinations of LMs, as shown on Table 3.

Also in this case talk-specific auxiliary LMs were trained on data (5M words) automatically selected using the ASR output of the second decoding step. The latter selection was carried out over both $W12$ and $G5$ text corpora (i.e. without using in-domain TED data). We only compared TFxIDF based method and the new one, proposed in this paper.

Table 3 gives the results on both development and test sets. In this case we haven't evaluated performance as a function of the number of words retained for auxiliary data selection (see figures 2 to 5). This latter number of words, according to previous experiments using IWSLT 2011 text data, has

Table 3: *Results obtained using baseline, "focused" and domain adapted LMs trained on text data delivered for IWSLT 2012 Evaluation Campaign.*

	dev-set		test-set	
	PP	%WER	PP	%WER
$LM_{W12} \oplus LM_{G5}$	179	18.8	159	18.1
$LM_{W12} \oplus LM_{G5} \oplus LM_{tf \cdot idf}$	155	18.4	140	17.6
$LM_{W12} \oplus LM_{G5} \oplus LM_{new}$	164	18.5	146	17.5
$LM_{W12} \oplus LM_{G5} \oplus LM_{ted}$	139	18.2	126	17.5

been fixed to 5 millions. Note also that with the new set of training text data the improvement given by the proposed focusing procedure is maintained (about 2% relative WER reduction on the dev set and about 3% WER relative reduction on the test set), performing very closely to domain adapted LMs.

5. Conclusions and Future Work

We have described a method for focusing LMs towards the output of an ASR system. The approach is based on the useful and efficient selection, according to a novel similarity score, of documents belonging to large sets of text corpora on which the LM used for automatic transcription was trained. Improvements on WER have been reached without making use of in-domain specific text data. In addition, comparisons with TFxIDF and PP based selection methods have been done, showing the effectiveness of the proposed approach, which resulted computationally less expensive than TFxIDF.

However, at present we are not able to decide if this result is quite general, or if it depends on the particular set of data used, or on the specific TED domain. Future works will try to extend the approach to domains different from TED.

6. References

[1] J. Gao, J. Goodman, M. Li, and K. Lee, "Toward a Unified Approach to Statistical Language Modeling for Chinese," *ACM Transactions on Asian Language Information Processing*, vol. 1, no. 1, pp. 3–33, 2002.

[2] D. Klakow, "Selecting Articles from the Language Model Training Corpus," in *Proc. of ICASSP*, Istanbul, Turkey, June 2000, pp. 1695–1698.

[3] R. C. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *ACL Conference*, Uppsala, Sweden, July 2010, pp. 220–224.

[4] S. Maskey and A. Sethy, "Resampling Auxiliary Data for Language Model Adaptation in Machine Translation for Speech," in *Proc. of ICASSP*, Taipei,Taiwan, April 2009, pp. 4817–4820.

[5] G. Lecorve, J. Dines, T. Hain, and P. Motlicek, "Supervised and unsupervised Web-based language model domain adaptation," in *Proc. of INTERSPEECH*, Portland, USA, September 2012.

[6] K. Thadani, F. Biadsy, and D. M. Bikel, "On-the-fly Topic Adaptation for You Tube Video Transcription," in *Proc. of INTERSPEECH*, Portland, USA, September 2012.

[7] X. Liu, W. Byrne, M. Gales, A. de Gispert, M. Tomalin, P. Woodland, and K. Yu, "Discriminative Language Model Adaptation for Mandarin Broadcast Speech Transcription," in *Proc. of ASRU*, Kyoto, Japan, December 2007, pp. 153–158.

[8] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. Cernocky, "Empirical Evaluation and Combination of Advanced Language Modeling Techniques," in *Proc. of INTERSPEECH*, Florence, Italy, August 2011, pp. 605–608.

[9] N. Ruiz, A. Bisazza, F. Brugnara, D. Falavigna, D. Giuliani, S. Jaber, R. Gretter, and M. Federico, "FBK@IWSLT 2011," in *Proc. of IWSLT workshop*, San Francisco, USA, December 2011.

[10] D. Giuliani and F. Brugnara, "Experiments on Cross-System Acoustic Model Adapatation," in *ASRU Workshop 2007*, Kyoto, Japan, Dec. 2007, pp. 117–122.

[11] J. Loof, D. Falavigna, R. Schluter, D. Giuliani, R. Gretter, and H. Ney, "Evaluation of Automatic Transcription Systems for the Judicial Domain," in *Proc. of IEEE SLT workshop*, San Francisco, USA, December 2010, pp. 194–199.

[12] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[13] G. Stemmer, F. Brugnara, and D. Giuliani, "Using Simple Target Models for Adaptive Training," in *Proc. of ICASSP*, vol. 1, Philadelphia, PA, March 2005, pp. 997–1000.

[14] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models," in *Proc. of INTERSPEECH*, Brisbane, Australia, September 2008, pp. 1618–1621.

[15] X. Aubert and H. Ney, "A word graph algorithm for large vocabulary continuous speech recognition," in *Proc. of ICSLP*, 1994, pp. 1355–1358.

[16] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," in *First International Conference on Machine Learning*, New Brunswick:NJ, USA, 2003.

Simulating Human Judgment in Machine Translation Evaluation Campaigns

Philipp Koehn

School of Informatics
University of Edinburgh
pkoehn@inf.ed.ac.uk

Abstract

We present a Monte Carlo model to simulate human judgments in machine translation evaluation campaigns, such as WMT or IWSLT. We use the model to compare different ranking methods and to give guidance on the number of judgments that need to be collected to obtain sufficiently significant distinctions between systems.

1. Introduction

An important driver of current machine translation research are annual evaluation campaigns where research labs use the latest prototype of their system to translate a fixed test set, which is then ranked by human judges. Given the nature of the translation problem, where everybody seems to disagree on what the right translation of a sentence is, it comes of no surprise that the methods used to obtain human judgments and rank different systems against each other is also under constant debate.

This paper presents a Monte Carlo simulation that closely follows the current practice in the evaluation campaigns carried out for the Workshop on Statistical Machine Translation (WMT [1]), the International Workshop on Spoken Language Translation (IWSLT [2]), and to a lesser degree, since it mostly relies on automatic metrics, the Open Machine Translation Evaluation organized by NIST (OpenMT[1]).

The main questions we answer are: How many judgments do we need to collect to reach a reasonably definitive statement about the relative quality of submitted systems? Are we ranking systems the right way? How do we obtain proper confidence bounds for the rankings?

2. Related Work

While manual evaluation of machine translation systems has a rich history, most recent evaluation campaigns and lab-internal manual evaluations restrict themselves to a ranking task. A human judge is asked, if, for a given input sentence, she prefers output from system A over output from system B.

While this is a straight-forward procedure, the question how to convert these pairwise rankings into an overall ranking of several machine translation systems has recently received attention. Bojar et al. [3] critiqued the ongoing practice in the WMT evaluation campaigns, which was subsequently changed. Lopez [4] proposed an alternative method to rank systems. We will discuss these methods in more detail below.

An intriguing new development in human involvement in the evaluation of machine translation output is HyTER [5]. Automatic metrics suffer from the fact that a handful of human reference translations cannot expected to be matched by other human or machine translators, even if the latter are perfectly fine translations. The idea behind HyTER is to list *all* possible correct translations in the compact format of a recursive transition network (RTN). These networks are constructed by a human annotator who has access to the source sentence. Machine translation output is then matched against this network using string edit distance, and the number of edits is used as a metric.

Construction of the networks takes about 1–2 hours per sentence. This cost is currently too expensive for evaluations such as WMT with its annually renewed test set and eight language pairs. But we are hopeful that technical innovations, for instance in automatic paraphrasing, will bring down this cost to make it a more viable option in machine translation evaluation campaigns.

3. Model

We now define a model which consists of machine translation systems that produce translations of randomly distributed quality. We will make design decisions and set the only free parameter (the standard deviation of the systems' quality distributions) to match statistics from the actual data of the WMT evaluation campaign.

In an evaluation, n **systems** $S = \{S_1, ...S_n\}$ participate. Each system produces translations with the average **quality** μ_n. When simulating an evaluation **experiment**, the quality μ_n of each system is chosen from a uniform distribution over the interval [0;10]. So, an experiment is defined by a list of average system qualities $E = (\mu_1, ...\mu_n)$.

Note: The range of the interval is chosen arbitrarily — the actual quality scores do not matter, only the relative scores of different systems. We use the uniform distribution to chose system qualities (opposed to, say, normal distribu-

[1] http://www.nist.gov/itl/iad/mig/openmt.cfm

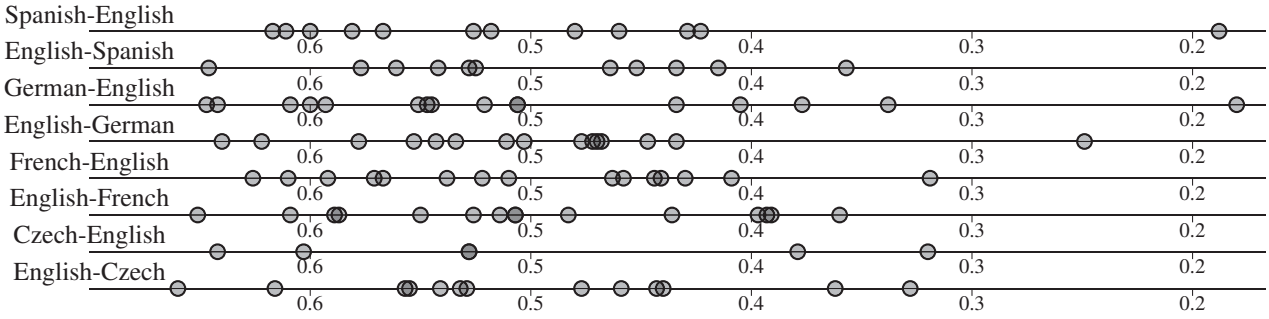


Figure 1: Win ratios of the systems in the WMT12 evaluation campaign. Except for the occasional outlier at the low end, the systems follow roughly a uniform distribution. For details on the computation of the win ratios see Section 4.3, our experiments show that uniformly distributed average system qualities lead to uniformly distributed win ratios.

tion) because this reflects the data from the WMT evaluation campaigns (see Figure 1).

In each evaluation experiment E, a sample of human judgments J_E is drawn. We follow here the procedure of the WMT evaluation campaign: We randomly select sets of 5 different systems $F_{E,i} = \{s_a, s_b, s_c, s_d, s_e\}$ with $1 \leq a, b, c, d, e \leq n$. Each system $j \in F_{E,i}$ produces a translation for the same input sentence, with a translation quality $q_{E,i,j}$ that is chosen from a normal distribution: $\mathcal{N}(\mu_j, \sigma^2)$. Based on this set of translations, we extract a set of 10 ($= \frac{5\times4}{2}$) pairwise rankings $\{(j_1, j_2)|q_{E,i,j_1} > q_{E,i,j_2}\}$ and add them to the sample of human judgments J_E.

Note:

- The variance σ^2 is the same for all systems. We discuss at the end of this section how the value of the variance is set.
- This procedure may appear unnecessarily complex. We could have just picked two systems, draw translation qualities q_{i,s_j} for each, compare them, and add a pairwise ranking to the judgment sample J_E. However, the WMT evaluation campaign follows the described procedure, because comparing a set of 5 systems at once yields 10 pairwise rankings faster then comparing 2 systems at a time, repeated 10 times. It is an open question, if the procedure adds distortions, so we match it in our model.
- The WMT evaluation campaign allows for ties. We ignore this in our model, since it adds an additional parameters (ratio of ties) that we would have to set. It is worth investigating, if allowing for ties changes any of our findings.
- Since it is not possible to tease apart the quality of the system and the perceived quality of a system by a human judge, we do not model the noise introduced by human judgment.

We still have to set the variance σ^2 which is used to draw translation quality scores q for a translation systems S_j with the average quality of μ_j. We base this number on the ratio of system pairs that we can separate with statistically significance testing, as follows:

Given the sample of human judgments in form of pairwise system rankings $J_E = ((a_1, b_1), (a_2, b_2), ...))$ with $1 \leq a_i, b_i \leq n, a_i \neq b_i$, we can count how many times a system S_j **wins** over another system S_k in pairwise rankings: $\mathit{win}(S_j, S_k) = |((a_i, b_i) \in J_E | a_i = j, b_i = k)$ — and how many times it **loses**: $\mathit{loss}(S_j, S_k) = 1 - \mathit{win}(S_k, S_j)$. Given these two numbers, we can use the sign test to determine if system S_j is statistically significantly better (or worse) than system S_k at a desired p-level (we use p-level=0.05).

The more human judgments we have, the more systems we can separate. Figure 2 plots the ratio of system pairs (out of $\frac{n(n-1)}{2}$) that are different according to the sign test against the number of pairwise judgments for all 8 language pairs of the WMT12 evaluation campaign. The variance for our model, chosen to match these curves, ranges from 7 to 12.

4. Ranking Methods

There are several ways to use the (actual or simulated) pairwise judgment data J_E to obtain assessments about the relative quality of the systems participating in a given evaluation campaign. We already encountered one such assessment: the statistically significantly better quality of one system over another another at a certain p-level according to the sign test. These assessments are reported in large tables in the WMT12 overview paper, but are somewhat unsatisfying because many system pairs are reported as not statistically significantly different.

Instead, we would like to report rankings of the systems. In this section, we will review two ranking methods proposed for this task, introduce a third one, and use our model to assess how often these ranking methods err.

4.1. Bojar

In the recent 2012 WMT evaluation campaign, systems were ranked by the ratio of how often they were ranked better or equal to any of the other systems. Following the argument

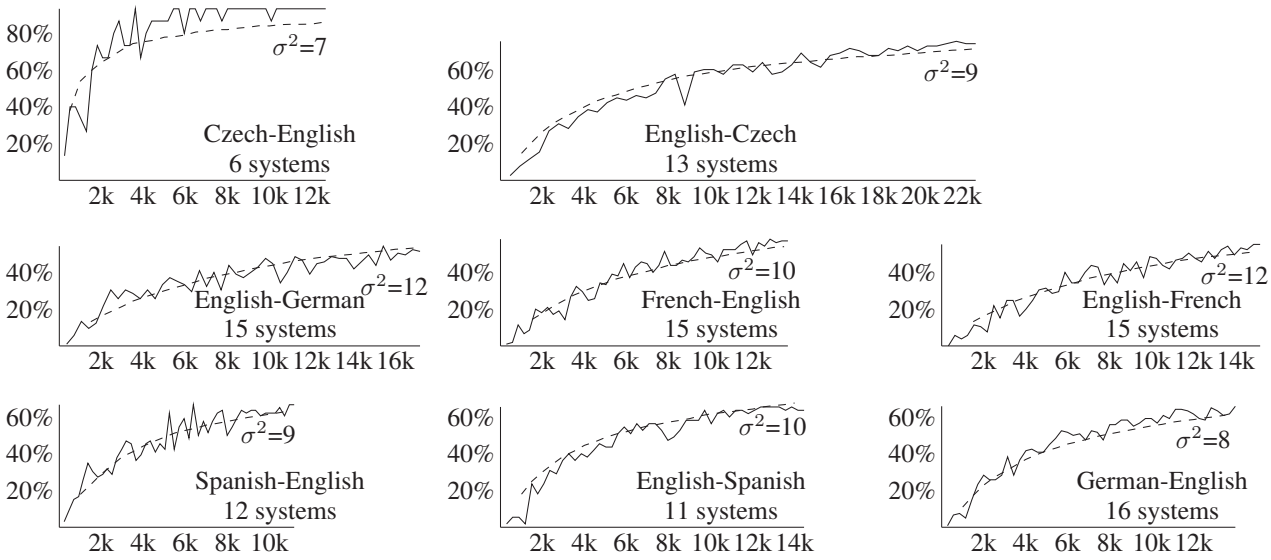


Figure 2: Ratio of system pairs that are statistically different according to the sign test with increased number of human judgments in the form of pairwise rankings. The graphs plot the actual ratio (solid lines) for data from the WMT12 evaluation campaign against the ratio (dashed lines) obtained from running our simulation with a translation quality variance σ^2. The variance is set to an integer to match the actual ratio as closely as possible. Higher variance and more systems cause slower convergence. Higher variance implies that the systems have more similar average quality.

of Bojar et al. [3], this ignores ties and uses the definition of wins and loss as defined above, to compute a ranking score:

$$\text{score}(S_j) = \frac{\sum_{k,k\neq j} \text{win}(S_j, S_k)}{\sum_{k,k\neq j} \text{win}(S_j, S_k) + \text{loss}(S_j, S_k)} \quad (1)$$

Systems were ranked by this number. This ranking method was used for the official ranking of WMT 2012. We refer to it here as BOJAR.

4.2. Lopez

Lopez [4] argues against using aggregate statistics over a set of very diverse judgments. Instead, a ranking that has the least number of pairwise ranking violations is said to be preferred. He defines a count function for pairwise order violations

$$\text{score}(S_j, S_k) = \max(0, \text{win}(S_j, S_k) - \text{loss}(S_j, S_k)) \quad (2)$$

Given a bijective ranking function $R(j) \rightarrow j'$ with $j, j' \in \{1, ..., n\}$ the total number of pairwise ranking violations is defined as

$$\text{score}(R) = \sum_{j,k|R(S_j)<R(S_k)} \text{score}(S_j, S_k) \quad (3)$$

Finding the optimal ranking R that minimizes this score is not trivial, but given the number of systems involved in this evaluation campaign, it is manageable.

4.3. Expected Win

In BOJAR, systems are put at an disadvantage, if they are compared more frequently against good systems than against bad systems. We can overcome this by first computing the win ratios between each system pair and then averaging the ratios:

$$\text{score}(S_j) = \frac{1}{n} \sum_{k,k\neq j} \frac{\text{win}(S_j, S_k)}{\text{win}(S_j, S_k) + \text{loss}(S_j, S_k)} \quad (4)$$

This score can also be understood as the expectation of a win against a randomly chosen opponent system.

4.4. Evaluation

The three methods above have been justified with an appeal to intuition. But now, with the model that we introduced in Section 3, we are able to run simulations that start with a gold standard ranking based on the systems' average translation scores μ_i, generate judgment data, apply the ranking methods, and then check the obtained rankings according to the methods against the gold standard ranking.

We chose an experimental setup that reflects a typical situation in the WMT evaluation campaign, with $n = 15$ systems and variance $\sigma^2 = 10$. We randomly draw 10,000 experiments, sample human judgments for each and rank the systems based on the methods discussed in this section (BOJAR, LOPEZ, EXPECTED). We evaluate the rankings R_m obtained by each method m against the gold standard ranking R by computing the ratio of system pairs where the worst

Judgments	Pairwise Method				Bootstrap Method			
$\|J_E\|$	range size	violations	clusters	violations	range size	violations	clusters	violations
10,000	8.1	0.8%	1.0	0%	4.6	3.4%	1.8	0.5%
20,000	6.3	0.8%	1.1	0%	3.7	2.4%	3.0	0.5%
30,000	5.4	0.7%	1.4	0%	3.3	2.3%	3.9	0.4%
40,000	4.9	0.9%	1.7	0.1%	3.0	2.0%	4.7	0.4%
50,000	4.5	0.9%	2.0	0.1%	2.9	2.1%	5.3	0.7%

Table 1: Quality of the confidence bounds obtained with the pairwise and bootstrap methods (see Section 5.1. The methods allow us to group the systems into clusters of comparable performance and indicate a range for the rank number in the rankings. Experiment with 15 systems, $\sigma^2 = 10$, and p-level 0.05, averaged over 400 runs.

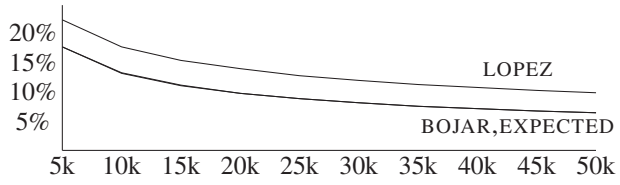


Figure 3: Errors of the different ranking methods discussed in Section 4: Ratio of system pairs where the worst system is ranked better.

system is ranked better.

$$\text{error}(R_m) = \frac{|\{j,k|R_m(S_j) < R_m(S_k), R(S_j) > R(S_k)\}|}{\frac{1}{2}n(n-2)} \quad (5)$$

Figure 3 shows the results of this study. Both BOJAR and EXPECTED perform better than LOPEZ, with an error of 13.2%/13.1% for the first two methods and 17.6% for LOPEZ with 10,000 pairwise rankings, and an error of 6.4% for the first two methods and 17.6% for LOPEZ with 50,000 pairwise rankings.

5. Confidence Bounds

Reporting a definitive ranking hides the uncertainty about it. It is useful to also report, how confident we are that a particular system S_j is placed on rank r_j. In this section, we aim to give this information in two forms:

- by determining the **rank range** $[r'_j, ..r''_j]$ into which the true rank of the system S_j falls with a given level of statistical significance, say, p-level 0.05
- by grouping systems into **clusters**, to which each system belongs with a given level of statistical significance

5.1. Methods

We now present two methods to produce this information, discuss how they can be evaluated, and report on experiments.

The first idea is to rely on the pairwise statistically significant distinctions that we can obtain by the sign test from the data. To give an example, if system S_j is significantly better than $b = 9$ systems, worse than $w = 2$ systems and indistinguishable from $e = 3$ systems, then its rank range is 3–6 (from $w + 1$ to $w + 1 + e$).

The second idea is to apply bootstrap resampling [6]. Given a fixed set of judgments J_E, we sample pairwise rankings from this set (allowing for multiple drawings of the same ranking). We then compute a ranking with the expected win method based on this resampling. We repeat this process a 1000 times, record each time the rank of a system S_j. We then sort the obtained 1000 ranks, chop off the top 25 and bottom 25 ranks and report the minimum interval containing the remaining ranks as rank range.

Clusters are obtained by grouping systems with overlapping rank ranges. Formally, given ranges defined by start(S_j) and end(S_j), we seek the largest set of clusters $\{C_c\}$ that satisfies:

$$\begin{aligned} &\forall S_j \exists C_j : S_j \in C_j \\ &S_j \in C_j, S_j \in C_k \rightarrow C_j = C_k \\ &C_j \neq C_k \rightarrow \forall S_j \in C_j, S_k \in C_k : \\ &\qquad \text{start}(S_j) > \text{end}(S_k) \text{ or } \text{start}(S_k) > \text{end}(S_j) \end{aligned} \quad (6)$$

5.2. Evaluation

We can measure the performance of the confidence bound estimation methods by the tightness of the rank ranges, the number of clusters, and the number of violations for each — a violation happens when the true rank of a system falls outside the rank range or if a system is placed in a cluster with a truly higher ranked system placed into a lower cluster or vice versa.

See Table 1 for results of a experiment with the same settings as above (variance $\sigma^2 = 10$, number of systems $n = 15$). The bootstrap resampling method yields smaller rank range sizes (about half) and a larger number of clusters (2–3 times as many). This does come at the cost of increased error, but note that the measured error is well below the statistical significance p-level of 0.05 used to run the bootstrap. If lower error is desired, smaller p-levels may be used.

Table 2 and 3 show the application of the method to two language pairs of the WMT12 evaluation campaign. In the

Rank	Range	Score	System
1	1	0.660	CU-DEPFIX
2	2	0.616	ONLINE-B
3	3–6	0.557	UEDIN
4	3–6	0.555	CU-TAMCH
5	3–7	0.541	CU-BOJAR
6	4–7	0.532	CU-TECTOMT
7	4–7	0.529	ONLINE-A
8	8–10	0.477	COMMERCIAL1
9	8–11	0.459	COMMERCIAL2
10	9–11	0.443	CU-POOR-COMB
11	9–11	0.440	UK
12	12	0.362	SFU
13	12	0.328	JHU

Table 2: Application of our methods to the WMT12 English–Czech evaluation: The 13 systems are split into 6 clusters. About 22,000 judgments were collected.

first example (English–Czech, $\sigma^2 = 9$, $n = 13$, 22,000 judgments) we see a nice separation into 6 clusters, while in the second example (French–English, $\sigma^2 = 10$, $n = 15$, 13,000 judgments) almost all systems are in the same cluster. Our findings in Table 1 suggest that collecting 30,000 judgments would allowed us to separate the systems into about 4 clusters, with each system ranging over only 3 ranks.

6. How Many Judgements?

A very practical question that we are trying to answer in this paper is: When we run a manual evaluation, how many judgments do we need to collect?

The answer to this questions depends on how many systems participate in the evaluation and the desired level of certainty — the first number is readily available and the second can be chosen at will. But the answer also depends on the variance σ^2 of the systems. This is a number that will become only clearer once a large number of judgments have been collected. The findings from the WMT12 evaluation campaign gives some guidance about the value of σ^2 — numbers between 8 and 12 seem to cover most cases.

Armed with these specifics, Table 4 gives an estimate about the minimum number of judgments required. For instance, for the WMT12 French–English pair ($n = 15, \sigma^2 = 10$), the organizers collected 13,000 judgments. This was sufficient to tell about 70% of pairs apart. To raise that number to 80%, about 40,000 judgments are required.

Note that we computed the number in the table with a grid search over the number of judgments, so all numbers are approximate.

7. Conclusions

We introduced a Monte Carlo model for the simulation of the methodology underlying current machine translation evalu-

Rank	Range	Score	System
1	1–3	0.626	LIMSI
2	1–4	0.610	KIT
3	1–5	0.592	ONLINE-A
4	2–6	0.571	CMU
5	3–7	0.567	ONLINE-B
6	5–8	0.538	UEDIN
7	5–8	0.522	LIUM
8	6–9	0.510	RWTH
9	8–12	0.463	RBMT-1
10	9–13	0.458	RBMT-3
11	9–14	0.444	SFU
12	9–14	0.441	UK
13	10–14	0.430	RBMT-4
14	12–14	0.409	JHU
15	15	0.319	ONLINE-C

Table 3: Compare to Table 2: In this example, only the last system was split off from the main cluster. Only about 13,000 judgments were collected. Our findings suggest that collecting 30,000 judgments would allowed us to break up the systems into about 4 clusters, with each system ranging over only 3 ranks.

n	σ^2	Ratio of significant pairs			
		50%	70%	80%	90%
6	8	1k	4k	8k	30k
6	10	2k	5k	10k	45k
6	12	2k	7k	20k	60k
8	8	2k	6k	14k	60k
8	10	3k	8k	20k	90k
8	12	4k	14k	35k	140k
10	8	4k	10k	25k	100k
10	10	5k	16k	40k	150k
10	12	6k	20k	50k	200k
12	8	5k	15k	35k	140k
12	10	7k	25k	60k	250k
12	12	9k	35k	80k	350k
15	8	8k	25k	50k	200k
15	10	12k	40k	80k	350k
15	12	15k	50k	120k	500k

Table 4: Guidance on how many pairwise judgments must be collected to obtain a certain ratio of statistically significant (p-level 0.05) distinctions for pairs of systems. In the WMT12 campaign 10,000–20,000 judgments were collected.

ation campaigns. We used the model to compare different ranking methods, introduced methods to obtain confidence bounds and give guidance on the number of judgment to be collected to obtain satisfying results. The findings show that recent WMT evaluation campaigns do not collect sufficient judgments and that the number of judgments should be doubled or increased three-fold.

8. Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 287658 (EU BRIDGE) and agreement 288487 (MosesCore).

9. References

[1] C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2012 workshop on statistical machine translation," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montreal, Canada: Association for Computational Linguistics, June 2012, pp. 10–48. [Online]. Available: http://www.aclweb.org/anthology/W12-3102

[2] M. Paul, M. Federico, and S. Stücker, "Overview of the IWSLT 2010 Evaluation Campaign," in *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, M. Federico, I. Lane, M. Paul, and F. Yvon, Eds., 2010, pp. 3–27.

[3] O. Bojar, M. Ercegovčević, M. Popel, and O. Zaidan, "A grain of salt for the wmt manual evaluation," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, July 2011, pp. 1–11. [Online]. Available: http://www.aclweb.org/anthology/W11-2101

[4] A. Lopez, "Putting human assessments of machine translation systems in order," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montreal, Canada: Association for Computational Linguistics, June 2012, pp. 1–9. [Online]. Available: http://www.aclweb.org/anthology/W12-3101

[5] M. Dreyer and D. Marcu, "Hyter: Meaning-equivalent semantics for translation evaluation," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 162–171. [Online]. Available: http://www.aclweb.org/anthology/N12-1017

[6] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

Semi-supervised Transliteration Mining from Parallel and Comparable Corpora

Walid Aransa, Holger Schwenk, Loic Barrault

LIUM, University of Le Mans
Le Mans, France
firstname.lastname@lium.univ-lemans.fr

Abstract

Transliteration is the process of writing a word (mainly proper noun) from one language in the alphabet of another language. This process requires mapping the pronunciation of the word from the source language to the closest possible pronunciation in the target language. In this paper we introduce a new semi-supervised transliteration mining method for parallel and comparable corpora. The method is mainly based on a new suggested Three Levels of Similarity (TLS) scores to extract the transliteration pairs. The first level calculates the similarity of of all vowel letters and consonants letters. The second level calculates the similarity of long vowels and vowel letters at beginning and end position of the words and consonants letters. The third level calculates the similarity consonants letters only.

We applied our method on Arabic-English parallel and comparable corpora. We evaluated the extracted transliteration pairs using a statistical based transliteration system. This system is built using letters instead or words as tokens. The transliteration system achieves an accuracy of 0.50 and a mean F-score 0.8958 when trained on transliteration pairs extracted from a parallel corpus. The accuracy is 0.30 and the mean F-score 0.84 when we used instead a comparable corpus to automatically extract the transliteration pairs. This shows that the proposed semi-supervised transliteration mining algorithm is effective and can be applied to other language pairs. We also evaluated two segmentation techniques and reported the impact on the transliteration performance.

1. Introduction

Transliteration is the process of writing a word (mainly proper noun) from one language in the alphabet of another language. This process requires mapping the pronunciation of the word from the original language to the closest possible pronunciation in the target language. Both the word and its transliteration are called a Transliteration Pair (TP). The automatic extraction of TPs from parallel or comparable corpora is called Transliteration Mining (TM). The transliteration pairs are important for many applications like Machine Translations (MT), machine transliteration, cross language information retrieval (IR) and Name Entity Recognition (NER). For example, in MT, TM can be used to improve the word alignments, or to train a system to transliterate proper nouns in out-of-vocabulary (OOV) words. In machine transliteration, the obtained TPs are used to train statistical transliteration system, while in IR, it is used to enrich the search results with orthographical variations.

Recently, TM has gained considerable attention from the research community. There are several methods to perform TM: supervised, unsupervised and semi-supervised. Also, some TM researches focus on parallel corpora and others on comparable corpora. In this paper we will focus on semi-supervised method with both parallel corpora and comparable corpora.

We applied our method on an Arabic-English transliteration task using letter based SMT system trained on the extracted transliteration pairs. Then, we used this transliteration system in our semi-supervised method to extract transliteration pairs from comparable corpora. Although this work focuses on Arabic-English, it can be applied to any language pair. We are conducting this research in the context of MT, in order to decrease the OOV rate in the translation task.

There are several challenges related to Arabic transliteration. One of the challenges is that some Arabic letters have no phonically equivalent letters in English (e.g. ض and ط), and also some English letters do not have phonically equivalent letters in Arabic (e.g. v). Another challenge is the missing of short vowels (i.e. diacritics) in the Arabic text, while it should be mapped to existing letters in English text during the transliteration process. Additionally, some Arabic letters can be mapped to any letter from a group of phonically close English letters (e.g. ب to p or b), and some Arabic letters can be mapped to a sequence of English letters (e.g. خ to 'kh'). There is also a tokenization challenge, since unlike English, sometimes, the Arabic name is concatenated to one clitic (e.g. preposition ب or conjunction و) or both together (e.g. وب), which requires an advanced detection and seg-

mentation for these clitics before performing the transliteration.

There are two types of transliteration, forward and backward. In forward transliteration, the names are transliterated from its original language to another language, like the Arabic origin name "محمد" transliterated to "Mohamed" in English. In backward transliteration, the transliterated names are transliterated back to the origin names in its original language, like "بوش" will be transliterated back to "Bush". For simplicity, in this paper we will not differentiate between forward transliteration and backward transliteration. In future work, we will focus on addressing the specific problems related to each transliteration type.

The paper is organized as follows: the next section presents related work, followed by a description of the TM algorithm when using parallel corpora. This technique is extend to comparable corpora in section 4. The paper concludes with a discussion of the perspectives of this work.

2. Related work

The related work includes TM and transliteration research. For TM, there are several methods to perform it, supervised, unsupervised and semi-supervised. Also, some TM researches focus on parallel corpora and others on comparable corpora. [1] uses variant of the SOUNDEX methods and n-grams to improve precision and recall of name matching in the context of transliterated Arabic name search. Original, SOUNDEX was developed by [2] which is an algorithm used for indexing names by sound as pronounced in English. The SOUNDEX code for a name consists of a letter followed by three numerical digits: the letter is the first letter of the name, and the digits encode the remaining consonants. Similar sounding consonants share the same digit. For example, the labial consonants B, F, P, and V are each encoded as the number 1. The method proposed by [1] reduces the orthographical variations by 30% using SOUNDEX improved precision slightly but they observed a decrease in recall. [3] presents two methods for improving TM, phonetic conflation of letters and iterative training of a transliteration model. The first method is an improved SOUNDEX phonetic algorithm. They propose SOUNDEX like conflation scheme to improve the recall and F-measure. Also iterative training method was presented that improves the recall but decreases the precision.

[4] presents an adaptive learning framework for Phonetic Similarity Modeling (PSM) that supports the automatic construction of transliteration lexicons. PSM measures the phonetic similarity between source and target words pairs. In a bi-text snippet, when an source language word EW is spotted, the method searches for the word's possible target transliteration CW in its neighborhood. EW can be a single word or a phrase of multiple source language words. In this paper, they initialize the learning algorithm with minimum machine transliteration knowledge, then it starts acquiring more transliteration knowledge iteratively from the Web. They study the active learning and the unsupervised learning strategies that minimize human supervision in terms of data labeling. They report that the unsupervised learning is an effective way for rapid PSM adaptation while active learning is the most effective in achieving high performance. Another TM method relies on a Bayesian technique proposed by [5]. This method simultaneously co-segments and force-aligns the bilingual segments through rewards the re-use of features already in the model. The main assumption that transliteration pairs can be derived by using bilingual sequence pairs already learned by the model, or by introducing a very short unobserved pair into the derivation. They assume that incorrect pairs are likely to have large contiguous segments that are costly to force-align with the model. The transliteration classifier is trained on features derived from the alignment of the candidate pair as well as other heuristic features. They report a results indicate that transliteration mining of English-Japanese using this method should be possible at high levels of precision and recall. [6] adapts graph reinforcement to work with large training sets. They introducs parametrized exponential penalty to formulation of graph reinforcement which led to improvement in precision. They report that TM quality using comparable corpora is impacted by the presence of phonically similar words in comparable text, so they extracted the related segments that have high translation overlap and used them for TM, which leads to higher precision for the suggested TM methods. An automatic language pair independent method for transliteration mining using parallel corpora is proposed by [7]. They models transliteration mining as interpolation of transliteration and non-transliteration sub-models. Two methods, unsupervised and semi-supervised were presented with the results that show that semi-supervised method is out performing unsupervised method.

For transliteration research, [8] uses two algorithms based on sound and spelling mappings using finite state machines to perform the transliteration of Arabic names. They report that transliteration model can be trained on relatively small list of names which is easier to obtain than training data needed for training phonetic based models. [9] presents DirecTL, a language independent approach to transliteration. DirecTL is based on an online discriminative sequence prediction model that employes EM-based many-to-many unsupervised alignment between target and source. While, [10] uses a joint source channel models on the automatically aligned orthographic transliteration units of the auto-

matically extracted TPs. They compare the results with three online transliteration systems and reported better results.

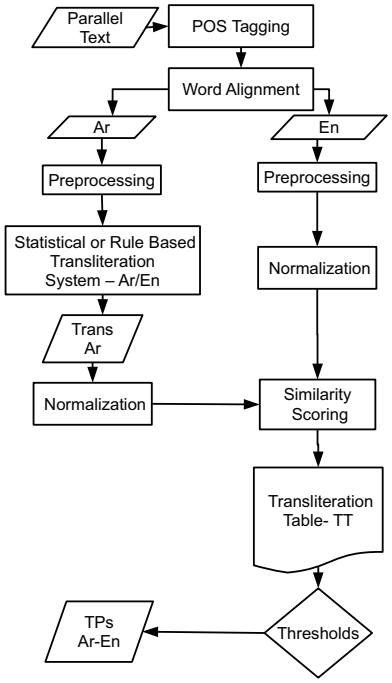


Figure 1: Extracting TPs from parallel corpora

3. Transliteration mining using parallel corpora - semi-supervised

In this section, we will introduce a corpus based computational method to extract TPs from parallel corpus. In order to evaluate the extracted pairs, we trained a letter based statistical transliteration system on TPs and evaluate the system performance which is correlated with the transliteration mining quality.

3.1. TM algorithm for parallel corpora

The algorithm as shown in Figure 1 is designed to compare two aligned words and detect the words which are transliteration of each other, with respect to the observations in section 3.3. We developed the following TM algorithm:

(1) First, the parallel corpus is tagged using a part-of-speech (POS) tagger. We used Stanford POS tagger [11] for English and Mada/Tokan [12] for Arabic POS tagging.

(2) Then, we align the tagged bitext using Giza++ [13], using the source/target alignment file, remove all aligned word pairs with POS tags other than noun (NN) or proper noun (PNN) tags and remove all English words starting with lower-case letters. Words which have most lowest alignment scores are removed (about 5% from the total number of aligned word pairs).

(3) After that removing the POS tags from Arabic and English words.

(4) Then, transliterate the Arabic word A into English using a rule based transliteration system (or a previously trained statistical based transliteration system).

(5) Normalize the transliteration of Arabic word A_t as well as the English word to $Norm_1$, $Norm_2$ and $Norm_3$ as explained in section 3.2. The objective of the normalization is folding English letters with similar phonetic to the same letter or symbol.

(6) For each aligned Arabic transliterated word A_t and English word E, use their normalized forms to calculate the three levels of similarity scores which we store in a transliteration table (TT).

(7) Extract TPs from the TT by applying a threshold on the three levels similarity scores. We selected the thresholds using empirical method shown in section 3.5.4.

3.2. English normalization and three levels similarity scores for TM

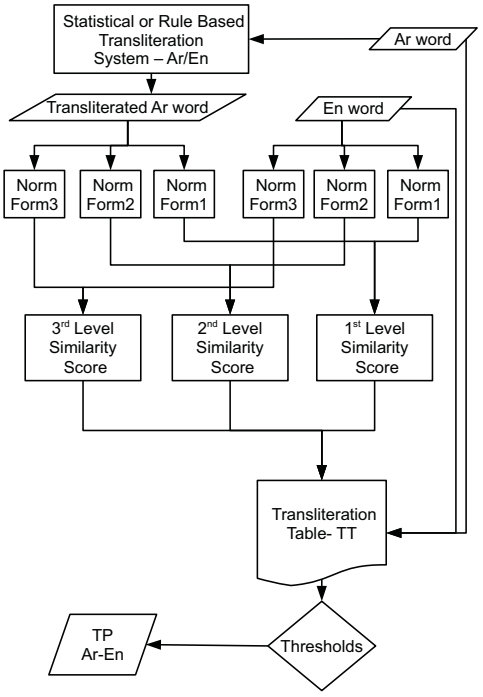


Figure 2: Calculating the three levels of similarity scores

As shown in Figure 2, we developed a three normalization functions which can be used to normalize the Arabic transliterated word and English word to be more comparable to each other phonically. These normalized forms are used to

calculate the similarity between the transliterated word and the English word based on three levels of similarity. The first level calculates the similarity of all vowel letters and consonants letters. The second level calculates the similarity of long vowels and vowel letters at beginning and end position of the words as well as consonants letters. The third level calculates the similarity of consonants letters only. The details of each normalization function as following:

(1) $Norm_1$ normalization function: Normalize the transliteration of Arabic word as well as the English word. The objective of the normalization is folding English letters with similar phonetic to one letter or symbol. In $Norm_1$, all letters are converted to lower case, phonically equivalent consonants and vowels are folded to one letter (e.g. p and b are normalized to b, v and f are normalized to f, i and e are normalized to e), double consonants are replaced by one letter, and finally a hyphen "-" is inserted after the initial two letters "al" -which is the transliteration of the usually concatenated Arabic article "ال"- if it is not already followed by it.

(2) $Norm_2$ normalization function: Using $Norm_1$ output, double vowels are replaced by one similar upper-case letter (i.e. ee is normalized to E), remove non-initial and non-final vowels only if not followed by vowel or not preceded by vowel.

(3) $Norm_3$ normalization function: Using $Norm_2$, hyphen - and vowels are removed.

Hence, for each Arabic word A and English word E. if A_t is the transliteration of A into English, we can calculate the following three levels similarity scores while i=1,2,3

$$TLS_i = \frac{Levenshtein(Norm_i(A_t), Norm_i(E))}{|Norm_i(E)|} \quad (1)$$

In this formula, Levenshtein function is the edit distance between the two words, which is the number of single-character edits required to change the first word into the second one.

3.3. Customized English pronunciation similarity comparison for Arabic-English transliteration

Our TM algorithm is based on the following pronunciation (and hence transliteration) observations in the English language considering the transliteration task from Arabic language characteristics:

1. In most cases, we can sort the letter's impact on transliteration from low to high as following:
   - Phonically similar vowels have low impact.
   - Phonically dissimilar vowels have medium impact.
   - Consonants letters have significant impact.
2. The double vowels produce long vowel sound have more impact on the pronunciation of the English word.
3. The sequence of two or more different vowel letters, has a special pronunciation which has more impact on the pronunciation of the English word.
4. The vowel at the initial position or at the final position in the word has significant impact on the pronunciation. The same applies for consonants (e.g. consider the following two names: Adham, Samy)

3.4. Transliteration system for TM evaluation

The transliteration system is built using the moses toolkit [14]. We train a letter-based SMT system on the list of TPs extracted using our TM algorithm explained in section 3.1. The distortion limit is set to 0 to disable any reordering. The transliteration system should be able to learn the proper letter mapping using the alignment of the letters, and hence be able to generate the possible transliterations of a name written in the source language script using the learned mapping rules into a name written in the target language script. This research focuses on the following points:

- Evaluate the performance of TM the algorithm by using the TPs to build a transliteration system. The transliteration system performance is correlated with the quality of the extracted TPs, and hence the TM performance.
- Acquiring a list of target language names for the letter based language model training.
- Study the impact of the segment length on the transliteration quality. In this context, two systems are trained to evaluate the segmentation for the word letters. We compared two segmentation scheme:
  - Simple segmentation of the word by separating individual letters.
  - Advanced segmentation of the word that segment the word to a group of 1-2 letters based on predefined phonetic units which combine two English letters -based on their position in the word- in one substring instead of separate letters (e.g. 'kh', 'kn', 'wh', 'sh' and 'ck').

- The impact of using different tuning metric, we compared the following metrics: TER, BLEU, (TER-BLEU)/2.

3.5. Experiments and evaluation

3.5.1. Purpose and data sets

The objectives of developing our transliteration system is to evaluate the quality of our TM algorithm and perform some research on improving the transliteration quality especially for unseen names in the training data. We evaluated the proposed TM algorithm using Arabic/English parallel corpus which contains about 3.8 million Arabic words and roughly 4.4 million English words. The evaluation of the TM algorithm is performed by training of a statistical system on the extracted TPs and evaluate the quality of transliteration output.

The extracted TPs are divided into three parts:

1. Training data set. The size of the training data is variable based on the selected three levels thresholds (9070 pairs to 10529 TPs).
2. Tuning data set (1k TPs).
3. Test data set. (1k TPs).

All occurrences of words in the TuningSet or TestSet were removed from the training data set.

3.5.2. Evaluation metrics

In order to evaluate the quality of our transliteration system, we used the de-facto standard metrics from ACL Name Entity Workshop (NEWS) [15]: ACC, mean F-Score, MRR, and MAP_{ref}. Here is a short description of each metric:

- ACC=Word Accuracy in Top-1, also known as Word Error Rate. It measures correctness of the first transliteration candidate in the candidate list produced by a transliteration system.
- F-Score= Fuzziness in Top-1. The mean F-score measures how different, on average, the top transliteration candidate is from its closest reference.
- MRR=Mean Reciprocal Rank measures traditional MRR for any right answer produced by the system, among the candidates.
- MAP_{ref} tightly measures the precision in the n-best candidates for the i-th source name, for which reference transliterations are available.

3.5.3. Acquiring a list of target language names for the language model training

We used two resources to get two lists of English names to train our letter based language model (LM). The first resource (LM1) is obtained from the English Gigaword corpus (using only XIN, AFP and NYT parts) by extracting a list of proper names using the Stanford name entity recognizer (NER) [16]. The second resource (LM2) is the English part of the extracted TPs. The Table 1 below compares the results of using LM1 vs. LM2. These results show that the target part (i.e. LM2) of the extracted TPs gives better ACC score while it has some impact on the mean F-score. We decided to use LM2 in all other experiments that measure other variables.

System	ACC	Mean F-Score	MRR	MAP_{ref}
LM1	0.43750	0.88160	0.54787	0.43750
LM2	0.44159	0.87860	0.54862	0.44160

Table 1: *LM1 vs. LM2*

3.5.4. Three levels similarity scores thresholds selections

Several systems were trained to evaluate the best thresholds to be used in our experiments. The experiments show that the best thresholds for 3-scores on tuning set are (TLS_3, TLS_2, TLS_1)=(0, 0.39, 0.49). The thresholds are highly dependent on the normalization functions $Norm_1$, $Norm_2$ and $Norm_3$, so changing the normalization functions will require a re-selection of the three thresholds. The scores of the TuningSet with different thresholds are mentioned in Table 2. Table 3 lists the systems with the TLS scores' thresholds used to select data to train each one.

System(*)	ACC	Mean F-Score	MRR	MAP_{ref}
SYS013 TPs=9167	0.43545	0.87940	0.54188	0.43545
SYS023 TPs=9070	0.44159	0.87860	0.54862	0.44160
SYS034 TPs=10529	0.44774	0.88226	0.55012	0.44774
SYS134 TPs=10529	0.43647	0.88042	0.54220	0.43647

Table 2: *Tuning set results with different thresholds*

System(*)	TLS_3	TLS_2	TLS_1
SYS013	0	0.19	0.39
SYS023	0	0.29	0.39
SYS034	0	0.39	0.49
SYS134	0.19	0.39	0.49

Table 3: *TLS scores' thresholds used for each system*

3.5.5. Segmentations techniques

We used two segmentation techniques, the first technique simply segments the NE into characters, the second one is an

System	ACC	Mean F-Score	MRR	MAP_{ref}
One letter	0.47951	0.89248	0.59226	0.47951
1-2 letters	0.50000	0.89589	0.61178	0.5000

Table 4: *One letter segmentation vs. Advanced segmentation*

advanced segmentation that group together letters that form one phonetic sound in one segment (e.g. ph, ch, sh, etc). Table 4 shows the results of both segmentation techniques. One can see that the second technique helps the letters alignment between source and target and hence improves the transliteration output.

3.5.6. Tuning metric selection

We used the mert tool for weight optimization [17]. We evaluated the impact of using mert tool with different metrics (BLEU, TER and (TER-BLEU)/2. Table 5 shows that (TER-BLEU)/2 gives better results than using BLEU alone or TER alone.

System	ACC	Mean F-Score	MRR	MAP_{ref}
BLEU	0.43648	0.87662	0.54322	0.43647
TER	0.43545	0.87638	0.54263	0.43545
$\frac{(TER-BLEU)}{2}$	0.44159	0.87860	0.54862	0.44159

Table 5: *Experiments with various tuning metrics*

3.5.7. Results

Using three levels similarity scores thresholds=(0, 0.29, 0.39) as explained in section 3.5.4, the total number of extracted TPs is 12988. Table 6 shows the percentage of extracted TPs as a function of the number of aligned words in the parallel text and the number of aligned words with an NNP/NN POS tag.

Data	Number of Words	Extracted TPs %
Bitext-Arabic	3.8M	0.24 %
Bitext-English	4.4M	0.21 %
List of aligned words	1249167	0.73 %
List of aligned NN*	161811	5.6 %

Table 6: *Extracted TPs rate*

In Table 7, we list the transliteration system results using the evaluation metrics mentioned in section 3.5.2. We report the scores for both TuningSet and TestSet. Both TuningSet and TestSet have not seen before in the training data.

System	ACC	Mean F-Score	MRR	MAP_{ref}
TuningSet	0.50000	0.89589	0.61178	0.5000
TestSet	0.46162	0.88412	0.58221	0.4616

Table 7: *TuningSet and TestSet scores*

4. Transliteration mining using comparable corpora - semi-supervised

In this section, we will introduce a corpus based computational method to extract transliteration pairs from comparable corpora. In order to evaluate the extracted pairs, we trained a letter based statistical transliteration system on them and evaluate the system performance which is correlated with the TM quality.

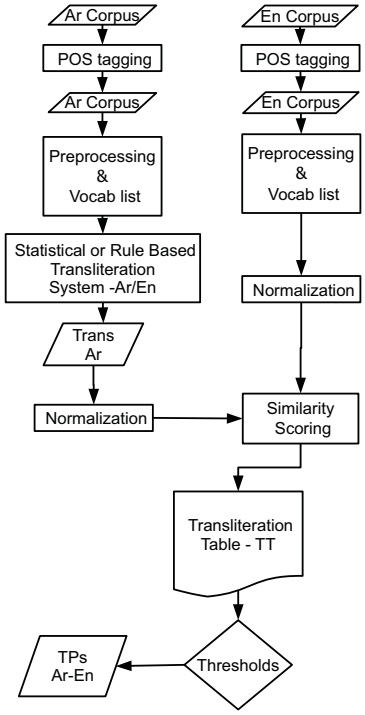


Figure 3: Extracting TPs from comparable corpora

4.1. TM algorithm for comparable corpora

Since it is easy to collect and find monolingual text than parallel text, it would be useful if we can perform TM using this large resources of monolingual text for any pair of languages. This method is inspired by the work of [18] on comparable corpora. We basically do the same at the letter level instead of the word level. Figure 3 shows an overview of the TM algorithm for comparable corpora. The algorithm is designed to remove the non-nouns words in order to minimize

the number of words in each monolingual text, then detects the words which are transliteration of each other, with respect to the observations listed in section 3.3, we score the similarity using three levels similarity scores to generated the transliteration table (TT), which is used later to extract the TPs using three thresholds on the three levels of similarity scores. The following steps explain the TM algorithm:

(1) First, each monolingual corpus is tagged using part-of-speech (POS) tagger. We used Stanford POS tagger [11] for English and Mada/Tokan [12] for Arabic POS tagging.

(2) Then, remove all words with POS tags other than noun (NN) or proper noun (PNN) tags and from the remaining words, remove all English words starts with lower-case letters.

(3) After that removing the POS tags from source text and target text.

(4) Derive two unique words lists (LIST_SRC and LIST_TRG) from both source and target texts.

(5) Then, transliterate source words list (LIST_SRC) into target language (LIST_SRC_TRANS) using rule based transliteration system (or previously created statistical based transliteration system).

(6) Normalize the transliteration of source words list as well as the English words list to the three normalized forms $Norm_1$, $Norm_2$ and $Norm_3$ as explained in section 3.2. The objective of the normalization is folding English letters with similar or close phonetic to same letter or symbol.

(7) Using the normalized values, for each transliterated word in the source language list WORD_AR_TRANS and target language word WORD_EN, calculate the 3-similarity scores between them which are stored in the transliteration table (TT).

(8) Extract TPs from the TT by applying a selected three thresholds on the three levels similarity scores.

4.2. Experiments and evaluation

4.2.1. Purpose and data sets

We evaluated the proposed TM algorithm by applying it on the Arabic Gigaword corpus (about 270.3 million Arabic words using only XIN, AFP and NYT parts) and the English Gigaword corpus (roughly 1470.3 million English words using only XIN, AFP and NYT parts).

We selected the thresholds using empirical method shown in section 4.2.2. The extracted TPs are used as training data. We used the same TuningSet and TestSet extracted from parallel corpus as mentioned in section 3.5.1.

As before, all occurrences of words in the TuningSet or TestSet were removed from the training data.

4.2.2. Three levels similarity scores thresholds selections

Several systems were trained to evaluate the best thresholds to be used in our experiments. Only two thresholds are compared, other thresholds are discarded because they almost give the same TPs. The experiments shows that the best thresholds for 3-scores on tuning set are (TLS_3, TLS_2, TLS_1)=(0, 0.29, 0.39) since they give slightly better mean F-Score and MRR. The scores of the TuningSet with different thresholds are mentioned in Table 8. Table 9 lists the systems with the TLS scores' thresholds used to select data to train each one.

System	ACC	Mean F-Score	MRR	MAP_{ref}
GSYS013 TPs=1.63M	0.30021	0.83973	0.40807	0.30021
GSYS023 TPs=1.96M	0.30021	0.84001	0.40817	0.30021

Table 8: *Tuning set results with different thresholds*

System(*)	TLS_3	TLS_2	TLS_1
GSYS013	0	0.19	0.39
GSYS023	0	0.29	0.39

Table 9: *TLS scores' thresholds used for each system*

4.2.3. Results

Using three levels similarity scores thresholds=(0, 0.29, 0.39) as explained in section 4.2.2, the total number of extracted TPs is 1.96 millions. Table 10 shows TPs rate with respect to the comparable corpora total number of words and the total number of words with NNP/NN POS tag. In Table 11, we list the transliteration system results using the evaluation metrics mentioned in section 3.5.2. We are reporting the scores for both TuningSet and TestSet. Both TuningSet and TestSet has not seen before in the training data.

Data	Number of Words	Extracted TPs %
Arabic Gigaword	270.3 M	0.73%
Arabic Gigaword NN*	18.7 M	10.48%
English Gigaword	1470.3 M	0.13%
English Gigaword NN*	8.1 M	24.20%

Table 10: *Extracted TPs rate*

5. Conclusions

In this paper we introduce a new semi-supervised transliteration mining method for parallel and comparable corpora. The method is mainly based on new suggested Three Levels of Similarity (TLS) scores to extract the transliteration pairs. The transliteration system trained on the transliteration pairs extracted from the parallel corpus achieves an accuracy of 0.50 and a mean F-score of 0.84 on the test set of unseen Arabic names. We also applied our translation mining approach on two Arabic and English monolingual corpora. The system trained on transliteration pairs extracted

System	ACC	Mean F-Score	MRR	MAP_{ref}
TuningSet	0.30021	0.84001	0.40817	0.30021
TestSet	0.27329	0.83345	0.39788	0.27329

Table 11: *TuningSet and TestSet scores*

from comparable corpora achieves an accuracy of 0.30 and a mean F-score of 0.84. This shows that the proposed semi-supervised transliteration mining algorithm is effective and can be applied to other language pairs.

6. Acknowledgment

This research was partially financed by DARPA under the BOLT contract.

7. References

[1] D. Holmes, S. Kashfi, and S. U. Aqeel, "Transliterated arabic name search." in *Communications, Internet, and Information Technology*, M. H. Hamza, Ed. IASTED/ACTA Press, 2004, pp. 267–273.

[2] R. Russell, "Specifications of letters," US patent number 1,261,167, 1918.

[3] K. Darwish, "Transliteration mining with phonetic conflation and iterative training," in *Proceedings of the 2010 Named Entities Workshop*, ser. NEWS '10. Association for Computational Linguistics, 2010, pp. 53–56.

[4] J.-S. Kuo, H. Li, and Y.-K. Yang, "Learning transliteration lexicons from the web," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ser. ACL-44. Association for Computational Linguistics, 2006, pp. 1129–1136.

[5] T. Fukunishi, A. Finch, S. Yamamoto, and E. Sumita, "Using features from a bilingual alignment model in transliteration mining," in *Proceedings of the 3rd Named Entities Workshop (NEWS 2011)*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, November 2011, pp. 49–57.

[6] A. El-Kahky, K. Darwish, A. S. Aldein, M. A. El-Wahab, A. Hefny, and W. Ammar, "Improved transliteration mining using graph reinforcement," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Association for Computational Linguistics, 2011, pp. 1384–1393.

[7] H. Sajjad, A. Fraser, and H. Schmid, "A statistical model for unsupervised and semi-supervised transliteration mining," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, July 2012, pp. 469–477.

[8] Y. Al-Onaizan and K. Knight, "Machine transliteration of names in arabic text," in *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*, ser. SEMITIC '02. Association for Computational Linguistics, 2002, pp. 1–13.

[9] S. Jiampojamarn, A. Bhargava, Q. Dou, K. Dwyer, and G. Kondrak, "Directl: a language-independent approach to transliteration," in *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, ser. NEWS '09. Association for Computational Linguistics, 2009, pp. 28–31.

[10] H. Sajjad, A. Fraser, and H. Schmid, "An algorithm for unsupervised transliteration mining with an application to word alignment," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 430–439.

[11] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. Association for Computational Linguistics, 2003, pp. 173–180.

[12] O. R. Nizar Habash and R. Roth, "Mada+tokan: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization," in *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, K. Choukri and B. Maegaard, Eds. Cairo, Egypt: The MEDAR Consortium, April 2009.

[13] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Comput. Linguist.*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[14] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07. Association for Computational Linguistics, 2007, pp. 177–180.

[15] A. K. M. L. Min Zhang, Haizhou Li, Ed., *Report of NEWS 2012 Machine Transliteration Shared Task*, vol. pages 10–20. Jeju, Republic of Korea: Association for Computational Linguistics, July 2012.

[16] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ser. ACL '05. Association for Computational Linguistics, 2005, pp. 363–370.

[17] N. Bertoldi, B. Haddow, and J.-B. Fouet, "Improved minimum error rate training in moses," *Prague Bull. Math. Linguistics*, pp. 7–16, 2009.

[18] S. Abdul Rauf and H. Schwenk, "Parallel sentence generation from comparable corpora for improved smt," *Machine Translation*, vol. 25, no. 4, pp. 341–375, Dec. 2011.

A Simple and Effective Weighted Phrase Extraction for Machine Translation Adaptation

Saab Mansour and Hermann Ney

Human Language Technology and Pattern Recognition
Computer Science Department
RWTH Aachen University, Aachen, Germany
{mansour,ney}@cs.rwth-aachen.de

Abstract

The task of domain-adaptation attempts to exploit data mainly drawn from one domain (e.g. news) to maximize the performance on the test domain (e.g. weblogs). In previous work, weighting the training instances was used for filtering dissimilar data. We extend this by incorporating the weights directly into the standard phrase training procedure of statistical machine translation (SMT). This allows the SMT system to make the decision whether to use a phrase translation pair or not, a more methodological way than discarding phrase pairs completely when using filtering. Furthermore, we suggest a combined filtering and weighting procedure to achieve better results while reducing the phrase table size. The proposed methods are evaluated in the context of Arabic-to-English translation on various conditions, where significant improvements are reported when using the suggested weighted phrase training. The weighting method also improves over filtering, and the combined filtering and weighting is better than a standalone filtering method. Finally, we experiment with mixture modeling, where additional improvements are reported when using weighted phrase extraction over a variety of baselines.

1. Introduction

Over the last years, large amounts of monolingual and bilingual training corpora were collected for statistical machine translation (SMT). Early years focused on structured data translation such as newswire and parliamentary discussions. Nowadays, due to the success of SMT, new domains of translation are being explored, such as talk translation in the IWSLT TED evaluation [1] and dialects translation within the DARPA BOLT project [2]. The introduction of the BOLT project marks a shift in the Arabic NLP community, changing the focus from handling Modern Standard Arabic (MSA) structured data (e.g. news) to dialectal Arabic user generated noisy data (e.g. emails, weblogs). Dialectal Arabic is mainly spoken and scarcely written, even when it is written, the lack of common orthography causes significant variety and ambiguity in lexicon and morphology. The challenge is even greater due to the domain of informal communication, which is noisy by its nature. In this work, we perform experiments on both the BOLT and the IWSLT TED setups, allowing us to explore both lectures and weblogs domains, drawing more robust conclusions and enabling a larger group of researchers to reproduce our experiments and results.

The task of domain adaptation tackles the problem of utilizing existing resources in the most beneficial way for the new domain at hand. Given some general domain data and a new domain to tackle, adaptation is the task of modifying the SMT components in such a way that the new system will perform better on the new domain than the general domain system.

In this work, we focus on translation model (TM) adaptation. The TM (e.g. phrase model) is the core component of state-of-the-art SMT systems, providing the building blocks (e.g. phrase translation pairs) to perform the search for the best translation. Several methods were suggested already for TM adaptation. We experiment with training data weighting, where one assigns higher weights to relevant domain training instances, thus causing an increase of the corresponding probabilities. Therefore, translation pairs which can be obtained from relevant training instances will have a higher chance of being utilized during search.

Weighted phrase extraction can be done at several levels of granularity, including sub-corpus level, sentence level and phrase level. In this work, we focus on sentence level weighting for phrase extraction. Previous work also suggested filtering, which can be seen as a crude weighting were sentences are assigned $\{0, 1\}$ weights. We compare weighting to filtering and show superior results for weighting. In a scenario where efficiency constraints are imposed on the SMT system, reducing the TM size can serve as a solution. For such a scenario, we suggest filtering combined with weighting, and show that this method achieves better results than filtering alone.

Finally, we explore mixture modeling, where a purely in-domain TM is interpolated with various adapted TMs, and show further improvements. The resulting method described in this paper is simple and easy to reimplement, yet effective.

The rest of the paper is organized as follows. Related work on data filtering, weighting and mixture modeling is de-

tailed in Section 2. The weighted phrase extraction training and the method for assigning weights are described in Section 3. Section 4 recaps briefly mixture modeling methods that will be used in the paper. Experimental setup including corpora statistics and the SMT system are described in Section 5. The results of the described methods are summarized in Section 6. Last, we conclude with few suggestions for future work.

2. Related work

A broad range of methods and techniques have been suggested in the past for domain adaptation for SMT. The techniques include, among others: *(i)* semi-supervised training where one translates in-domain monolingual data and utilizes the automatic translations for retraining the LM and/or the TM ([3],[4]), *(ii)* different methods of interpolating in-domain and out-of-domain models ([5], [6], [7]) *(iii)* and sample weighting on the sentence or even the phrase level for LM training ([8],[9]) and TM training ([10],[11],[12]). Note that filtering is a special case of the sample weighting method where a threshold is assigned to discard unwanted samples.

Weighted phrase extraction can be done at several levels of granularity. [6] perform TM adaptation using mixture modeling at the corpus level. Each corpus in their setting gets a weight using various methods including language model (LM) perplexity and information retrieval methods. Interpolation is then done linearly or log-linearly. The weights are calculated using the development set therefore expressing adaptation to the domain being translated. [13] also performs weighting at the corpus level, but the weights are integrated into the phrase model estimation procedure. His method does not show an advantage over linear interpolation. A finer grained weighting is that of [10], who assign each sentence in the bitexts a weight using features of meta-information and optimizing a mapping from feature vectors to weights using a translation quality measure over the development set. [11] perform weighting at the phrase level, using a maximum likelihood term limited to the development set as an objective function to optimize. They compare the phrase level weighting to a "flat" model, where the weight directly models the phrase probability. In their experiments, the weighting method performs better than the flat model, therefore, they conclude that retaining the original relative frequency probabilities of the TM is important for good performance.

In this work, we propose a simple yet effective method for weighted phrase extraction expressing adaptation. Our method is comparable to [10] assigning each sentence pair in the training data a weight. We differ from them by using a weight based on the cross-entropy difference method proposed in [9] for LM filtering and later adapted in [12] for TM filtering. In weighting, all the phrase pairs are retained, and only their probability is altered. This allows the decoder to make the decision whether to use a phrase pair or not, a more methodological way than removing phrase pairs completely when filtering. We compare our weighting method to filtering and show superior results. In some cases, one might be interested in reducing the size of the TM for efficiency reasons. We combine filtering with weighting, and show that this leads to better performance than filtering alone.

Last, as done in some of the previous work mentioned above, we experiment with mixture modeling over the weighted phrase models. We use linear and log-linear interpolation similar to [6]. We differ from [13] by showing improved results over linear interpolation of baseline models. [14] analyze the effect of adding a general-domain corpus at different parts of the SMT training pipeline. A method denoted as "x+yE" performed best in their experiments. This method extracts all phrases from a concatenation of in-domain and general corpora, then, if a phrase pair exists in the in-domain phrase table it is assigned the in-domain probability, otherwise it is assigned the probability from the concatenation phrase table. We call this method an *ifelse* combination and test it in our experiments.

3. Weighted phrase extraction

The classical phrase model is trained using a "simple" maximum likelihood estimation, resulting in a phrase translation probability being defined by relative frequency:

$$p(\tilde{f}|\tilde{e}) = \frac{\sum_r c_r(\tilde{f},\tilde{e})}{\sum_{\tilde{f}'}\sum_r c_r(\tilde{f}',\tilde{e})} \quad (1)$$

Here, $\tilde{f},\tilde{e}$ are contiguous phrases, $c_r(\tilde{f},\tilde{e})$ denotes the count of $(\tilde{f},\tilde{e})$ being a translation of each other (usually according to word alignment and heuristics) in sentence pair (s_r,t_r). One method to introduce weights to equation (1) is by weighting each sentence pair by a weight w_r. Equation (1) will now have the extended form:

$$p(\tilde{f}|\tilde{e}) = \frac{\sum_r w_r \cdot c_r(\tilde{f},\tilde{e})}{\sum_{\tilde{f}'}\sum_r w_r \cdot c_r(\tilde{f}',\tilde{e})} \quad (2)$$

It is easy to see that setting $\{w_r = 1\}$ will result in equation (1) (or any non-zero equal weights). Increasing the weight w_r of the corresponding sentence pair will result in an increase of the probabilities of the phrase pairs extracted. Thus, by increasing the weight of in-domain sentence pairs, the probability of in-domain phrase translations could also increase. Next, we discuss several methods for setting the weights in a fashion which serves adaptation.

3.1. Weight estimation

Several weighting schemes can be devised to manifest adaptation. One way is to manually assign suitable weights to corpora using information about genre, corpus provider, compilation method and other attributes of the corpora. For example, a higher weight (e.g. 10) can be assigned to in-domain

corpora sentences, while a lower weight (e.g. 1) is assigned to other corpora sentences.

LM cross-entropy scoring can be used for both monolingual data filtering for LM training as done in [9], or bilingual data filtering for TM training as done in [12]. Next, we recall the scoring methods introduced in the above previous work and utilize it for our proposed weighted phrase extraction method.

Given some corpus I which represents the domain we want to adapt to, and a general corpus O, [9] first generate a random subset $\hat{O} \subseteq O$ of approximately the same size as I (this is not required for the method to work, and is used to make the models generated by the corpora more comparable), and train the LMs LM_I and $LM_{\hat{O}}$ using the corresponding training data. Then, each sentence $o \in O$ is scored according to:

$$H_I(o) - H_{\hat{O}}(o) \tag{3}$$

where $H_M(o)$ ($M \in \{I, \hat{O}\}$) is the per-word cross-entropy according to a language model trained on M. Let $o = w_1 \dots w_n$, then we have

$$H_M(o) = -\frac{1}{n}\sum_{i=1}^{n} \log p_M(w_i|w_{i-1}) \tag{4}$$

for a 2-gram LM case.

The intuition behind equation (3) is that we are interested in sentences as close as possible to the in-domain, but also as far as possible from the general corpus. [9] show that using equation (3) performs better in terms of perplexity than using in-domain cross-entropy only ($H_I(o)$). For more details about the reasoning behind equation (3) we refer the reader to [9].

[12] adapted the LM scores for bilingual data filtering for the purpose of TM training. In this case, we have source and target in-domain corpora I_{src} and I_{trg}, and correspondingly, general corpora O_{src} and O_{trg}, with random subsets $\hat{O}_{src} \subseteq O_{src}$ and $\hat{O}_{trg} \subseteq O_{trg}$. Then, we score each sentence pair (s_r, t_r) by:

$$d_r = [H_{I_{src}}(s_r) - H_{\hat{O}_{src}}(s_r)] + [H_{I_{trg}}(t_r) - H_{\hat{O}_{trg}}(t_r)] \tag{5}$$

We utilize d_r for our suggested weighted phrase extraction. d_r can be assigned negative values, and lower d_r indicates sentence pairs which are more relevant to the in-domain. Therefore, we negate the term d_r to get the notion of higher weights indicating sentences being closer to the in-domain, and use an exponent to ensure positive values. The final weight is of the form:

$$w_r = e^{-d_r} \tag{6}$$

This term is proportional to perplexities and inverse perplexities, as the exponent of entropy is perplexity by definition.

As done in [12], we compare using (5) to source only cross-entropy difference $[H_{I_{src}}(s) - H_{\hat{O}_{src}}(s)]$ and target only cross-entropy difference $[H_{I_{trg}}(t) - H_{\hat{O}_{trg}}(t)]$, in addition to source only in-domain cross-entropy $H_{I_{src}}(s)$.

4. Mixture modeling

Mixture modeling is a technique for combining several models using weights assigned to the different components. Domain adaptation could be achieved using mixture modeling when the weights are related to the proximity of the components to the domain being translated. As we generate several translation models differing by the training corpora domain and extraction method, interpolating these models could yield further improvements. In this work, we focus on two variants of mixture modeling, namely linear and loglinear interpolation.

4.1. Linear interpolation

Linear interpolation is a commonly used framework for combining different SMT models into one ([6]). As we experiment with interpolating two phrase models in this work (in-domain and other-domain), we obtain the following simplified interpolation formula:

$$p(\tilde{f}|\tilde{e}) = \lambda p_I(\tilde{f}|\tilde{e}) + (1-\lambda)p_O(\tilde{f}|\tilde{e}) = \tag{7}$$

λ is assigned a value in the range $[0, 1]$ to keep the resulting phrase model normalized. We set the value empirically on the development set testing different λ with steps of 0.1. Phrase pairs which appear in one model but not in the second are assigned small probabilities by the second model. The probabilities of the final mixture model are renormalized.

4.2. Loginear interpolation

Loglinear interpolation of phrase models fits directly into the loglinear framework of SMT ([7]). The weights of the different phrase models could be then tuned directly within the tuning procedure of the SMT system. This results in doubling the number of phrase model features, which could cause additional search errors, overfitting and finding an inferior local optima. Again, we assign a small probability to unknown phrase pairs. In this case, we do not perform renormalization to avoid overweighting of unknown phrase pairs.

5. Experimental setup

5.1. Training corpora

To evaluate the introduced methods experimentally, we use the BOLT Phase 1 Dialectal-Arabic-to-English task. The dialect chosen for Phase 1 is Egyptian Arabic (henceforth *Egyptian*). We confirm our findings by some final experiments on the IWSLT 2011 TED Arabic-to-English task.

The BOLT program goes beyond previous projects, shifting the focus from translating structured standardized text, such as Modern Standard Arabic (MSA) newswire, to a user generated noisy text such as Arabic dialect emails or weblogs. Translating Arabic dialects is a challenging task due to the scarcity of training data and the lack of common orthography causing a larger vocabulary size and higher ambigu-

Data style	Sentences	Tokens
United Nations	3557K	122M
Newswire	1918K	57M
Web	13K	280K
Newsgroup	25K	720K
Broadcast	91K	2M
Lexicons	213K	530K
Iraqi, Levantine	617K	4M
General (sum of above)	6434K	187M
Egyptian	240K	3M

Table 1: BOLT bilingual training corpora style and statistics. The number of tokens is given for the source side.

Set	Sen	Tok	OOV/IN	OOV/ALL
BOLT P1 Egyptian-to-English				
Egy (IN)	240K	3M		
General	6.4M	187M		
dev	1219	18K	387 (2.2%)	160 (0.9%)
test	1510	27K	559 (2.1%)	201 (0.7%)
IWSLT 2011 TED Arabic-to-English				
TED (IN)	90K	1.6M		
UN	7.9M	228M		
dev	934	19K	408 (2.2%)	184 (1.0%)
test	1664	31K	495 (1.6%)	228 (0.8%)

Table 2: Bilingual corpora statistics: the number of tokens is given for the source side. OOV/X denotes the number of OOV words in relation to corpus X (the percentage is given in parentheses). ALL denotes the concatenation of all training data for the specific task.

ity. Due to the scarcity of in-domain training data, MSA resources are being utilized for the project. In such a scenario, an important research question arises on how to use the MSA data in the most beneficial way to translate the given dialect. The training data for the BOLT Phase 1 program is summarized in Table 1. The table includes data style and size information. Most of the BOLT training data is available through the linguistic data consortium (LDC) and is regularly part of the NIST open MT evaluation [1].

The IWSLT 2011 evaluation campaign focuses on the translation of TED talks, a collection of lectures on a variety of topics ranging from science to culture. It is important to stress that IWSLT 2011 is different from previous years' campaigns by the genre shifting from the traveling domain (BTEC task) to lectures (TED task). Further, the amount of training data provided for the TALK task is considerably larger than for the BTEC task. For Arabic-to-English, the bilingual data consists of roughly 100K sentences of in-domain TED talks data and 8M sentences of out-of-domain United Nations (UN) data. This makes the task more similar to real-life MT system conditions, and the discrepancy between the training and the test domain opens a window for a variety of adaptation methods.

The bilingual training and test data for the Egyptian-to-English and Arabic-to-English tasks are summarized in Table 2[2]. The English data was tokenized and lowercased while the Arabic data was tokenized and segmented with the ATB scheme (this scheme splits all clitics except the definite article and normalizes the Arabic characters alef and yaa).

From Table 2, we note that the general data considerably reduces the number of out-of-vocabulary (OOV) words. This comes with the price of increasing the size of the training data by a factor of more than 50. A simple concatenation of the corpora might mask the phrase probabilities obtained from the in-domain corpus, causing a deterioration in performance. One way to avoid this contamination is by filtering the general corpus, but this discards phrase translations completely from the phrase model. A more principled way is by weighting the sentences of the corpora differently, such that sentences which are more related to the domain will have higher weights and therefore have a stronger impact on the phrase probabilities.

For language model training purposes, we use an additional 8 billion words for BOLT (4B words from the LDC gigaword corpus and 4B words collected from web resources) and 1.4 billion words for IWSLT (supplied as part of the campaign monolingual training data [3]).

5.2. Translation system

The baseline system is built using a state-of-the art phrase-based SMT system similar to Moses [15]. We use the standard set of models with phrase translation probabilities for source-to-target and target-to-source directions, smoothing with lexical weights, a word and phrase penalty, distance-based reordering and an n-gram target language model. The lexical models are trained on the in-domain portion of the data and kept constant throughout the experiments. This way we achieve more control on the variability of the experiments. In the experiments, we update the phrase probability features in both directions of translation. The SMT systems are tuned on the *dev* development set with minimum error rate training [16] using BLEU [17] accuracy measure as the optimization criterion. We test the performance of our system on the *test* set using the BLEU and translation edit rate (TER) [18] measures. We use TER as an additional measure to verify the consistency of our improvements and avoid over-tuning. The BOLT results are case insensitive while the IWSLT results are case sensitive. In addition to the raw automatic results, we perform significance testing over the *test*

[1] For a list of the NIST MT12 corpora, see `http://www.nist.gov/itl/iad/mig/upload/OpenMT12_LDCAgreement.pdf`

[2] The test sets for BOLT are extracted from the LDC2012E30 corpus - BOLT Phase 1 DevTest Source and Translation V4.

[3] For a list of the IWSLT TED 2011 training corpora, see `http://www.iwslt2011.org/doku.php?id=06_evaluation`

Translation model	dev		test	
	BLEU	TER	BLEU	TER
Unfiltered				
EGY	24.6	61.2	22.2	62.6
EGY+GEN	25.3	60.6	22.5	61.9
Filtered				
EGY+GEN-1Mbest	25.4	60.5	22.9	61.6
EGY+GEN-1Mrand	25.3	60.6	22.6	61.7
Weighted phrase extr.				
10EGY+1GEN	25.6	60.2	22.8	61.5
ppl_1-src(EGY+GEN)	25.6	60.7	22.9	61.5
ppl-src(EGY+GEN)	25.6	60.6	23.3‡	61.0
ppl-trg(EGY+GEN)	25.6	60.6	22.8	61.8
ppl(EGY+GEN)	25.6	60.1	23.3‡	60.9‡
ppl(EGY+GEN1Mbest)	25.6	60.0	23.0	61.4
Mixture modeling				
-loglin-EGY+GEN	24.7	61.3	22.0	62.8
-loglin-ppl(EGY+GEN)	24.9	61.1	22.1	62.3
-linear-EGY+GEN	25.7	60.4	22.9	61.4
-linear-ppl(EGY+GEN)	26.0	59.9	23.3‡	60.6‡
-ifelse-EGY+GEN	25.6	60.2	23.0	61.1
-ifelse-ppl(EGY+GEN)	25.7	60.2	23.1	61.0

Table 3: BOLT 2012 Egyptian-English translation results. BLEU and TER results are in percentages. *EGY* denotes the Egyptian in-domain corpus, *GEN* denotes the general other corpora. Significance is marked with ‡ and measured over the *EGY+GEN* baseline.

set. For both BLEU and TER, we perform bootstrap resampling with bounds estimation as described in [19]. We use the 90% and 95% (denoted by † and ‡ correspondingly in the tables) confidence thresholds to draw significance conclusions.

6. Results

In this section, we compare the proposed methods of weighted phrase extraction against unfiltered (in-domain and full) and filtered translation model systems. We start by testing our methods on the BOLT task, and finally verify the results on the IWSLT task.

6.1. BOLT results

The results of the BOLT Phase 1 Egyptian-English task are summarized in Table 3. Adding the general-domain (*GEN*) corpora to the in-domain (*EGY*) corpora system (unfiltered) increases the translation quality slightly by +0.3% BLEU on the *test* set. This increase might be attributed to the fact that the number of OOVs is decreased by adding the GEN corpora three folds. But, in addition, the various corpora that assemble the general-domain corpus are collected from various resources, increasing the possibility that there exists relevant training data to the domain being tackled.

When adding to *EGY* a filtered *GEN* corpus, where the 1000K best sentences according to the bilingual cross-entropy difference (equation (5)) are kept (*EGY+GEN-1000K-best*), the results improve by another +0.4% BLEU on *test* in comparison to the full *EGY+GEN* system. Thus, the filtering is able to retain sentences which are more relevant to the domain being translated. As a control experiment, we selected 1000K sentences from the *GEN* corpus randomly and added them to the *EGY* corpus (*EGY+GEN-1000K-rand*). In the BOLT setup, the cross-entropy based filtering seems to have only slight edge over random selection, perhaps due to the generality and usefulness of *GEN*.

In the third block of experiments, we compare the suggested methods for weighted phrase extraction. In the first experiment, we give higher weights to bilingual sentences from in-domain (10) as opposed to smaller weights to the general corpus (1). The resulting system (*10EGY+1GEN*) is comparable to the filtered *EGY+GEN-1000K-best*. In comparison to the *EGY+GEN* baseline, small improvements are observed on *dev* (+0.3% BLEU) and on *test* (+0.3% BLEU). Next, we compare the suggested weighting schemes, including source only in-domain cross-entropy based (denoted by *ppl_1-src* in the table), source only cross-entropy difference (*ppl-src*), target only cross-entropy difference (*ppl-trg*) and bilingual cross-entropy difference (*ppl*). We weight the bilingual training sentences (both in-domain and general-domain *EGY+GEN*) by the corresponding perplexity weight. All the weighting schemes improve over the baseline, where *ppl_1-src* and *ppl-trg* perform worst among the methods, and bilingual cross-entropy difference *ppl* has a slight edge on TER over source side only *ppl-src*. The *ppl(EGY+GEN)* system achieves the best results where +0.8% BLEU and -1.0% TER are observed on *test* in comparison to the *EGY+GEN* baseline. The improvements on both BLEU and TER are statistically significant at the 95% level, the only system being able to achieve that among weighted and filtered systems. In the final experiment, we combine filtering with weighting, where the best 1000K sentences of *GEN* are concatenated to *EGY* and a weighted phrase extraction using perplexity is done over this concatenation (*ppl(EGY+GEN-1000K-best)*). This system improves slightly over the unweighted *EGY+GEN-1000K-best* system, with +0.2% BLEU and -0.5% TER on *dev*, and +0.1% BLEU and -0.2% TER on *test*. Thus, if one is interested in a smaller TM, filtering combined with weighting is the best method to use according to our experiments.

In the last block of experiments, model combination is tested. We compare mixing the in-domain TM *EGY* with standard *EGY+GEN* TM and weighted *ppl(EGY+GEN)* one, using log-linear and linear interpolation as done in [6], and ifelse combination as done in [14]. The first observation is that log-linear interpolation performs poorly and worse than linear interpolation, supporting the results of [6] and [13] and contradicting [12]. [12] describe a special case where the overlap between the combined phrase tables in their experiments is small, which could explain the difference. Linear

Translation model	dev		test	
	BLEU	TER	BLEU	TER
Unfiltered				
TED	27.2	54.1	25.3	57.1
TED+UN	27.1	54.8	24.4	58.6
Filtered				
TED+UN-1Mbest	27.7	53.7	25.5	56.9
TED+UN-1Mrand	27.4	54.0	25.1	57.1
Weighted phrase extr.				
10TED+1UN	28.2	53.4	25.4	56.8
ppl_I-src(TED+UN)	27.9	53.3	25.5	55.8
ppl-src(TED+UN)	28.1	53.2	26.0	56.5
ppl-trg(TED+UN)	28.0	53.0	25.8	56.2
ppl(TED+UN)	28.1	52.9	26.0	56.2†
ppl(TED+UN-1Mbest)	28.1	53.1	25.8	56.3
Mixture modeling				
-loglin-TED+UN	26.8	53.9	24.0	58.3
-loglin-ppl(TED+UN)	27.2	53.9	24.7	57.6
-linear-TED+UN	28.0	53.1	25.9	56.2†
-linear-ppl(TED+UN)	28.1	53.3	25.9	56.1‡
-ifelse-TED+UN	28.4	52.6	25.9	56.0
-ifelse-ppl(TED+UN)	28.2	52.8	25.7	56.4

Table 4: IWSLT TED 2011 Arabic-English translation results. BLEU and TER results are in percentages. *TED* denotes the TED lectures in-domain corpus, *UN* denotes the united nations corpus. Significance is marked with ‡ and measured over the *TED* baseline.

combination on the other hand performs well, always improving over the respective combined standalone TMs. The mixture weight value for linear interpolation is set empirically by ranging the weight of the in-domain corpus *EGY* from $[0, 1]$ with steps of 0.1. The best result on the development set was achieved for a weight of 0.9. The linear mixture of *EGY* and *EGY+GEN* already achieves large improvements over the baseline. Still, interpolation with the weighted phrase table system (*EGY-linear-ppl(EGY+GEN)*) achieves the best results, improving over the mixture counterpart *EGY-linear-EGY+GEN* by +0.4% BLEU and up-to -0.8% TER on *test*. For both linear interpolation settings, $\lambda = 0.9$ for equation (7) performed best on the development set. Even though the ifelse combination is rather simple, the results are surprisingly good, still, the best linear combination performs better than the ifelse method. Similar to the other combination methods, using the weighted phrase table has a slight edge over the unweighted counterpart.

6.2. IWSLT TED results

The results of the IWSLT TED 2011 Arabic-English task are summarized in Table 4. Unlike the BOLT task, adding the out-of-domain *UN* corpus to the in-domain *TED* corpus system decreases the translation quality by -0.9% BLEU on the *test* set. This suggests a big discrepancy between the in-domain and the out-of-domain bilingual training corpora. Even though the *UN* corpus decreases the OOV ratio by a factor of 2 according to Table 2, the 100 times larger *UN* corpus masking the in-domain phrase probabilities seems to be more important and decisive for the degradation in performance. This claim is supported by the result of the *TED+UN-1000K-rand* system, which improves over *TED+UN*, due to the smaller *UN* selection that is being used and reducing the contamination of the in-domain phrase probabilities. When adding to *TED* a filtered *UN* corpus, where the 1000K best sentences according to the bilingual cross-entropy difference are kept (*TED+UN-1000K-best*), the results improve by 0.8% BLEU on *dev*, but smaller improvement of 0.2% BLEU is observed on *test*. In the context of filtering, cross-entropy based filtering is again performing better than random selection.

In the third block of experiments, we compare the suggested methods for weighted phrase extraction. The trends are similar to the BOLT results, where the perplexity based weighting achieves the best results and big improvements over the in-domain baseline, where the improvements on TER are statistically significant at the 90% level. A combined filtering and weighting (*ppl(TED+UN-1000K-best)*) performs better than unweighted filtering (*TED+UN-1000K-best*) by +0.3% BLEU and bigger -0.6% TER improvements on *test*.

For the mixture modeling results, loglinear interpolation decreases the performance dramatically, while linear interpolation achieves comparable results to the best weighted extraction, and no further improvements were observed. We hypothesize that mixture modeling did not yield improvements for IWSLT due to the big discrepancy between *TED* and *UN*, limiting the margin of improvements that is possible to achieve.

7. Conclusions

In this work, we utilize cross-entropy based weights for domain adaptation. We extend on previous work, where the weights are used for filtering purposes, by incorporating the weights directly into the standard maximum likelihood estimation of the phrase model. The weighted phrase extraction influences the phrase translation probabilities, while keeping the set of phrase pairs intact. We find this a more methodological way for adaptation than a hard decision where filtering is done. In some scenarios where efficiency constraints are imposed on the SMT system, filtering might be necessary. We propose a combined filtering and weighting method.

The proposed methods are evaluated in the context of Arabic-to-English translation on two conditions, IWSLT TED MSA lectures and BOLT Egyptian weblogs. The weighted phrase extraction method shows consistent improvements on both tasks, with up-to +1.1% BLEU and -1.7% TER improvements over the purely in-domain BOLT baseline, and +0.7% BLEU and -0.9% TER over the TED

baseline. The new method is also improving over filtering, and the combined filtering and weighting is better than a standalone filtering method. Thus, if one is interested in a smaller TM, filtering combined with weighting is the best method to use according to our experiments.

Finally, we tried mixture modeling of the in-domain and the various adapted TMs. Log-linear interpolation performed poorly in our experiments, which is consistent with previous work. On the other hand, linear interpolation performed well, achieving comparable results to the best system on the TED task, and further improvements on the BOLT task. We hypothesize that interpolation could not help for the TED task due to the big distance between the (scientific, cultural) lectures and the parliamentary discussions domains, limiting the improvement range of adaptation at the sentence level. On the BOLT task, interpolation with weighted phrase extraction performed better than interpolation with a standard phrase model, supporting the good performance of our suggested new method.

In future work, it will be interesting to compare different weighting methods in the weighted maximum likelihood estimation framework. Additionally, the effect of the granularity of weighting could be evaluated, comparing sentence versus corpus versus documents (any set of sentences) weighting.

8. Acknowledgements

This work was partially realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation, and also partially funded by the Defense Advanced Research Projects Agency (DARPA) under Contract No. 4911028154.0. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

9. References

[1] M. Federico, L. Bentivogli, M. Paul, and S. Stker, "Overview of the IWSLT 2011 evaluation campaign," in *International Workshop on Spoken Language Translation*, 2011, pp. 11–27.

[2] R. Zbib, E. Malchiodi, J. Devlin, D. Stallard, S. Matsoukas, R. M. Schwartz, J. Makhoul, O. Zaidan, and C. Callison-Burch, "Machine translation of arabic dialects," in *HLT-NAACL*, 2012, pp. 49–59.

[3] N. Ueffing, G. Haffari, and A. Sarkar, "Transductive learning for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 25–32. [Online]. Available: http://www.aclweb.org/anthology/P07-1004

[4] H. Schwenk, "Investigations on large-scale lightly-supervised training for statistical machine translation," in *International Workshop on Spoken Language Translation*, 2008, pp. 182–189.

[5] Y. Lu, J. Huang, and Q. Liu, "Improving statistical machine translation performance by training data selection and optimization," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 343–350. [Online]. Available: http://www.aclweb.org/anthology/D/D07/D07-1036

[6] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," in *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 128–135. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0717

[7] P. Koehn and J. Schroeder, "Experiments in domain adaptation for statistical machine translation," in *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 224–227. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0733

[8] J. Gao, J. Goodman, M. Li, and K.-F. Lee, "Toward a unified approach to statistical language modeling for chinese," *ACM Transactions on Asian Language Information Processing*, vol. 1, pp. 3–33, March 2002. [Online]. Available: http://doi.acm.org/10.1145/595576.595578

[9] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the ACL 2010 Conference Short Papers*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 220–224. [Online]. Available: http://www.aclweb.org/anthology/P10-2041

[10] S. Matsoukas, A.-V. I. Rosti, and B. Zhang, "Discriminative corpus weight estimation for machine translation," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, August 2009, pp. 708–717. [Online]. Available: http://www.aclweb.org/anthology/D/D09/D09-1074

[11] G. Foster, C. Goutte, and R. Kuhn, "Discriminative instance weighting for domain adaptation in statistical machine translation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: Association for Computational Linguistics, October 2010, pp. 451–459. [Online]. Available: http://www.aclweb.org/anthology/D10-1044

[12] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, July 2011, pp. 355–362. [Online]. Available: http://www.aclweb.org/anthology/D11-1033

[13] R. Sennrich, "Perplexity minimization for translation model domain adaptation in statistical machine translation," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Avignon, France: Association for Computational Linguistics, April 2012, pp. 539–549. [Online]. Available: http://www.aclweb.org/anthology/E12-1055

[14] B. Haddow and P. Koehn, "Analysing the effect of out-of-domain data on smt systems," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 422–432. [Online]. Available: http://www.aclweb.org/anthology/W12-3154

[15] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantine, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic, June 2007, pp. 177–180.

[16] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, July 2003, pp. 160–167.

[17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, July 2002, pp. 311–318.

[18] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA, August 2006, pp. 223–231.

[19] P. Koehn, "Statistical Significance Tests for Machine Translation Evaluation," in *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, Barcelona, Spain, July 2004, pp. 388–395.

Applications of Data Selection via Cross-Entropy Difference for Real-World Statistical Machine Translation

Amittai Axelrod, QingJun Li, William D. Lewis

Microsoft Research
Redmond, WA 98052, USA
amittai@alum.mit.edu
{v-qingjl,wilewis}@microsoft.com

Abstract

We broaden the application of data selection methods for domain adaptation to a larger number of languages, data, and decoders than shown in previous work, and explore comparable applications for both monolingual and bilingual cross-entropy difference methods. We compare domain adapted systems against very large general-purpose systems for the same languages, and do so without a bias to a particular direction. We present results against real-world general-purpose systems tuned on domain-specific data, which are substantially harder to beat than standard research baseline systems. We show better performance for nearly all domain adapted systems, despite the fact that the domain-adapted systems are trained on a fraction of the content of their general domain counterparts. The high performance of these methods suggest applicability to a wide variety of contexts, particularly in scenarios where only small supplies of unambiguously domain-specific data are available, yet it is believed that additional similar data is included in larger heterogenous-content general-domain corpora.

1. Introduction

The common wisdom in SMT is that "a lot of data is good" and "more data is better". This wisdom is backed up by evidence that scaling to ever larger data shows continued improvements in quality, even when one trains models over billions of n-grams [1]. Likewise, doubling or tripling the size of tuning data can show incremental improvements in quality as well [2]. Not all data is equal, however, and the kind of data one chooses depends crucially on the target domain. In a domain-specific setting, SMT benefits less from large amounts of general domain content; rather, it benefits from more content in the target domain, even if that content is appreciably smaller then the available pool of general content [3]. This fact has become more crucial as the community involved in the application of SMT has grown larger. The extended SMT community now includes an increasing number of multinational firms and public entities who wish to apply SMT to practical uses, such as automatically translating online knowledge bases, interacting with linguistically diverse customers over IM, translating large bodies of company-internal documentation for satellite offices, or even just broadening Web presence into new markets. For these new seats at the SMT table, data is still a gating factor for quality, but it is gated across another dimension: domain. For these SMT users, the rule really is not "more data is better", but rather its corollary, "more data *like my data* is better".

In this paper, we broaden the application of data selection methods for domain adaptation to a larger number of languages, data, and decoders than shown in previous work, and explore comparable applications for both monolingual [4] and bilingual [3] cross-entropy difference methods. The languages chosen for our study are typologically diverse, consisting of English, Spanish, Hebrew and Czech. A diverse sample of languages demonstrates that factors related to data sparsity, namely morphological complexity and structural divergence (*a la* [5]), are not significant factors in the successful application of the methods.

Further, we compare domain adapted systems against very large general purpose systems, whose data forms the supply of out-of-domain data we adapt from. Showing performance gains against such large systems ([3] constitutes prior work for Chinese-English) is a much harder baseline to beat than a simple out-of-the-box installation of a standard SMT toolkit. Our gains are made appreciably harder since we treat as one baseline a large general purpose system *tuned on target domain data*. For thoroughness, we also demonstrate resilience of the methodology to direction of translation, e.g., we not only apply the method to translating English $\rightarrow$ X but also to X $\rightarrow$ English, and to the decoder chosen, e.g., we use both phrase-based and tree-to-string decoders. In all cases, we demonstrate improvements in performance for domain-adapted systems over baselines that are trained on significantly larger supplies of data (10x more).

2. Task-Specific SMT

There has been much recent interest in methods for improving statistical machine translation systems targeted to a specific task or domain. The most common approach is that of

domain adaptation, whereby a system is trained on one kind of data, and then adjusted to apply to another. The adjustment can be as simple as retuning the model parameters on a task-specific dev set, such as [6]. Another common approach is to modifying the general-domain model using an in-domain model as a guide, or enhancing an in-domain model with portions of a general domain model, such as [7] among others.

We seek to accomplish the same goal as domain adaptation techniques, only by using the available data more effectively instead of modifying the model's contents. A *data selection* method is a procedure for ranking the elements of a pool of sentences using a relevance measure, and then keeping only the best-ranked ones. These data selection methods make binary decisions – keep or discard – but there are also soft-decision approaches, termed *instance weighting*.

Data selection methods have been used for some time in other NLP applications such as information retrieval (IR) (using tf-idf) and language modeling (using perplexity). One focus for those applications is mixture modeling, wherein data is selected to build sub-models, which are then weighted and combined into one larger model that is domain-specific [8]. These approaches were later combined by [9] and [10] to apply IR methods for build a translation mixture model using additional corpora. A different way of using all the available data yet highlighting its more relevant portions is to apply instance weighting. The main difference is that only one model is trained, rather than building multiple models and interpolating them against some held-out data. Experiments by [11] and [12] modified the n-gram counts from each sentence according to their relevance to the task at hand.

Moving away from mixture models, perplexity is commonly used as a selection criterion, such as by [13], to select additional training data for expanding a single in-domain language model. This method has the advantage of being extremely simple to apply: train a language model, score each additional sentence, and select the highest-ranked. This was applied to SMT by [14]. The main idea was repurposed by [4] to rank each additional sentence s by the *cross-entropy difference* between an in-domain language model and an LM trained on all of the additional data pool:

$$\underset{s \in POOL}{\text{argmin}} \quad H(s, LM_{IN}) - H(s, LM_{POOL})$$

The optimal selection threshold must be determined via grid search, but it is otherwise straightforward to apply. The cross-entropy difference criterion was first applied to the task of SMT by [3]. They also proposed a bilingual version of the criterion, consisting of the sum of the monolingual cross-entropy difference scores for two languages $L1$ and $L2$:

$$\begin{aligned}\underset{s \in POOL}{\text{argmin}} \quad & [H_{L1}(s, LM_{IN}) - H_{L1}(s, LM_{POOL})] \\ & +[H_{L2}(s, LM_{IN}) - H_{L2}(s, LM_{POOL})]\end{aligned}$$

Both the monolingual and bilingual versions have been used in recent SMT work, such as by [15] on Arabic-English and French-English, [16] for German-English and French-English systems, and in previous IWSLT evaluations for Chinese-English by [17] among others.

3. Effectiveness of Cross-Entropy Difference as a Data Selection Method

Our goal is to provide a more comprehensive survey of the impact of cross-entropy difference as a selection method for SMT. Cross-entropy difference has been shown to improve performance on domain-specific tasks, but to date the published work has focused on highly-constrained targets, such as IWSLT 2010 BTEC/DIALOG tasks and moderately-sized additional data (Europarl, UN corpora). The 2012 IWSLT TED talks are more realistic, as is the Gigaword corpus as a data pool. However, the TED talks exhibit great topical variety without a unifying domain. In this work we go further and provide experimental results on a broader, yet domain-specific, task and a much larger set of data to select from. As a result, we are in a position to evaluate the effectiveness of cross-entropy difference against a very large general-purpose statistical machine translation system, and examine the cases in which data selection may help. We also compare the relative effectiveness of the monolingual and bilingual versions of cross-entropy difference. We consequently built systems on three typologically diverse language pairs (Spanish/English, Czech/English, and Hebrew/English), in both translation directions. These corpora vary greatly in the amount of general bilingual training data available and the amount of bilingual in-domain data. Furthermore, we use two kinds of SMT systems to determine whether the system improvements depend on the flavor of SMT system used.

4. Experimental Setup

We used custom-built phrase-based and tree-to-string (T2S) systems for training the models for our engines. Our T2S decoder requires a source-side parser, and was used for all language pairs where the source had a parser: for all English → X pairs, as well as for Spanish → English. Lacking parsers for Czech and Hebrew, we used our custom built phrase-based decoder (functionally equivalent in many respects to the popular Moses phrase-based decoder [18]) to train the Czech → English and Hebrew → English systems.

For all English → X systems, we trained a 5-gram LM over all relevant monolingual data (the target side of the parallel corpus). Target side LMs for all X → English systems also used 5-gram LMs, trained over the target side of parallel data. For a subset of the systems in our study, we trained a second much larger 5-gram English language model over a much larger corpus of English language data (greater than 10 gigawords), including Web crawled content, licensed corpora (such as LDC's Gigaword), etc. We used Minimum Error Rate Training (MERT) [19] for tuning the lambda values for all systems, and results are reported in terms of BLEU score [20] on lowercased output with tokenized punctuation.

For the English → Spanish systems we trained a 5-gram LM, similar to that used for English, that is, one trained over Web crawled content, licensed corpora, and other sources. This LM was greater than 5 gigawords. For the equivalent English→Czech and English→ Hebrew systems, we built an additional 5-gram LM trained on the target side of the general purpose systems.

The bilingual general-purpose training data varied significantly between language pairs, reflecting the inconsistent availability of parallel resources for less common language pairs. As a result, we had 25 million sentences of parallel English-Spanish training data, 11 million sentences for Czech-English, and 3 million sentence pairs for Hebrew-English. In all cases these are significantly more data than has been made available for these language pairs in open MT evaluations, so this work addresses in part the question of how well the cross-entropy difference-based data selection methods scale.

Our target task is to translate travel-related information as might be written in guidebooks, online travel reviews, promotional materials, and the like. Note that this is significantly broader than much previous work in the travel domain, such as pre-2011 IWSLT tasks targeting conversational scenarios with a travel assistant. Our in-domain data for the Spanish-English language pair consisted of online travel review content, manually translated from English into Spanish (using Mechanical Turk), and a set of phrasebooks between English and Spanish. The total parallel in-domain content consisted of approximately 4 thousand sentences, which was strictly used for tuning and testing. For the monolingual selection methods, we used a corpus of online travel content in English, travel guidebooks, and travel-related phrases. This corpus consisted of approximately 600 thousand sentences.

For Czech-English and Hebrew-English we used translated travel guidebooks, consisting of 129k and 74k sentences (2.1m words and 1.2m words), respectively. The monolingual methods for these two language pairs, unlike Spanish-English, used the English side of the Czech-English and Hebrew-English guidebook (respectively). For these two language pairs we can therefore directly compare the monolingual and bilingual data selection methods. The held-out development and test sets for the Spanish-English systems consisted of crowdsourced human translations of data from a travel review website. For Czech-English and Hebrew-English, we used held-out portions of the same guidebooks used for the training data.

Because our baseline comparison is against a real-world SMT system, we used additional monolingual resources to train an output-side language model, and used it in lieu of an LM trained only on the output side of the parallel training corpus. We used the same LM for all X→English systems. The large monolingual LM ("All-mono" in the tables below) consistently yielded +0.75-3 BLEU over using only the output side of the bilingual training data. We are thus able to compare the performance of translation models trained on only a subset of the parallel data vs ones trained on all the data, without having to worry about the effect of the data selection process on LM coverage, as LM size and coverage has a substantial impact on SMT system performance.

In all cases, we built the following systems:

1. A baseline using all the available bilingual data to train the translation model, and all available monolingual data in the output language to train the language model. This system is tuned on a standard non-travel dev set (*e.g. WMT2010*), and represents a baseline of a very large scale SMT system with no adaptation.
2. Another baseline using all the available bilingual data to train the translation model, and all available monolingual data in the output language to train the language model. This baseline is tuned on the travel-specific devset for the language pair. Due to the size of the corpora involved, this may be considered a difficult baseline and is also the easiest way to build a domain-specific system using an existing general SMT system, since it does not require retraining.
3. An SMT system using only the top 10% of the bilingual training corpus to train the translation model, with the language model trained on the target side of this subset. The quantity of 10% was chosen empirically as generally representative of a well-performing adapted SMT system.
4. An SMT system using only the top 10% of the bilingual training corpus to train the translation model, but with a language model trained on all available monolingual data (like the baseline systems). This is more realistic than System #3 above, as it shows the effect of just reducing the size of the phrase table training corpus, but does not affect its ability to assemble fluent output.
5. A system with one translation model and one language model trained on the top 10%, as in System #3, but with the addition of a second language model using all the monolingual data.
6. A system with one translation model and one language model trained on the top 10%, as in System #3, but with the addition of a second translation model using all the bilingual data and a second language model using all the monolingual data. This is a general-purpose SMT system that has been augmented with a domain-specific phrase table and language model, and reflects what is achievable by considering all sources of training data for task-specific performance.

5. Results

5.1. Spanish↔English Language Pair

The English-Spanish language pair is the one with the most available general-coverage parallel data: 25 million

sentences. This is 20% larger than any previous cross-entropy difference experiment (*c.f.* 21m sentence pairs for English→French in [15]). This amount of data means the large-scale translation system is reasonably strong. For example, the baseline English→Spanish BLEU score on the WMT 2010 test set is 32.21, when tuned on the WMT 2010 dev set (see Table 1). However, this is also a language pair with an extremely limited amount of parallel travel-specific data: practically none, as there is not enough to train even a language model on. In this situation, we assembled all available monolingual English travel data (consisting of the English half of bilingual travel data for other language pairs) and used it exclusively to select relevant training data from the large Spanish-English corpus.

The English↔Spanish systems were tuned on 2,930 travel review sentences, and tested on 776 sentences from the same source. We used an additional 992 travel-related sentences translated from online hotel reviews as a second test set. Of interest also is the degradation in performance of a travel-tuned system on non-travel data, so we evaluated all the systems on the WMT2010 test set. Results for English→Spanish are in Table 1, and for Spanish→English are in Table 2.

Table 1 shows that by augmenting the baseline system with the translation model and language model trained on the top 10% of the training data, it is possible to gain an extra +0.3 BLEU points on the travel task, an extra +0.6 BLEU on the hotel reviews, while only losing -0.2 on the WMT task compared to just retuning the baseline system on the travel devset. Depending on the application, this may be a worthwhile tradeoff. However – and as expected – overall performance on the general WMT2010 task decreases by over a BLEU point when tuning on the travel domain. This must be taken into consideration when deciding how to use existing SMT systems for additional tasks.

The results in Table 2 are similar in story; the main difference is that the impact of corpus size for language model training is more apparent because the output language is English. Using all monolingual data instead of just the bilingual corpus to train the LM adds at least 3 BLEU points to the score of all the systems that use it; this is why we use the large LM for all but one of our experimental SMT systems.

5.2. Czech↔English Language Pair

For the Czech↔English translation pair we have less than half as much parallel general-domain text (11m sentences) than the Spanish↔English pair, however, there is substantially more bilingual in-domain text. We are therefore able to compare the effectiveness of the monolingual vs bilingual selection methods for both translation directions. For the monolingual methods we build an LM on the English half of the travel data, and for the bilingual selection method we build language models on each side and apply them as per the equation in Section 2. The un-adapted baseline system is tuned on WMT dev2010, which is 4,807 sentences in size. The travel-adapted systems were tuned on 1,984 sentences of guidebook data, and the held-out test set consists of 4,844 sentences from the same guidebook. These datasets are large enough to provide stable and representative results.

We first examine results for the English → Czech direction, tabulated in Table 3. Tuning the baseline system on travel-specific data improved performance by +0.4 on the guidebook test set, but caused a loss of -0.5 on the WMT test set. When comparing against the domain-tuned baseline, we see that the models built on data selected via the monolingual cross-entropy method always decrease performance, if only slightly. The systems trained on data selected via the bilingual criterion do slightly better, but could be described as being at best equal to the baseline on the guidebook data, but are even worse on the WMT test set. We therefore have a case where cross-entropy difference as a data selection method does not outperform simply retuning an existing system on a dev set pertaining to the new target task.

Table 4 contains results from experiments in the other direction, from Czech → English. As before, the retuned baseline system gains +1.5 on the guidebook data, but loses -2 on the WMT. The data selection results, however, differ markedly from the other translation direction, even though the selection criteria are exactly the same. Using the monolingually-selected systems we can see that using the LM trained on the selected data is slightly harmful, but that the large language model is surprisingly powerful, making a +4 BLEU impact. The selected translation mode is good for a +2 BLEU improvement on its own, and using all the models together yields a +2.8 improvement over the retuned baseline on the guidebook data, at a cost of -1.4 to the WMT test set performance. The bilingually selected methods are consistently better, but only marginally so (+0.1 BLEU).

Thus data selection methods provide substantial improvements when translating Czech → English, and none from English → Czech. Two differences between the systems are that the former is a phrasal MT system, and the latter is a treelet translation system. Furthermore, the output language model is significantly better when translating into English than into Czech, simply due to the differing amounts of LM training data.

5.3. Hebrew↔English Language Pair

Our Hebrew↔English translation pair has the least amount of parallel training data of the ones we tested, but still has 3 million sentences, making it larger than the Europarl corpus which is a standard for European languages. The baseline large-scale system was tuned on 2,000 sentences extracted

Table 1: *English to Spanish*

Model	Phrase Table 1	TM 2	LM 1	LM 2	Travel Reviews	Hotel Reviews	WMT 2010
Baseline	All	–	All-mono	–	33.27	28.19	31.00
Baseline (WMT2010)	All	–	All-mono	–	32.28	**29.09**	**32.21**
Top 10% TM, All-mono LM	Top 10%	–	All-mono	–	32.78	28.09	28.07
Top 10% only	Top 10%	–	Top 10%	–	32.61	27.25	25.60
+All-mono LM	Top 10%	–	All-mono	Top 10%	33.12	28.18	28.19
+ All TM	Top 10%	All	All-mono	Top 10%	**33.55**	28.80	30.81

Table 2: *Spanish to English*

Model	Phrase Table 1	TM 2	LM 1	LM 2	Travel Reviews	Hotel Reviews	WMT 2010
Baseline	All	–	All-mono	–	39.43	32.79	31.38
Baseline (WMT2010)	All	–	All-mono	–	38.71	32.03	**32.11**
Top 10% only	Top 10%	–	Top 10%	–	37.18	30.04	26.48
+All-mono LM	Top 10%	–	All-mono	Top 10%	39.49	32.38	29.57
+All TM	Top 10%	All	All-mono	Top 10%	**40.00**	**33.28**	31.05

Table 3: *English to Czech*

Model	Phrase Table 1	TM 2	LM 1	LM 2	Guidebook	WMT 2010
Baseline	All	–	All-mono	–	27.73	15.03
Baseline WMT2010	All	–	All-mono	–	*27.33*	**15.59**
Monolingual Top 10% only	Top 10%	–	Top 10%	–	24.80	12.63
Monolingual Top 10% TM, All-mono LM	Top 10%	–	All-mono	–	27.84	13.95
+ Top 10%LM	Top 10%	–	All-mono	Top 10%	27.69	13.59
+ All TM	Top 10%	All	All-mono	Top 10%	27.43	14.25
Bilingual Top 10% only	Top 10%	–	Top 10%	–	24.92	12.52
Bilingual Top 10% TM only, All-mono LM	Top 10%	–	All-mono	–	27.68	13.67
+ Top 10% LM	Top 10%	–	All-mono	Top 10%	27.77	13.48
+ All TM	Top 10%	All	All-mono	Top 10%	**27.80**	*14.88*

Table 4: *Czech to English*

Model	Phrase Table 1	TM 2	LM 1	LM 2	Guidebook	WMT 2010
Baseline	All	–	All-mono	–	34.06	21.83
Baseline (WMT2010)	All	–	All-mono	–	32.52	**23.88**
Monolingual Top 10% only	Top 10%	–	Top 10%	–	30.48	15.86
Monolingual Top 10% TM, All-mono LM	Top 10%	–	All-mono	–	34.64	19.46
+ Top 10% LM	Top 10%	–	All-mono	Top 10%	34.32	19.36
+ All TM	Top 10%	All	All-mono	Top 10%	35.36	22.40
Bilingual Top 10% only	Top 10%	–	Top 10%	–	30.64	15.90
Bilingual Top 10% TM, All-mono LM	Top 10%	–	All-mono	–	34.66	19.51
+ Top 10% LM	Top 10%	–	All-mono	Top 10%	34.55	19.38
+ All TM	Top 10%	All	All-mono	Top 10%	**35.48**	22.15

from the results of web queries. The travel domain data, like for Czech↔English, consists of travel guidebooks. We held out 1,979 sentences as a development set, plus an additional 4,764 sentences as a stable test set. We also report results on the WMT 2009 test set, so as to provide a comparison with other published work in SMT.

The results for translating from English→Hebrew are shown in Table 5. Retuning the baseline general-domain system on the travel dev set increases the BLEU score on the guidebook test set by +0.4, at a cost of -0.3 on the WMT 2009 set. There is not much difference in the results from selecting the best 10% of the general training corpus with the monolingual vs bilingual cross-entropy difference. In both cases, adding an LM trained on the selected data does no better than just using the largest LM possible. However, just using the most relevant data for a translation model provides a slight improvement (+0.3), and augmenting the baseline system with models trained on just the best selected data provide a total improvement of +1 BLEU on the guidebook test set. The only difference between the monolingual and bilingual versions of the selection criterion is that the best monolingually-selected system loses only -0.1 BLEU on the unrelated WMT 2009 test set, compared to -0.7 with the bilingually-selected equivalent.

Results for data selection for Hebrew→English systems can be found in Table 5. Retuning the existing large-scale baseline system provides a +0.4 increase on the guidebook test set, and a +0.1 improvement on the WMT set. The latter is slightly unexpected. However, using cross-entropy difference to augment the SMT system provides a total improvement of almost +1 BLEU.

In general, the systems selected by monolingual cross-entropy difference do the same as their counterparts picked using bilingual cross-entropy difference, if not marginally better. Unlike in the previous translation direction, replacing the general-domain phrase table with one built on the most-relevant 10% of the training data generally made things slightly worse. Only augmenting the general system with the models trained on the selected subsets improved performance over the retuned baseline. As before, the gain of +0.7 BLEU on the guidebook test set was offset by a loss of -0.2 to -0.5 on the WMT 2009 test set.

6. Analysis

Generally, the difference between monolingual-on-English side and bilingual cross-entropy difference was minor. This is in contrast to prior work on Chinese→English, which suggested that the bilingual method was notably better [3]. One key difference between that work and this one is that they tested monolingual methods on the input side, namely Chinese. In this work the monolingual method was always computed using the English language, regardless of whether it was input or output. It may simply be that the monolingual cross-entropy difference score is sufficient, if the language used for the selection criterion is capable of being well-represented by an n-gram model by virtue of having simpler morphology or lesser long-range dependencies than the other member of the language pair. When it is unclear which of the two languages is better suited, then the bilingual cross-entropy method is a safe choice, as it provides generally the same effectiveness and does not seem to do any harm. That said, the experiments on Spanish↔English confirm prior work that bilingual in-domain data is not strictly necessary to adapt an SMT system to a target task.

Only one translation direction English↔Czech showed no need for data selection. In that particular case, the same improvement could be obtained by simply retuning the existing general-purpose system. However, Czech is the most morphologically complex of the languages used in this work and one could argue that it therefore suffers more from n-gram sparsity than other languages when trying to build a translation or language model on a corpus of a specific size. That the average English↔Czech system score was 7 BLEU points lower than the reverse translation direction points to the difficulty of translating into Czech. Perhaps the optimal number of sentences to select is substantially larger than for other language pairs, and so that 10% of the data could produce a system equally good as a system on the full data simply means if 20 or 30% of the data were selected then one might see a significant improvement beyond that baseline.

The overall scores for translating Hebrew↔English were the lowest, presumably due to morphological complexity coupled with the least amount of training data. Nonetheless, the gains from domain adaptation via data selection were still large in both directions. The systems trained on data selected with bilingual cross-entropy difference performed similarly on the guidebook test set as the ones trained on monolingually-selected data. However, the bilingually-selected systems performed slightly worse on the WMT 2009 test set, raising the same question as English↔Czech: how much of a morphologically rich language can be usefully captured by an n-gram language model trained on a small in-domain corpus?

Interestingly, translating into English was always improved using data selection methods. This is somewhat counterintuitive, as the larger output-side language model might be assumed to mask changes to the other components of the SMT system, much as a larger language model is assumed to always improve translation output. Furthermore, reducing the size of the language model always hurt significantly, and the best systems always included the largest LM. This may indicate that it is less important to adapt the language model than it is to provide more domain-accurate phrase tables.

In most cases, the performance improvement on the travel task of a task-specific SMT system was greater than the performance loss on the regular test set (*e.g.* WMT test

Table 5: *English to Hebrew*

Model	Phrase Table 1	TM 2	LM 1	LM 2	Guidebook	WMT 2009
Baseline	All	–	All-mono	–	12.45	14.53
Baseline ReqLog	All	–	All-mono	–	12.04	**14.88**
Monolingual Top 10%	Top 10%	–	Top 10%	–	10.37	10.17
Monolingual Top 10% TM only	Top 10%	–	All-mono	–	12.79	11.75
+All-mono LM	Top 10%	–	All-mono	Top 10%	12.77	11.57
+ All TM	Top 10%	All	All-mono	Top 10%	**13.46**	14.43
Bilingual Top 10%	Top 10%	–	Top 10%	–	10.33	10.01
Bilingual Top 10% TM only	Top 10%	–	All-mono	–	12.88	11.55
+All-mono LM	Top 10%	–	All-mono	Top 10%	12.80	11.66
+ All TM	Top 10%	All	All-mono	Top 10%	**13.49**	13.84

Table 6: *Hebrew to English*

Model	Phrase Table 1	TM 2	LM 1	LM 2	Guidebook	WMT 2009
Baseline	All	–	All-mono	–	18.58	**25.18**
Baseline ReqLog	All	–	All-mono	–	18.18	25.03
Monolingual Top 10%	Top 10%	–	Top 10%	–	16.47	16.08
Monolingual Top 10% TM only	Top 10%	–	All-mono	–	18.13	19.36
+All-mono LM	Top 10%	–	All-mono	Top 10%	18.17	19.54
+ All TM	Top 10%	All	All-mono	Top 10%	**19.12**	24.92
Bilingual Top 10%	Top 10%	–	Top 10%	–	16.46	16.15
Bilingual Top 10% TM only	Top 10%	–	All-mono	–	18.09	19.16
+All-mono LM	Top 10%	–	All-mono	Top 10%	18.20	18.85
+ All TM	Top 10%	All	All-mono	Top 10%	**19.05**	24.77

2010). This implies that the trade-offs between performance on two distinct targets are not unbounded: one rarely loses more than one gets. Thus one may make an informed decision as to whether domain adaptation is worth while by comparing against acceptable drops in performance on other tasks of interest.

Finally, despite half of the translation systems being built using phrase-based SMT and the other half with syntactic/treelet systems, this does not seem to have an obvious impact on the appropriateness of data selection methods for improving in-domain performance.

7. Conclusions

We have presented a broader survey of tailoring a general translation system to a target task by selecting a subset of the training data using cross-entropy difference. We performed experiments in both translation directions for three language pairs. These language pairs exhibit varying levels of morphological complexity, amounts of parallel general-purpose data, and amounts of parallel in-domain data. We systematically compared methods of using the selected training data against real-world baselines consisting of very large general-purpose SMT systems using all available additional monolingual resources for language models, and show gains over these baselines of +0.3/1.3 BLEU for Spanish↔English, +0.5/3.0 for Czech↔English, and +0.7/1.4 for Hebrew↔English. These results confirm all prior work showing that only a fraction of general-purpose data is needed for a task-specific SMT system of at least equivalent performance on the domain of interest. We have also shown how domain adaptation adversely affects performance on non-domain-specific tasks, but the results also indicate that the loss in performance on a general task is often less than the improvement on the domain of interest, both quantifying and arguably justifying the tradeoff.

8. Acknowledgements

We gratefully acknowledge the assistance of Marco Chierotti in acquiring the crowdsourced translations of the travel domain data.

9. References

[1] Brants, T., Popat, A., Xu, P., Och, F., Dean, J. “Large Language Models in Machine Translation.” EMNLP (Empirical Methods in Natural Language Processing). 2007.

[2] Koehn, P,. and Haddow, B. “Towards Effective Use of Training Data in Statistical Machine Translation.” WMT (Workshop on Statistical Machine Translation). 2012.

[3] Axelrod, A., He, X., and Gao, J. “Domain Adaptation via Pseudo In-Domain Data Selection”. EMNLP (Empirical Methods in Natural Language Processing). 2011.

[4] Moore, R. C. and Lewis, W. “Intelligent Selection of Language Model Training Data”. ACL (Association for Computational Linguistics). 2010.

[5] Dorr, B. “Machine Translation Divergences: A Formal Description and Proposed Solution.” ACL (Association for Computational Linguistics). 1994.

[6] Li, M., Zhao, Y., Zhang, D., and Zhou, M. “Adaptive Development Data Selection for Log-linear Model in Statistical Machine Translation”. COLING (International Conference on Computational Linguistics). 2010.

[7] Bisazza, A., Ruiz, N., Federico, M. “Fill-Up versus Interpolation Methods for Phrase-Based SMT Adaptation”. WMT (Workshop on Statistical Machine Translation). 2011.

[8] Iyer, R., Ostendorf, M., and Gish, H. “Using Out-of-Domain Data to Improve In-Domain Language Models”. IEEE Signal Processing Letters. 4(8):221-223. 1997.

[9] Lu, Y., Huang, J., and Liu, Q. “Improving Statistical Machine Translation Performance by Training Data Selection and Optimization.” EMNLP (Empirical Methods in Natural Language Processing). 2007.

[10] Foster, G., and Kuhn, R. “Mixture-Model Adaptation for SMT”. WMT (Workshop on Statistical Machine Translation). 2007.

[11] Matsoukas, S., Rosti, A.-V., and Zhang, B. “Discriminative Corpus Weight Estimation for Machine Translation.” EMNLP (Empirical Methods in Natural Language Processing). 2009.

[12] Foster, G., Goutte, C., and Kuhn, R. “Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation.” EMNLP (Empirical Methods in Natural Language Processing). 2010.

[13] Gao, J., Goodman, J., Li, M., and Lee, K.-F. “Toward a Unified Approach to Statistical Language Modeling for Chinese”. ACM Transactions On Asian Language Information Processing. 1(1):333. 2002.

[14] Yasuda, K., Zhang, R., Yamamoto, H., and Sumita, E. “Method of Selecting Training Data to Build a Compact and Efficient Translation Model.” IJCNLP (International Joint Conference on Natural Language Processing). 2008.

[15] Mansour, S., Wuebker, J., and Ney, H. “Combining Translation and Language Model Scoring for Domain-Specific Data Filtering.” IWSLT (International Workshop on Spoken Language Translation). 2011.

[16] Banerjee, P., Kumar, S., Roturier, J., Way, A., van Genabith, J. “Translation Quality-Based Supplementary Data Selection by Incremental Update of Translation Models”. COLING (International Conference on Computational Linguistics). 2012.

[17] He, X., Axelrod, A., Deng, L., Acero, A., Hwang, M.-Y., Nguyen, A., Wang, A., and Huang, X. “The MSR System for IWSLT 2011 Evaluation”. IWSLT (International Workshop on Spoken Language Translation). 2011.

[18] Koehn, P., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Moran, C., Dyer, C., Constantin, A., and Herbst, E. “Moses: Open Source Toolkit for Statistical Machine Translation”. ACL (Association for Computational Linguistics) Interactive Poster and Demonstration Sessions. 2007.

[19] Och, F. “Minimum Error Rate Training in Statistical Machine Translation.” ACL (Association for Computational Linguistics). 2003.

[20] Papineni, K., Roukos, S., Ward, T., and Zhu, W. “BLEU: a Method for Automatic Evaluation of Machine Translation”. ACL (Association for Computational Linguistics). 2002.

[21] Gascó, G., Rocha, M.-A., Sanchis-Trilles, G., Andrés-Ferrer, J., and Casacuberta, F. “Does More Data Always Yield Better Translations?” EACL (European Association for Computational Linguistics). 2012.

[22] Federico, M. “Language Model Adaptation through Topic Decomposition and MDA Estimation.” ICASSP (International Conference on Acoustics, Speech, and Signal Processing). 2002.

[23] Quirk, C., and Menezes, A. “Dependency Treelet Translation: The Convergence of Statistical and Example-Based Machine Translation?” Machine Translation. 20:43-65. 2006.

[24] Quirk, C. and Moore, R. “Faster Beam-Search Decoding for Phrasal Statistical Machine Translation”. Machine Translation Summit XI. 2007.

[25] He, X., and Deng, L. “Robust Speech Translation by Domain Adaptation.” Interspeech. 2011.

A Universal Approach to Translating Numerical and Time Expressions

Mei Tu Yu Zhou Chengqing Zong

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
{mtu,yzhou,cqzong}@nlpr.ia.ac.cn

Abstract

Although statistical machine translation (SMT) has made great progress since it came into being, the translation of numerical and time expressions is still far from satisfactory. Generally speaking, numbers are likely to be out-of-vocabulary (OOV) words due to their non-exhaustive characteristics even when the size of training data is very large, so it is difficult to obtain accurate translation results for the infinite set of numbers only depending on traditional statistical methods. We propose a language-independent framework to recognize and translate numbers more precisely by using a rule-based method. Through designing operators, we succeed to make rules educible and totally separate from codes, thus, we can extend rules to various language-pairs without re-coding, which contributes a lot to the efficient development of an SMT system with good portability. We classify numbers and time expressions into seven types, which are Arabic number, cardinal numbers, ordinal numbers, date, time of day, day of week and figures. A greedy algorithm is developed to deal with rule conflicts. Experiments have shown that our approach can significantly improve the translation performance.

1. Introduction

Recently, statistical machine translation (SMT) models, especially the phrase based translation models [1], have been widely used and have achieved great improvements. However, there are still some hard problems. One of them is how to translate OOV words. Among all OOV words, the numerical and time expressions (we generally call numbers hereafter) are typically and widely distributed in some corpora. According to our rough statistics in a corpus of travelling domain, there are about 15 percent sentences containing numbers in all 5000 sentences. Theoretically, numbers are innumerable and the forms of numbers vary greatly from universal Arabic numbers to language-dependent number words. For example, “1.234 kg” is an Arabic number with units, the English expression “nineteen eighty-five” consists of cardinal number words, while “1.345 million” is a combination of Arabic number and cardinal number word. Due to the non-exhaustive characteristics and variability of numbers, translating numbers in the traditional SMT framework often suffers from the OOV problem even when the size of training data is very large. Thus we have to seek an efficient way to develop a new module for recognizing and translating of numbers (RTN).

According to the characteristics of numbers, it is intuitive to do RTN work through a framework with rules [2]. Traditionally, rules always depend on the specific languages they are applied to. Researchers have to build specific rule-based framework for each language-pair, thus resulting in low efficiency. Moreover, when the source or target language changes, codes are required to be rewritten accordingly. It costs much time to transplant rules. Considering that RTN is very important for text translations among all languages, we address on designing a uniformed framework to solve the RTN problem.

Based on the analysis above, in this paper we propose a language-independent rule-based approach for RTN. The proposed approach has been successfully applied and verified on bidirectional translation of Chinese-English and other language pairs. The experimental results give a much positive evidence of our work.

The remainder of this paper is organized as follows: Section 2 describes the definition of rules and symbols. Section 3 presents how to apply the rules to recognize and translate numbers. Our experimental results and analysis are presented in Section 4. Section 5 introduces related work. Finally, we give concluding remarks and mention our future work in Section 6.

2. Rules definition

Even though forms of numbers are various, the written manner and usage of number are relatively standardized. When we construct rules, such characteristics contribute a lot, and we also refer to some pervious work on rule-based systems [3-8]. In this section, we will give the details of the definition of the translation rules.

2.1. Overview of the rule-based framework

To depict our RTN module clearly, we use Figure 1 to illustrate the components of rules and how they guide the recognition and translation process.

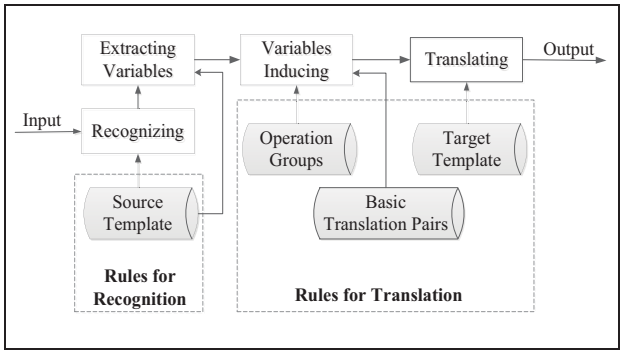


Figure 1: Rules and the workflow of RTN module

As seen in Fig.1, the first step of our module is to recognize numbers in an input sentence under the guide of the database of *Source Template*, which is in forms of regular expressions. *Source Template* consists of variables to be transformed and constants working as anchor words. After recognition, the variables will be used for inducing which is in fact a translating procedure with the assistance of *Operation Groups*

and *Basic Translation Pairs*. *Operation Groups* contain a variety of operations governing the procedure of variable inducing, while the *Basic Translation Pairs* are those translations pairs frequently used. At the final stage of our module, which is after inducing, the *Target Template* will determine the word order for each translated fragment. In order to give clearer explanation of the workflow of our module, we take "he will arrive on the 15th of May" as the input sentence and the Chinese as output language for example. At the first step, "15th of May" will be recognized by our module. And "15th" and "May" are regarded as variables, while "of" are constants. In the stage of inducing, "15th" is transformed to "十五"(fifteenth) and "May" is transformed to "五月"(May) by a series of operations. At last, we reconstruct the transformed variables to the final translation "五月 十五 号 ". In summary, *Source Template*, *Target Template, Operation Group* together with *Basic Translation Pairs* form a rule.

By observing many instances of numbers, we group numbers into seven categories. Rules will be created for each category. The categories and components of rules are described in details in the following sub-sections.

2.2. Types of number

According to the characteristics of numbers, we classify them into seven common used types as follows:

- ***Arabic number***: Arabic numerals are most widely used for counting and measuring in many languages such as Indo-European languages and Chinese. We give some examples of them in Table 1, as well as the following types.
- ***Cardinal number***: Beside Arabic number, there is also another totally different written system of numbers in many languages. Different with Arabic numbers, it is language-dependent. For example, in English, we use "one, two,…., hundred, thousand, …" to represent numbers. In addition, we also put numbers which combine cardinal numbers and Arabic numbers into this type.
- ***Ordinal number***: It represents the rank of something related with the order or position. We put them into a different group from the two types of numbers above because its written form differs from the Arabic and Cardinal numbers in many languages.
- ***Date***: The day, month, and year are always in a fixed expression.
- ***Time of day***: The time of the day often contains following several common types, "XX:XX", time expression in Arabic numbers, in cardinal numbers or the combination of Arabic and cardinal numbers.
- ***Day of week***: It includes words or expressions that represent Monday to Sunday. In some languages, like Chinese, there are several ways to represent them.
- ***Figures:*** Other numbers except above are put in to this group, such as telephone numbers, room numbers, and numbers of product labels.

Table 1: Number examples of types above

Types	Instances
Arabic Number	3.1415 ; 100,000 ; 50%
Cardinal Number	six hundred and eighty-three; 11.3 million; 一千二百(one thousand two hundred);
Ordinal Number	twenty-first ; 第二(the second)
Date	September 3rd ; eighth of August, 2008; 2000 年 1 月 1 号(January 1st, 2000)
Time of day	twelve o'clock ; half past ten a.m. ;7:00; 早八点 (eight a.m.); 八点半(8:30)
Day of week	Monday; Sunday; 星期二(Tuesday); 周六(Saturday)
Figures	telephone number one o o one one two two six ; 幺九二八 (one nine two eight)

2.3. Source template

2.3.1. Regular expression for number recognition

In many sequence searching tasks, regular expressions are chosen to match a certain sequence, for their linear complexity and simplicity. So we adopt it to recognize numbers. For example, in an English text, the regular expression for any day of May is written as follows,

Eg.1:

" (1|2|3){0,1}(1st|2nd|3rd|[4-9]th) of (May)"

We can easily extend the above regular expression to recognize date in other months by adding the alternatives of "May".

One of the most centered questions in recognition is whether the coverage of the regular expression is precise as well as complete. There are three cases in our experiments. Let us use R to represent the real coverage of the regular expression we write, and S to represent the coverage it aims to have. Then we describe the three cases as follows.

Firstly, in most cases, R is exactly equal to S, so we can easily write the regular expression to match numbers such as the double-figure numbers.

Secondly, there are exceptions that $R \succ S$, which means that the sequence extracted by our source template is not a numerical expression that we expect to get, even if it matches our template. For example, the word "second" has two kinds of common meaning. One is the ordinal form of "two" which is an ordinal number, while the other is a unit of time, like "per second", where "second" is not used as a number. Therefore, if there is no explicit anchor word in the surrounding context, like "the second day", to indicate that "second" is an ordinal number, we keep it unrecognized.

The third case is pseudo unequal. Take the regular expression in Eg.1 for example. Our purpose is to match the month-day sequence, which is of course from the first day to the last day in May. But this pattern includes not only 31 days, but also the 32nd to 39th.So if there was "on the 32nd of May" in the text, it would be captured by that pattern. However, "on the 32nd of May" is against common sense, and merely appears in the language, thus we regard this kind of inequality as pseudo inequality and ignore it.

From the analysis above, we conclude that the only difficulty of using regular expression for searching lies in the second situation. In order to ensure the accuracy of our rules, it

is necessary to add more surrounding context to the regular expression.

2.3.2. Variables and constants

After the recognition process has been finished, the next step is to extract variables from the recognized sequences. To distinguish variables and constants clearly, we use brackets "()" which is compatible with the original regular expression to enclose the sequence of variables.

In this paper, we call the sequence enclosed in brackets "Var_N", in which "N" is the rank number. Then a recognized sequence can be divided into variable sequences and constant sequences. Parts of variables are used for being induced in the next stage, and they are what we care most for. We rewrite the Eg.1 pattern in section 2.3.1 as this,

Var_2

((1|2|3){0,1})(1st|2nd|3rd|[4-9]th) of (May)

Var_1 *Var_3* *Var_4*

where the variables are marked underlined. Only Var_1, Var_3, and Var_4 will be transformed in the next stage.

2.4. Target template

For each rule, a target template and a source template are built in pairs. And the target template is also constructed with variables and constants, which determine the final translation directly. For example,

Source : ((1|2|3){0,1})(1st|2nd|3rd|[4-9]th) of (May)

Var_1 *Var_3* *Var_4*

Target: "Var_4 Var_1Var_3 号"

Figure 2: An example of source template and target template pair

Given the source template above, we can write a corresponding target template to convey the same meaning as the source side. The variable in the target template will be replaced with its representing sequence in the final stage of the translation, i.e. we will replace Var_4 with the Chinese translation of "May", similar to Var_1 and Var_3.

2.5. Basic translation pairs

Basic translation pairs provide translations of basic units frequently used. Take the translation of English to Chinese for example. Fig. 3 shows some examples of the basic translation pairs. Each pair is in form of "<A>/<B>", which means sequence A in source side will be translated into B in our RTN module.

We build an index at the beginning of each group to make it clearer and easier to search. The index consists of rule indexes and group number, like "<Date><#1>" which represents the first group of basic translation pairs of Date numbers. Note that the translation pairs we show in Fig.3 can depends on concrete situations, such as the pair "<1>/< 十 >" in "<Arabic><#2>". The Arabic number "1" is actually translated into "一"(one) in Chinese, but when "1" is at the decade position like "12,13,14 …", we use "十"instead of "一" and translate the numbers into "十二,十三,十四…"(twelve, thirteen, fourteen …) in Chinese.

```
<Date><#1>
        <January> /<一 月>
        <February>/<二 月>
        … …
        <May>/<五 月>
        … …
        <December>/<十二 月>
<Arabic><#2>
        <1>/<十>
        <2>/<二十>
        … …
        <9>/<九十>
<Ordinal><#1>
        <1st>/<一>
        <2nd>/<二>
        … …
        <9th>/<九>
<Cardinal><#1>
        <one>/<一>
        <two>/<二>
        … …
        <nine>/<九>
```

Figure 3: Basic translation pairs for each type

2.6. Operation groups

In order to do variable inducing from the source template to the target template, we define a series of operations for variables, which make our templates dynamic and educible, compared to the traditional static methods. Educible templates own the advantage that the rule-makers need to only care about the template and operations, instead of how to make the rules work in codes.

An operation has three terms: a subject variable, an operator, and an object. Its form is designed as,

@Subject_Var_N+Operator+Object

where "@" is a hint symbol to indicate which variable will be transformed. **Subject_Var_N** is an element of {*Var*}, while **Object** can be one of the following forms, the index of a basic translation pair, or a variable, or a sequence of words, which depends on the different operators. In the following, we list all the operators in detail,

- Terminate (T): It is an end mark, which means that all the operations are terminated.
- Join (J): the subject variable will be joined with the object variable. The object can be either another variable or a sequence of words. After joining, the new sequence becomes the subject variable.
- Replace (R): if the object is the index of a basic translation pair, the subject variable will be replaced with its translation. If the object is a sequence of words, then the subject variable is thus replaced with the word sequence.
- Replace Continuously (RC): it is similar to Replace, but the subject variable will be replaced word by word instead of as a whole sequence.

We give some examples with their explanations for each operator in Table 2.

Table 2: The Example of operators

Symbol	Example
T	@Var_1+T+NULL (No operation will be applied to the variable one)
J	@Var_1+J+Var_2 (Var_2 will be jointed to Var_1)
R	@Var_1+R+<Cardinal><#1> (Var_1 will be replaced by the translation given in the basic translation pairs of the index <Cardinal><#1>)
RC	@Var_1+RC+<Cardinal><#1> (Each word of Var_1 will be replaced by the translation given in the basic translation pairs of the index <Cardinal><#1>)

All the operators we define above own two features. First, the result of a piece of operation should still be a variable, which we call it "completeness". Second, the two-argument operator is of non-commutativity. That is why we call the arguments "subject" and "object". Operators are extendable, and we can define many other operators in theory. But in our experiment, the four operators above are enough for inducing in most cases.

After defining the operators, we can transform variables. We use the Eg.1 in section 2.3.1 to explain how the operations work. As the source template for recognition is "((1|2|3){0,1})(1st|2nd|3rd|[4-9]th) of (May)", we write the following operations to transform variables,

@Var_1+R+<Arabic><#2> (1)
@Var_3+R+<Ordinal><#1> (2)
@Var_4+R+<Date><#1> (3)

The Operation (1) translates the decade number of the day to its cardinal form in Chinese. Operation (2) translates the number under 10 to its Chinese expression. At last, the month expressions are transformed to Chinese by Operation (3). After these three operations, all English numbers are translated into Chinese. After that, given the target template as "Var_4 Var_1Var_3 号", we will obtain the Chinese month-day expression finally.

If there is a sentence "he will arrive on the 15th of May" to be dealt with, then the interim results and the final result can be listed as follows,

After "on the 15th of May" is captured by the recognition pattern, the variable inducing starts:

(1): "1" is replaced by "十",
(2): "5th" is replaced by "五"
(3): "May" is replaced by "五 月"

The final Chinese result is "五 月 十 五 号" after substitutions for the variables in the target template.

Several translations for one source sequence are allowed, for which we can design several groups of operations for one recognition pattern. For example, the source sequence " on the 15th of May" can be translated to another kind of expression "5 月 15 日". We only need to put a separator between two groups of operations to let the system know that there is more than one group of transformed operations. Here we use a semicolon as the separator, and two continuous semicolons as the end of all groups of operations.

In the next section, we will describe the matching and integrating strategies.

3. Matching and integrating strategy

When the rules are put into use, the first thing we should care about is how to alleviate the rule conflicts, which is an important problem to use the rules in current SMT systems. In this section, we will describe our strategies in details.

3.1. Matching strategy

Generally speaking, the matching conflicts are caused by two problems: one lies on the inconsistency with tokenization, the other comes from the rule system itself.

As stated above, the recognition pattern on the source side is the regular expression, which is sensitive to the written formats. Consequently, some changes to the expressions or word segmentation in the source text may lead to a different matching result. Some languages, such as Chinese, suffer from the inconsistency of segmentation standard. So for such languages, we have to make our rules as flexible and robust as possible, by adding some alternative spaces. For example, "[0-9][[:space:]]?号" is more capable than "[0-9]号".

For the second problem, when the sequences captured by multiple rules overlap, optimization for the best choice is needed. Let us describe them mathematically. When we use patterns to recognize number sequences in one sentence, we will obtain a group of sequences grouped as {S} which contains *m* elements (sequences), and the corresponding patterns are grouped as {P} with *m* elements too. Among the *m* elements of {S}, *n* of them are under the condition that any one of the n elements overlaps with at least another one of them. Then we say that the n elements are "in conflict". From {S}, there is always a maximum sub set $\{S'\}$ with n elements in conflict, and we re-write the *n* elements as $S'_0, S'_1, ..., S'_{n-1}$, and the corresponding patterns are $P'_0, P'_1, ..., P'_{n-1}$. Then we address the optimization problem as follows,

$$Opt.\begin{cases} C = Max\{\bigcup_{i=0}^{n-1} C_i\} \\ R = Min\{\sum_{j=0}^{n-1} R_j\} \\ O = Min\{\sum_{k<l,l=1}^{n-1} O_{kl}\} \end{cases} \quad (1)$$

Where C_i is the coverage of S'_i, and $R_j = 1$ if S'_j is chosen, otherwise $R_j = 0$. If S'_k and S'_l are both chosen and overlapping, $O_{kl} = 1$, otherwise equals to zero. Our ultimate goal is to cover the longest sequence with fewest rules and fewest overlaps. Thus we adopt three optimization sub-goals, and the first one is more important to us.

For the first and second sub-goals, we design an algorithm based on a greedy method, which controls the complexity in linear time. Considering the optimization of *C* and *R*, we can write the state transition function as follows,

$$f_{k+1} = \max\{f_k \bigcup C'_{k+1}\} \quad (2)$$

$$h_{k+1} = \min\{h_k + R'_k\} \quad (3)$$

where $f_k = Max\{\bigcup_{i=0}^{k} C'_i\}$, $h_k = \min\{\sum_{j=0}^{k} R'_j\}$. We only need to sort the S'_i according to the starting position (the previous word owns higher priority) and coverage length (the longer sequence owns higher priority), and then pick them in order until obtaining the maximum union.

As for the third sub-goal, we need to save the intermediate ending positions so as to allow backtracking to the former state. The pseudo codes of the matching strategy we describe here are given in Figure 4. The captured sequences which contain numbers are saved in NumberSequenceSet in Line 1. Lines 2 and 3 focus on sorting the sequences according to the priority stated in the previous paragraph. Line 4 puts sequences in conflicts into a set. Line 5 is for initialization. The main body of the greedy algorithm is shown in Line 6~13, which is used for searching for the optimized set of sequences to get the widest coverage with the lowest cost (counts of sequences needed).

```
// Greedy algorithm for matching strategy
1:  NumberSequenceSet ← Recognize(srcSentence, Rule)
2:  SortForStartPosition (NumberSequenceSet)
3:  SortForCoverageLength (NumberSequenceSet)
4:  ConfSet ←  NumberSequenceSet.FilterConfront()
5:  CoverageEnd.assign(0); EndPosSet={}; FinalSet={}
6:  For each index in ConfSet:
7:      CurrentEnd = (ConfSet[index]).EndPos
8:      if CoverageEnd.value<= CurrentEnd:
9:          if StartPos(ConfSet[index]) in EndPosSet:
10:             i = EndPosSet.find(StartPos(ConfSet[index]))
11:             FinalSet.delete(i, FinalSet.size()-i)
12:             FinalSet.add(index)
13:             EndPosSet.add(CurrentEnd)
14: FilteredNumberSeqSet ← Output(FinalSet)
```

Figure 4: Pseudo Codes of the greedy algorithm for matching conflict

3.2. Integration approach

It is also a problem to integrate the number recognition and translation module (RTN module) into an SMT system. Traditionally there are three ways. One is in the preprocessing step by translating numbers before putting the source sentence into SMT, while the second way is in the post-editing step by translating numbers after the translation of SMT. Considering that the matching pattern has high requirements about the written formats, we adopt the third way which is more flexible by adding the related number translation knowledge to the translation model. Figure 5 illustrates how to merge the number translation system.

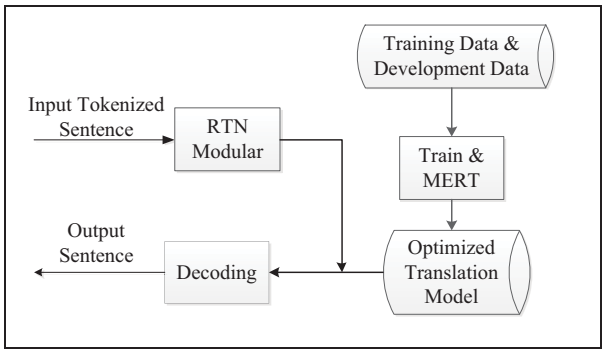


Figure 5: The systematic framework of merging RTN module into SMT

In this framework, we firstly capture the number in the input sentence and then translate those recognized numbers into target translations by the RTN module. Thus we can build a phrase-table of numbers with all the translation probabilities as 1, by considering that we definitely believe our rule-based translations of numbers. After that, the phrase-table of numbers are added into the original optimized translation model to obtain a new united table. Finally, the decoding candidates will be searched from the united table.

4. Experiments

4.1. Experiment setup

We use the IWSLT 2009 (the 6th International Workshop on Spoken Language Translation) corpus for the Chinese-English evaluation task as the bilingual training data, which includes the BTEC, CT-CE and CT-EC corpora. Because there are no test references, we randomly choose part of the development corpus as the testing set and the rest as the development set. The statistics of the training set, the development set and the testing set are listed in Table 3, 4, and 5 respectively.

Table 3: The corpus statistics for BTEC task

Corpus	Size
Training set	19,972 sentence pairs
Development set	1,000 sentences with 16 references
Test set	1,508 sentences with 16 references

Table 4: The corpus statistics for CT-CE task

Corpus	Size
Training set	30,033 sentence pairs
Development set	3,000 sentences with 16 references
Test set	1,447 sentences with 16 references

Table 5: The corpus statistics for CT-EC task

Corpus	Size
Training set	30.033 sentence pairs
Development set	800 sentences with 7 references
Test set	665 sentences with 7 references

GIZA++[1] is used to get alignments from the training corpus with grow-diag-final-and option. We train a 5-gram language model with SRILM[2] on the target part of our training corpus.

[1] http://www.statmt.org/moses/giza/GIZA++.html
[2] http://www.speech.sri.com/projects/srilm/

The translation model is generated by Moses [1] (2010-8-13 version) with default parameter settings. The bestN option is set up to 100 in MERT.

4.2. Experiment results

Table 6 shows the total number of rules we build for each type.

Table 6: The rule counts for each type

Type	En-Ch	Ch-En
Arabic	29	20
Cardinal	41	80
Ordinal	7	7
Date	36	29
Clock	13	15
Week	1	2
Figure	5	18

In our experiments, we find that the most complicated cases are among cardinal numbers and date expressions. Take the English date expression for example. When we say "the third of September", there are different ways to convey the same date, such as "the 3rd of September", "September, 3","Sep. 3" and so on. Fortunately, they are somehow regular and thus easy to write rules for other forms by analogy.

Before we apply all the rules on the translation system, we calculate the statistics manually about the ratio of sentences containing numbers in Table 7.

Table 7: The ratio of sentences with numbers

Corpus	Development	Test
BTEC	9.3%	6.7%
CT-CE	13.5%	13.5%
CT-EC	15.6%	17.8%

From Table 7, we can see that the ratios are different in different tasks. In order to alleviate the interference caused by the distribution difference, we make two kinds of evaluation about the rule contributions. One of them is an evaluation on the original test set, the other is the evaluation on sentences with numbers, which we call the **with-number** sentence set. Next, we will carry out our experiments upon the two sets.

Table 8 shows the performance of using the RTN module to recognize number sequences in the development sets of the BTEC, the CT-CE, and the CT-EC corpus. In the table, the precision is the ratio of correctly captured numbers' counts to all captured ones. The recall is the ratio of correctly captured numbers' counts to the manually marked ones. In fact the performance largely relies on the rule-makers. The more numerical and time expressions they discover the better the performance will be.

Table 8: Performance of automatic recognition by RTN

Corpus	Precision	Recall	F-score
BTEC	0.98	0.90	0.94
CT-CE	0.96	0.93	0.94
CT-EC	0.91	0.84	0.88

4.2.1. Results on development and testing set

Table 9 shows how the translation quality measured by BLEU [9] on the original testing set changes score when we add the additional transferred translation table generated by the RTN module into the phrase-based translation table. The C-E evaluation is based on the case-insensitive BLEU-4 score, and the E-C evaluation is based on the BLEU-4 score of words.

Table 9: BLEU scores of development and testing set

		BTEC	CT-CE	CT-EC
Dev.	Baseline	41.25	33.64	34.54
	Ori + RTN	41.37	33.71	34.84
Test	Baseline	37.67	32.56	33.38
	Ori + RTN	37.79	32.99	33.71

Table 10: BLEU scores of with-number sentence set

		BTEC	CT-CE	CT-EC
With-number Dev.	Baseline	40.71	31.53	39.58
	Ori + RTN	42.05	31.93	41.35
With-number Test	Baseline	30.43	30.15	38.34
	Ori + RTN	31.77	31.81	39.28

In Table 9, Ori stands for Original phrase table generated by the training data, and Ori+RTN means the new translation table after adding the additional transferred translation table generated by the RTN module. Combined with Table 7, results in Table 9 shows an obvious trend in the testing set that the more with-number sentences in testing set, the more improvement of translation performance will be achieved, which can be further confirmed in Table 10 in the experiments on the with-number sentence set.

Comparing the BLEU scores in Table 9 and 10, we can get several hints. First, although the baseline performance seems rather good, our module is still able to improve the translation quality. With the help of precise and exact number translations, the results from machine translation system become more understandable and correct.

Second, the number seems a barrier in Chinese to English translation. The probable reason of that is that numbers in Chinese may be cut into several words instead of a complete word through word segmentation. As is known to us, different segmentation results may lead to different meanings for the computer. For example, the cardinal number "五十六" is a word which means "fifty-six" or "56". But after word segmentation, it is cut into "五 十 六", which becomes three words with three numbers. If the decoder searches translation candidates of this sequence in the phrase-table, there is more than one choice. Because the SMT only depends on the language model and the translation model, the decoder may give a wrong answer "five ten six" instead of "fifty-six".

In order to give a clearer view of the results, we list some typical with-number sentences with their translations in Table 11, where S is short for source sentence, T the translation of S through the baseline translation system without the RTN module, T* the translation of S through Ori+RTN.

The 1st example in Table 11 shows that the rule-based translations are better in word order for the translation of continuous figures. The baseline translation of the 2nd source sentence is wrong, and the currency unit "圆"(yen) is left unknown. Our RNT module helps to translate the number and unit correctly, which also reduce the OOV words in the translation. The remaining examples are from the results of the English to Chinese task. The 3rd sentence is an example to correct the wrong number. The translation in T cannot be understood. After correcting the numbers by the translation from the RTN module, the system is able to generate an

[1] http://www.statmt.org/moses/index.php?n=Main.HomePage

understandable translation. The last sentence is also an example of the reduction of the OOV number words.

The negative effects of OOV words include not only the unknown meaning of themselves, but also a result of confusing word order, as seen in the examples above. Sometimes the reduction of OOV words can contribute a lot to the word order in the final translation [10].

Table 11: Some with-number examples with their translations

1	S	你 的 房间 号码 是 **二 一 零** 。
	T	your room number is **two o one** .
	T*	your room number is **two one zero** .
2	S	就 买 这个 **六 百 三 十 圆** 的 吧 。
	T	i 'll buy this thirty **six hundred** , the **圆** .
	T*	i 'll buy this **six hundred and thirty yen** for that .
3	S	**six hundred eighty three** yen ?
	T	**三 六 八 百** 日元 吗 ?
	T*	**六 百 八 十 三** 日元 吗 ?
4	S	you 'll stay in a **hundred-dollar** room with a bath on the eleventh , and in a **ninety-dollar** room on the twelfth .
	T	你 会 住 在 **hundred-dollar** 带 浴室 的 房间 号 和 **ninety-dollar** 的 房间 十 二 号 。
	T*	你 会 住 在 **一 百 美元** 带 浴室 的 房间 在 十一号 和 **九 十 美元** 的 房间 在 十二号 。

4.2.2. Errors analysis

Translation rules are indeed helpful in our experiments, however, there are still some errors and problems currently remaining unsettled. In the following we list the most common errors.

- Numbers with multiple translations in the target side: in our experiments, the translations are sometimes correct for numbers but wrong for the unit following the number in the target language. For example, When we translate "the thirteenth", we would obtain two translation results through our rules, "在 十三 号" and "第 十三". They are assigned the same probability, and the final choice is determined only by the language model, which may lead to a wrong final choice if they merely occur in the language model. On the other hand, it is likely to omit one or two senses when we create rules.
- Number translations before and after the words "to", or "and" are sometimes inconsistent: The numerical expressions are often not complete, and the same sequence will be omitted, such as "August eleventh, twelfth and thirteenth". It is possible to recognize the days before "and", but the last number is hard to track.
- In our framework, we just have tried integrating the translation of numbers into the SMT system. Although the translations of numbers are corrected by our module, their positions are sometimes wrong. As a matter of fact, there is a complicated but better way to get rid of it. If we replace the numbers with their corresponding types at the training stage, as well as the source sentence at the decoding stage, then the completeness and independence of numbers are guaranteed, which is promising to improve the translation quality much more, which we will test it in the future work.

4.3. Extent experiment

We also did experiments on Inner Mongolian to Chinese (IM-C), Uyghur to Chinese (U-C) and Japanese to Chinese (J-C) for further verification, where CWMT'2011[1] corpora are used as multi-language experimental data. Because of the lack of reference of the testing set, we only observe the improvement on the development set. Table 12, 13, and 14 separately present the corpus statistics, including the with-number sentences which we counted manually.

Table 12: The corpus statistics for Inner Mongolian-Chinese

Corpus	Size
Training set	134,567 sentence pairs
Development set	1000 sentences with 4 references
With-number set	134 sentences

Table 13: The corpus statistics for Uyghur -Chinese

Corpus	Size
Training set	100,000 sentence pairs
Development set	700 sentences with 4 references
With-number set	169 sentences

Table 14: The corpus statistics for Japanese-Chinese

Corpus	Size
Training set	564,996 sentence pairs
Development set	500 sentences with 4 references
With-number set	217 sentences

Table 15 gives the experimental results of the development of IM-C, U-C and J-C.

Table 15: BLEU scores of development set and with-number set

		IM-C	U-C	J-C
Dev.	Baseline	24.58	53.42	42.24
	Ori + RTN	24.85	54.12	42.72
With-number	Baseline	24.26	47.40	42.51
	Ori + RTN	26.24	48.86	43.47

We can see that the translation performance improves much both on the development set and on the with-number set. In table 16, we give some samples of rules and the number translations of the baseline system and our system. In the table, the basic translation pairs they use are not presented in detail, due to the space limitation of this paper. But the source template, target template and operation groups are shown.

Table 16: Examples of with-number translation and rules

M-C
S: ᠳᠥᠷᠪᠡᠨ ᠶᠡᠬᠡ ᠭᠣᠣ ᠪᠠ ᠲᠠᠪᠢᠨ ᠨᠢᠭᠡᠨ
T: 四 一百五十 元
T*:四百五十元
Source Template :
(ᠨᠠ ᠬᠣᠭ |ᠬᠣᠭᠣᠷ ᠬᠡᠪ ᠬᠡᠪ ᠬᠡᠷᠡᠭ ᠬᠡᠪ ᠪᠠ ᠬᠡᠷᠡᠭ ᠬᠡᠪ ᠭᠣᠣ ᠬᠡᠪ ᠪᠠ ᠬᠡᠷᠡᠭ ᠬᠡᠪ ᠨᠠ ᠬᠡᠪ ᠲᠠᠪᠢᠨ ᠬᠡᠪ ᠵᠢᠷᠭᠤᠭ ᠬᠡᠪ)
(ᠵᠢᠷᠭᠤᠭ |ᠬᠤᠶᠠᠷ ᠬᠡᠪ ᠳᠥᠷᠪᠡᠨ ᠬᠡᠪ ᠬᠤᠶᠠᠷ ᠬᠡᠪ ᠲᠠᠪᠢᠨ ᠬᠡᠪ ᠲᠠᠪ |ᠪᠤᠳᠠᠯᠠᠭᠤᠨ ᠬᠡᠪ ᠬᠤᠭᠤᠷᠠᠨ ᠬᠡᠪ ᠲᠠᠪᠢᠨ ᠬᠡᠪ ᠵᠢᠷᠭᠤᠭ ᠬᠡᠪ)
(ᠨᠢᠭᠡᠨ ᠬᠡᠪ ᠳᠥᠷᠪᠡᠨ ᠬᠡᠪ ᠲᠠᠪᠢᠨ ᠬᠡᠪ ᠵᠢᠷᠭᠤᠭ ᠬᠡᠪ ᠵᠢᠷᠭᠤᠭ ᠬᠡᠪ ᠨᠢᠭᠡᠨ ᠬᠡᠪ) (ᠳᠥᠷᠪᠡᠨ ᠲᠠᠪᠢᠨ ᠵᠢᠷᠭᠤᠭ)?
Target Template : var_1 var_2 var_3 var_4
Operation Group：
@var_1+R+<Cardinal><#1>@var_1+R+<Cardinal><#3>
@var_2+R+<Cardinal><#3>@var_3+R+<Cardinal><#6>

[1]http://mt.xmu.edu.cn/cwmt2011/en/index.html

@var_4+R+<Date><#4>;;

U-C

S: 48 يا شلار
T: 48 ياشلار
T*:48 岁
Source Template :
([0-9]+)(ق لد ابد ىإد ا شلار|ذ ىإد لوم|ذ ىإ قيكد رلد تد ارماق نى|تد)
Target Template : var_1 var_2
Operation Group：@var_2+R+<Date><#5>;;

J-C

S: 4 月 3 日
T: 4 月 三 天
T*:4 月 3 日
Source Template :
([1-9]|10|11|12) 月 ((1|2|3){0,1}[0-9]) 日
Target Template : var_1 月 var_2 日
Operation Group：@var_1+T+NULL @var_2+T+NULL;;

5. Related work

In fact, the research on number translations started in early. [2] analyzed cardinal number names in five languages, and implemented the rules governing the construction of number names. The translation from a figure expression into a word expression is also done. [11] did part work of of [2], which only discussed about how to translate number names to Dutch. But they did not separate rules from source codes, which handicaps the scalability of rules. And the number representation they cared about is limited to cardinals with well formats, which is not enough for processing complicated and not so standardized text. [12] proposed an embedded-in-code-rule based framework to do the number recognition and translation task on Chinese-English translation. [13] combined a rule based machine translation system with a statistic based post editor, which showed helpful for a more accurate performance than only using statistical machine translation. However, although the former works on RTN have achieved successes in different ways, as far as we see, works on language-independent and code-separate rule-based number recognition and translation are very rare. Our work may be a big step in the work of number recognition and translation.

6. Conclusions

In this paper, we design a language-independent and rule-based method to translate numbers more precisely. The templates are totally separated from the source codes, which makes multiple language translation of numbers with one framework practical. We design a variety of operators to make the templates educible and dynamic compared to traditional templates, which allows rule-makers to get rid of source code to present the inducing progress and gives more flexible rooms to establish rules. Rule-makers can easily write templates and operations according to their bilingual knowledge of numbers. In our experiments we build groups of rules for bidirectional English-Chinese translation, as well as for some other languages to Chinese. After we apply the rules on the phrase-based translation, the translation quality improved upon both the development set and the testing set. The performance on with-number sentences set are all improved around 1 point BLEU score.

In the future work, we are about to explore more suitable integrating strategy for the RTN module into the SMT system. Rules for other language pairs will be built to improve their translation performance.

7. Acknowledgements

The research work has been partially funded by the Natural Science Foundation of China under Grant No. 60975053 and the Hi-Tech Research and Development Program (“863” Program) of China under Grant No. 2011AA01A207, supported by the External Cooperation Program of the Chinese Academy of Sciences. This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

8. References

[1]Koehn, P., Och, F.J. and Marcu, D., "Statistical phrase-based translation," in the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology Stroudsburg, PA, USA, 2003, pp. 48-54.

[2]Hugo Brandt, Corstius, "Automatic translation between number names," Grammars for number names, vol. 7, p. 103, 1968.

[3]Hayes-Roth, F., "Rule-based systems," Communications of the ACM, vol. 28, pp. 921-932, 1985.

[4]McDermott, J., "R1: A rule-based configurer of computer systems," Artificial intelligence, vol. 19, pp. 39-88, 1982.

[5]Ilgun, K., Kemmerer, R.A. and Porras, P.A., "State transition analysis: A rule-based intrusion detection approach," Software Engineering, IEEE Transactions on, vol. 21, pp. 181-199, 1995.

[6]Ishibuchi, H., Nozaki, K., Yamamoto, N. and Tanaka, H., "Selecting fuzzy if-then rules for classification problems using genetic algorithms," Fuzzy Systems, IEEE Transactions on, vol. 3, pp. 260-270, 1995.

[7]Hammer, J., Garcia-Molina, H., Nestorov, S., Yerneni, R., Breunig, M. and Vassalos, V., "Template-based wrappers in the TSIMMIS system," ACM Sigmod Record, vol. 26, pp. 532-535, 1997.

[8]Yen, J. and Wang, L., "Simplifying fuzzy rule-based models using orthogonal transformation methods," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 29, pp. 13-24, 1999.

[9]Papineni, K., Roukos, S., Ward, T. and Zhu, W.J., "BLEU: a method for automatic evaluation of machine translation," in the 40th Annual Meeting on Association for Computational Linguistics, 2002, pp. 311-318.

[10]Zhang, Jiajun, Zhai, Feifei and Zong, Chengqing, "Handling unknown words in statistical machine translation from a new perspective " presented at the NLPCC, Beijing, 2012.

[11]Corstius, H.B., "Automatic translation of numbers into Dutch," Foundations of Language, vol. 1, pp. 59-62, 1965.

[12]Zhai, Feifei, Xia, Rui and Zong, Chengqing, “An Approach to Recognizing and Translating Chinese & English Time and Number Named Entities,” in the 7th China Workshop on Machine Translation (CWMT’2009), Nanjing, 2009.

[13]Terumasa, E., "Rule based machine translation combined with statistical post editor for japanese to english patent translation," 2007, pp. 13-18.

Evaluation of Interactive User Corrections for Lecture Transcription

Henrich Kolkhorst, Kevin Kilgour, Sebastian Stüker, and Alex Waibel

International Center for Advanced Communication Technologies – InterACT
Institute for Anthropomatics
Karlsruhe Institute of Technology, Germany
henrich.kolkhorst@student.kit.edu,
{kevin.kilgour, sebastian.stueker, alexander.waibel}@kit.edu

Abstract

In this work, we present and evaluate the usage of an interactive web interface for browsing and correcting lecture transcripts. An experiment performed with potential users without transcription experience provides us with a set of example corrections.

On German lecture data, user corrections greatly improve the comprehensibility of the transcripts, yet only reduce the WER to 22%. The precision of user edits is relatively low at 77% and errors in inflection, case and compounds were rarely corrected. Nevertheless, characteristic lecture data errors, such as highly specific terms, were typically corrected, providing valuable additional information.
Index Terms: speech recognition, user study, transcript correction, lectures

1. Introduction

Recording and archiving of lectures is feasible and practiced at several universities (e.g., the MIT OpenCourseWare project [1]). Nevertheless, automatic speech recognition (ASR) on lecture data is non-trivial, for example due to highly specific contents and spontaneity in speech style. Since word error rates (WERs) should be less than 25% for a lecture archive to be perceived useful [2], careful adaptation is needed.

Besides enabling searchable archives of lectures, ASR is necessary for spoken language translation. At KIT, significant research is conducted to enable simultaneous translation of lectures [3], requiring good speech-to-text performance for further processing. Misrecognition of content words, such as substituting the word "censor" for "tensor" in a mathematical context, impairs the readability of transcripts and can cause substantial errors in subsequent computation steps.

Most of these errors can easily be corrected by humans. However, professional transcription on a larger scale is unrealistic due to required time and resulting costs. Especially large-vocabulary recognizers often contain the correct words in their lattice (e.g., a 1-best WER of 55% on lecture data compared to a lattice WER of 30% [4]) and, given adequate tools, users of lecture archives can quickly correct such errors. Ideally, corrections of existing transcripts should also be used to improve future recognition results on similar data.

In this work, we investigate the quality of error corrections by users of a lecture archive and the usability of such corrections for system adaptation. We present an interface for browsing transcriptions in which error corrections can be made quickly, along with the results of a user experiment involving the correction of German lectures.

After giving an overview of related work on user corrections and their utilization for adaptation in Section 2, we describe the interface and experimental setup in Section 3 and 4. Results of a user experiment are presented and analyzed in Section 5.

2. Related Work

In a setting of webcast archives, Munteanu et al. describe a "wiki-like" transcript edit tools for lectures, which can be used to correct errors in speech recognition output [5]. In a user study, students corrected lectures (from a single course), reducing the WER of the ASR output by 53%. However, the initial WER of 50% to 60% was quite high. If the actual transcription of a sixth of a lecture is available, transformation-based learning from the correct transcript has been shown to reduce WER by 12.9% [6].

Within the framework of the MIT OpenCourseWare project, Hsu and Glass investigate the possible improvements based on partial user transcriptions by adapting language model (LM) interpolation parameters. They show that with 300 words of transcription, adaptation on recognizer hypotheses is outperformed and about 1% absolute reduction of WER can be achieved from a 33.2% baseline. However, they use parts of the reference transcription and not actual user data.

Yu worked on correction of MIT lecture transcripts based on re-recognition of error-prone regions [7]. Using oracle corrections from reference transcript, relative WER reductions of 39% were obtained. In a user test, precision and recall of user corrections were both at 97%, but no re-recognition was performed with actual user data. The correctors in the test were primarily speech researchers and therefore probably aware of transcription guidelines.

Ogata and Goto used confusion network output for error correction during online speech recognition and showed that in theory, 83% to 99% of errors in podcast transcripts could be corrected based on confusion networks [8]. Additionally, Ogata et al. investigated user corrections in their "PodCastle" podcast transcription service [9, 10]. User corrections reduced WER by more than 50% and 46 hours of corrected podcast data were collected. However, the actual correction data was not analyzed in detail.

Based on user correction, Ogata et al. used Maximum Likelihood Linear Regression (MLLR) and subsequent Maximum A Posteriori Estimation (MAP) to adapt the acoustical model. The model adaptations yielded relative improvements in WER up to 23% when a large number of episodes had been corrected (between 7 and 20 hours of training data) [9].

Recent work has investigated the use of platforms for human intelligence tasks, such as Amazon's Mechanical Turk (MTurk), for transcription and correction of ASR transcripts. Marge et al. used MTurk workers to transcribe clean instructional audio segments and found the quality of transcripts to be at 5% WER. Considering cost and accuracy, they suggest using three to five workers for transcrip-

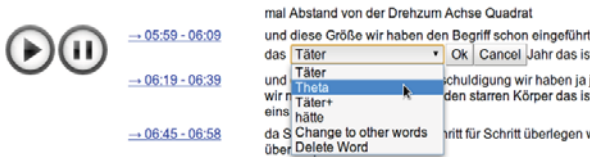


Figure 1: Screenshot of alternate confusion network hypotheses

tion [11]. Lee and Glass use a two-stage process to generate lecture transcripts from MTurk tasks. As a first step, short utterances are transcribed, yielding a WER of 16%. In a second stage, workers are asked to correct a baseline transcription from the first stage. Integrating a detection for poor quality transcripts and giving workers performance estimations as feedback, they report a WER of 10.2% after the second stage [12].

3. Interface Features

To facilitate the interaction with lecture archives, interface usability and familiarity is essential, making web applications a natural choice. The interface has been implemented solely based on HTML5 standards without the need for browser plugins to assure platform independence. The Google Web Toolkit[1] was used for implementation.

ASR lattices are converted to *confusion networks* [13], a representation with total ordering of word hypotheses which are collapsed into "clusters" at specific time slots. This enables the display of time-aligned alternate hypotheses in the interface. Playback of the lecture's audio recording can be started from the beginning or users can jump to specific utterances. By default, the 1-best transcript is displayed and the current utterance is subtly highlighted during audio playback.

By clicking on a word, a list of alternate hypotheses for the time slot along with the option to delete or enter a different word is displayed (Figure 1). To prevent a cluttered or complex interface, users cannot move words between utterances or insert words at specific time slots. Instead, existing slots can be modified to consist of multiple words. Changes are saved instantaneously to enable frictionless interaction. It is possible to redirect hypotheses of online recognition into the web interface.

4. Experimental Setup

A user study was performed to evaluate the correction performance of students using the web interface. Since corrections will typically take place "offline" (not during the lecture), the initial ASR hypotheses have been generated by a system combination to achieve a high-quality basis for subsequent editing.

4.1. Corpus Characteristics

For the experiment, German lectures from a variety of topics were used. The lectures form a subset of the KIT Lecture Corpus for Speech Translation [14]. The lectures "Algorithms for Planar Graphs" (ALGO), "Formal Systems" (FORM), "Cognitive Systems" (COGSYS), "Machine Translation" (MT) and "Multiprocessors" (PROC) cover different areas of computer science. The "Technical Mechanics" (MECH) lecture is from an unrelated, but still technical area, whereas the lectures about "Population Geography" (GEO), "World War 2" (WW2) and "Copyright Law" (LAW) cover non-technical topics.

[1] https://developers.google.com/web-toolkit/

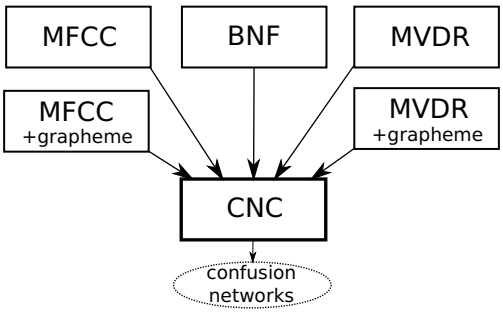


Figure 2: Decoding systems for generation of baseline transcription

The lectures were recorded with a close-talking microphone and the audio data has a sampling rate of 16kHz. The speaker style varies significantly. Some lectures contain many hesitations (COGSYS) whereas others are characterized by false starts (MT) or significant amounts of read formulas (MECH, ALGO).

The lectures have been divided into sections edited by the users (EDIT set) and unedited sections used for evaluating adaptation performance (EVAL set). Only lectures in which the unedited sections contained more than 1000 words have been included in the EVAL set to make it reasonably different from the EDIT set.

4.2. Baseline ASR System

The baseline hypothesis which are displayed in the web interface and are editable by the user are produced with the Janus Speech Recognition Toolkit's Ibis Decoder [15] through a confusion network combination (CNC) [13] of five speaker independent systems developed for the 2011 Quaero Evaluation as depicted in Figure 2. It is an improved version of the 2010 evaluation system [16] and similar to the Spanish system described by Kilgour et al. [17]. The underlying systems use three different frontends, mel-frequency cepstral coefficients (MFCC), warped minimum variance distortionless response (MVDR), and MVDR based bottleneck features (MVDR-BNF). Additionally, two systems use graphemes instead of phonemes. The system combination has been chosen to provide state-of-the-art transcripts as basis for corrections. The language model is built from the transcripts of the quaero training data, scraped newspaper data and webdumps.

The vocabulary is case-sensitive and fairly large containing roughly 300k sub-words and 480k pronunciation variants. The sub-words are used in order to improve the recognition of compound words. Sub-words of the 1-best hypothesis are merged appropriately for display in the user interface. A substantial amount of sub-words are, however, also full words. The sub word LM doesn't correctly merge all compound words, so many of them are still falsely recognized as multiple independent words.

The word error rates of this setup on the different lectures are listed in the second column of Table 1. Generally, many errors can be attributed to a mismatch between the training data, which consists primarily of broadcast news, and the lecture data. Especially the frequent use of rare and non-German terms causes problems. For example, the term "phrase alignment" is central to the MT lecture, but the lack of the English pronunciation of "phrase" leads to continuous misrecognition and makes many utterances difficult to comprehend.

Lecture	WER ASR	WER User	Rel. WER Improvement (content words only)	fraction of words edited	edit precision	#words	# users
MECH	32.65	21.15	35.2% (30.8%)	17.3%	80.1%	1493	3
GEO	22.85	19.18	16.1% (15.7%)	13.0%	82.9%	1393	3
WW2	28.57	24.96	12.6% (19.0%)	13.1%	71.4%	1019	2
ALGO	35.92	24.76	31.1% (38.9%)	20.3%	68.6%	1836	4
FORM	29.14	20.68	29.0% (34.6%)	17.4%	85.2%	3137	5
COGSYS	33.61	17.39	48.3% (46.6%)	21.1%	89.7%	876	2
MT	38.65	22.98	40.4% (48.7%)	24.5%	83.5%	4704	6
PROC	35.89	26.19	27.0% (28.1%)	19.9%	72.0%	1365	4
LAW	28.73	20.53	28.5% (24.2%)	19.7%	72.4%	2461	4
total	32.71	22.08	32.5% (34.7%)	19.6%	77.4%	18284	11
mean	31.78	21.98	29.8% (31.9%)	18.5%	77.2%	2032	3.7
std. dev.	4.89	2.93	11.1 (11.5)	3.7	7.1	1229	1.3

Table 1: Overview of corrections by lecture (EDIT set). User values are aggregated over all users who edited the particular lecture. Word error rate and precision are case-insensitive. The relative improvement of the WER by user edits is given on all words and on "content words" only (excluding the 1000 most frequent words in training).

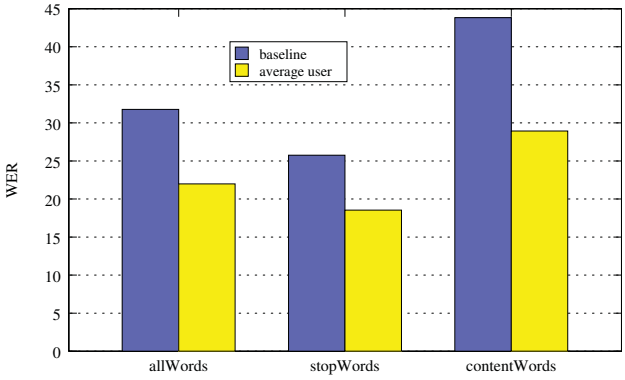


Figure 3: Differences between word error rates of "stop words" (the 1000 most frequent words in the training data) and content words

4.3. Setup of User Study

The experiment was carried out with 11 test subjects, most of them university students with a technical major and some familiarity with the subject matter of the lectures. However, none of them had experience in transcription nor did anybody use the interface before. Each user was asked to correct three lecture segments of five minutes each.

Test subjects were asked to correct the ASR transcripts based on their own judgment, i.e. correct the transcripts primarily to improve readability and correct only the errors that they feel to be problematic. The omission of fine-grained transcription guideline aims to simulate an every-day usage of a learning environment where users will only correct certain subsets of errors and not adhere to detailed rules for editing.

In order to be able to analyze the influence of different correctors and observe the familiarization with and usage of the interface, the experiment was carried out in a controlled lab environment. Lecture segments used in the experiment were edited by at least two subjects and each subject edited segments with, on average, 1669 words of which 393 words were corrected (see Table 2). To enable comparison of different users, all were presented with the unedited recognizer hypothesis.

5. Results of User Corrections

User corrections improved the transcript quality substantially, yet not comparable with professional transcription. The initial (case-insensitive) word error rates of the baseline transcript ranged from 22.9 to 38.7, with technical lectures generally having more errors. In total, users improved the word error rate by about a quarter from 32.71 to 22.08 (cf. Table 1).

On average, users edited every fifth word slot in the baseline hypotheses. However, in almost a quarter of these edits, incorrect edit operations are made. Furthermore, if alternate hypotheses from the confusion network are chosen, the precision drops below 50% (Table 2). This is primarily due to the selection of compound fragments instead of entering the complete word.

Whereas the quality of the baseline transcripts varies substantially, the user corrections reduced the variance of errors, attenuating the difference between technical and non-technical lectures. If errors are analyzed on "stop words" (the 1000 most frequent words in the training data) and "content words" (the rest) separately, the correction performance varies substantially between lectures, with on average slightly greater improvement on content words (see Figure 3).

5.1. Characteristics of User Edits

Based on manual inspection, the user edits substantially increased the readability and comprehensibility of the lecture transcripts due to the correction of words central to the lecture excerpts. However, there are some peculiarities in the user edits which contribute to their relatively low precision.

Spelling errors are relatively common in the user corrections, especially in rare terms. For example, not all users were familiar with the word "tensor" and users frequently used misspelled words like "aligment" in their corrections which is acceptable for human users, but obviously hurts the WER. Similar problems occur, if lecturers use German inflection on English words such as "pointe" as the plural form of "point".

Another frequent issue in user corrections was the treatment of German compounds. Despite the existence of sub-words in the ASR vocabulary, many compounds are still misrecognized in the ASR hypothesis, for example when only one part is correctly detected. Often, users corrected only the first part (replaced it with the actual compound), but did not remove the second part. Since the deletion function was generally used in other cases, it can be assumed that these duplications did not substantially bother the users.

	mean	std. dev.
number of words edited	393.4	96.1
word error rate (case-independent)	21.95	2.80
insertions (%)	8.1	1.3
deletions (%)	34.9	5.3
substitutions (%)	47.0	6.2
word error rate (case-dependent)	24.33	2.96
ratio of edits chosen from confusion network alternatives in relation to total edits (%)	64.6	8.9
precision of edits chosen from confusion networks (case-independent) (%)	46.6	3.0

Table 2: Detailed statistics of user edits, aggregated over all users.
Users select confusion network alternatives in a majority of cases which are mostly incorrect and lead to a high substitution error.

Errors in case were rarely corrected by users. In their own corrections, some users used correct capitalization rules whereas others preferred lower-case corrections. Generally, alternate confusion network hypotheses were chosen regardless of correct capitalization. Ignoring the case, if multiple users corrected the same hypotheses, the inter-user agreement of 89% is higher than their overall precision.

A manual inspection of user corrections shows that users frequently did not correct or insert missing adverbs or adjectives whereas the general sentence structure was usually corrected.

5.2. Analysis

Overall, the relative improvement of transcript quality is much less than described by Munteanu et al. [5]. However, due to the lower initial error rates, the resulting WER is similar, supporting the observation, that a WER of 25% is somewhat acceptable for user and better, more fine-grained, corrections are perceived as too cumbersome.

Despite different degrees of familiarity with the lecture topics, all users performed quite similar. However, the precision of user edits is relatively low, much less than the 97% described by Yu [7].

Some loss in precision could be attributed to compounds and inflection in the German language and a user preference of making the lecture transcripts readable rather than completely correct appears to be a reasonable explanation. This agrees with the observation, that case and (compound) spelling was rarely corrected.

Especially the precision of less than 50% if an alternate hypotheses from the confusion network is chosen suggests that users will accept suboptimal corrections if they can be selected quickly. Nevertheless, characteristic lecture data errors were corrected manually if essential for the meaning. Phrases central to a lecture were continuously corrected even if they were mostly misrecognized.

6. Utilization for system adaptation

It is desirable that user correction do not only improve existing transcript, but rather improve future recognition performance. In this work, we investigated the use of corrected transcripts for system adaptation, compared with unsupervised adaptation on the CNC hypotheses.

Based on user corrections, a "consensus" transcript was created by using the most frequent user correction for each confusion network slot or the recognizer hypothesis if the slot has not been edited. Out-of-vocabulary words (OOVs) inserted by the users were split into existing sub-words in the vocabulary if possible. The rest was added to the vocabulary (without manual selection) with generated pronunciations.

Following the objective of improving simultaneous recognition of lectures, the "offline" correction should be used to adapt a single "online" system. Hence, the adaptation and evaluation is performed with a single MVDR system as opposed to the system combination of the first pass. Adaptation consists of vocal tract length normalization [18] and MLLR [19].

Evaluating the adapted system on unedited segments of the lectures shows that the low precision of user edits is problematic when using them as a basis for adapting models. When consensus transcripts are generated based on the user edits and used instead of the CNC output, small improvements on uncorrected data can be seen on content words, yet overall improvements in WER are not significant.

This lack of improvement compared to adaptation on the CNC output can be attributed to the relative sparsity of edited words and the heterogeneity of user edits, especially concerning compound treatment and typographical errors.

7. Conclusion

In this work, we presented a web interface for interactive correction of lecture transcripts and performed a user experiment to obtain information about quality and characteristics of user corrections without transcription guidelines.

User corrections improved the comprehensibility and quality of transcripts from a human perspective, i.e. for presentational purposes. This reduced the word error rate by a third to a level of 22%, which is, however, substantially worse than transcription quality. Especially the precision of user edits is relatively low at 77%, primarily due to errors in inflection, case and compound structure. This diminishes the usefulness of user-corrected segments for adaptation.

Future work will focus on utilization of corrections and refined adaptation methods. Additionally, it would be interesting to analyze user corrections on a larger scale in an actual setting and investigate the impact on subsequent machine translation.

8. Acknowledgements

'Research Group 3-01' received financial support by the 'Concept for the Future' of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative.

The work leading to these results has received funding from the European Union under grant agreement no 287658.

9. References

[1] J. Glass, T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," in *INTERSPEECH-2007*, 2007, pp. 2553–2556.

[2] C. Munteanu, R. Baecker, G. Penn, E. Toms, and D. James, "The effect of speech recognition accuracy rates on the usefulness and usability of webcast archives," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ser. CHI '06. New York, NY, USA: ACM, 2006, pp. 493–502.

[3] C. Fügen, A. Waibel, and M. Kolss, "Simultaneous translation of lectures and speeches," *Machine Translation*, vol. 21, pp. 209–252, 2007.

[4] C. Chelba, J. Silva, and A. Acero, "Soft indexing of speech content for search in spoken documents," *Computer Speech & Language*, vol. 21, no. 3, pp. 458–478, 2007.

[5] C. Munteanu, R. Baecker, and G. Penn, "Collaborative editing for improved usefulness and usability of transcript-enhanced webcasts," in *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 373–382.

[6] C. Munteanu, G. Penn, and X. Zhu, "Improving automatic speech recognition for lectures through transformation-based rules learned from minimal data," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*. Association for Computational Linguistics, 2009, pp. 764–772.

[7] G. T. Yu, "Efficient error correction for speech systems using constrained re-recognition," Master's thesis, Massachusetts Institute of Technology, 2008.

[8] J. Ogata and M. Goto, "Speech repair: Quick error correction just by using selection operation for speech input interfaces," in *INTERSPEECH-2005*, 2005, pp. 133–136.

[9] J. Ogata, M. Goto, and K. Eto, "Automatic transcription for a web 2.0 service to search podcasts," in *INTERSPEECH-2007*, 2007, pp. 2617–2620.

[10] J. Ogata and M. Goto, "Podcastle: Collaborative training of acoustic models on the basis of wisdom of crowds for podcast transcription," in *INTERSPEECH-2009*, 2009, pp. 1491–1494.

[11] M. Marge, S. Banerjee, and A. Rudnicky, "Using the amazon mechanical turk for transcription of spoken language," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, march 2010, pp. 5270 – 5273.

[12] C. Lee and J. R. Glass, "A transcription task for crowdsourcing with automatic quality control," in *INTERSPEECH*, 2011, pp. 3041–3044.

[13] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[14] S. Stüker, F. Kraft, C. Mohr, T. Herrmann, E. Cho, and A. Waibel, "The KIT lecture corpus for speech translation," in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, 2012, to appear.

[15] H. Soltau, F. Metze, C. Fügen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *Automatic Speech Recognition and Understanding, 2001. ASRU '01. IEEE Workshop on*, 2001, pp. 214–217.

[16] S. Stüker, K. Kilgour, and F. Kraft, "Quaero 2010 speech-to-text evaluation systems," in *High Performance Computing in Science and Engineering '11*, W. E. Nagel, D. B. Kröner, and M. M. Resch, Eds. Springer Berlin Heidelberg, 2012, pp. 607–618.

[17] K. Kilgour, C. Saam, C. Mohr, S. Stüker, and A. Waibel, "The 2011 KIT Quaero speech-to-text system for spanish," in *IWSLT-2011*, 2011, pp. 199–205.

[18] P. Zhan and M. Westphal, "Speaker normalization based on frequency warping," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2, apr 1997, pp. 1039 –1042 vol.2.

[19] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171 – 185, 1995.

Factored Recurrent Neural Network Language Model in TED Lecture Transcription

Youzheng Wu, Hitoshi Yamamoto, Xugang Lu, Shigeki Matsuda, Chiori Hori, Hideki Kashioka

Spoken Language Communication Laboratory,
National Institute of Information and Communications Technology,
Kyoto, Japan

youzheng.wu@nict.go.jp

Abstract

In this study, we extend recurrent neural network-based language models (RNNLMs) by explicitly integrating morphological and syntactic factors (or features). Our proposed RNNLM is called a factored RNNLM that is expected to enhance RNNLMs. A number of experiments are carried out on top of state-of-the-art LVCSR system that show the factored RNNLM improves the performance measured by perplexity and word error rate. In the IWSLT TED test data sets, absolute word error rate reductions over RNNLM and n-gram LM are 0.4∼0.8 points.

1. Introduction

Language models (LM) are a critical component of many application systems such as automatic speech recognition (ASR), machine translation (MT) and optical character recognition (OCR). In the past, statistical back-off n-gram language models with sophisticated smoothing techniques have gained great popularity because of their simplicity and good performance. Recently, neural network based language models (NNLMs), such as the feed-forward NNLM [3, 19], the recurrent NNLM (RNNLM) [15, 16] and the deep NNLM [2], have been continuously reported to perform well amongst other language modeling techniques. Among them, RNNLMs are state-of-the art [2, 14], which embed words in a continuous space in which probability estimation is performed using artificial neural networks consisting of input layer, hidden layer, and output layer. Due to consistent improvement in terms of perplexity and word error rate and their inherently strong generalization, they have become an increasingly popular choice for LVCSR and statistical MT tasks.

Many of these RNNLMs only use one single feature stream, i.e., surface words, which are limited to generalize over words without using linguistic information, including morphological, syntactic, or semantic. In this paper, we extend word-based RNNLMs by explicitly integrating morphological and syntactic factors (or features), called a factored RNNLM (fRNNLM), and show its performance in a LVCSR system. The experimental results of our state-of-the-art recognizer on transcribing TED lectures[1] demonstrate that it significantly enhances performance measured in perplexity and word error rate (WER).

This paper is organized as follows: In Section 2, we describe our proposed factored RNNLM in detail. Section 3 shows the performance of our model as measured by both perplexity and WER. We introduce related studies in Section 4. We finally summarize our findings and outline future plans in Section 5.

2. Proposed method

The purpose of this paper is to integrate additional linguistic information into a RNNLM, called a factored RNNLM, which can improve the generalization of RNNLM using multiple factors of words (stems, lemmas, parts-of-speech, etc.) instead of surface forms of words as input to recurrent neural networks. First of all, let us use an example to illustrate the shortcomings of surface word RNNLM. In extreme cases, the training data might only contain the following sentence: "difference between developed countries and developing countries". During training in the RNNLM that treats each word as a token in itself, the bi-gram "developing countries" is a completely unseen instance. However, for our factored RNNLM that incorporates stem features, "developing countries" belongs to seen instances in a sense because it shares the same stem bi-gram "develop countri" with the previous bi-gram "developed countries." This coincides with our intuition; "developed" and "developing" should add knowledge to each other during training. Our factored RNNLM may be more effective for such morphologically rich languages as Czech, Arabic, or Russian. This paper however, only evaluates it on English.

2.1. fRNNLM

The architecture of our factored RNNLM is illustrated in Fig. 1. It consists of input layer x, hidden layer s (state layer), and output layer y. The connection weights among layers are denoted by matrixes U and W. Unlike RNNLM, which pre-

[1] http://www.ted.com/

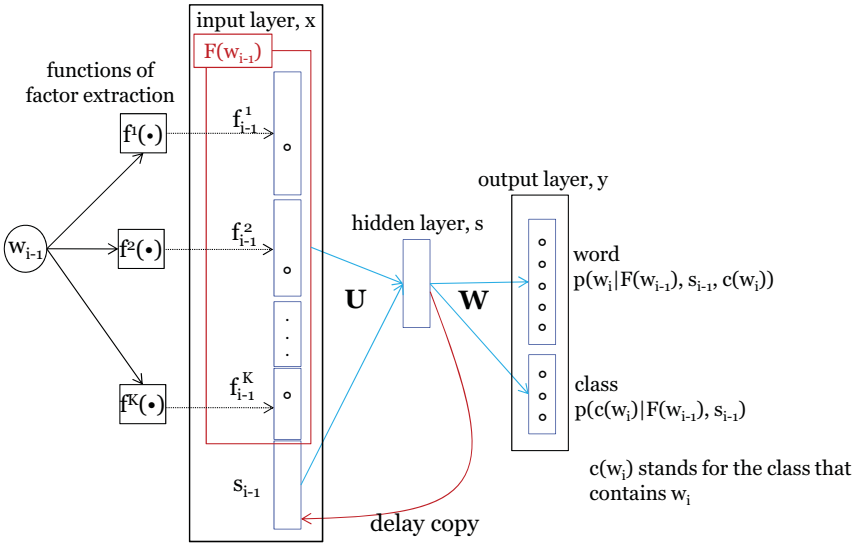


Figure 1: Architecture of factored recurrent NNLM.

Table 1: An example of factor sequences.

Word:	difference	between	developed	countries	and	developing	countries
Lemma:	difference	between	developed	country	and	developing	country
Stem:	differ	between	develop	countri	and	develop	countri
Part-of-speech:	NN	IN	JJ	NNS	CC	VBG	NNS

dicts probability $P(w_i|w_{i-1}, s_{i-1})$, our factored RNNLM predicts probability $P(w_i|F(w_{i-1}), s_{i-1})$ of generating following word w_i and is explicitly conditioned on a collection or bundle of K factors of one preceding word. It is implicitly conditioned on the factors of the entire history by the delay copy of hidden layer s_{i-1}. Here, $F(w_{i-1})$ is the vector concatenated from K factor vectors f^k_{i-1} $(k = 1, ..., K)$, f^k_{i-1} stands for the k-th factor vector encoded from the k-th factor of preceding word w_{i-1}, and the functions of factor extraction $f^k(\cdot)$ are used to extract the corresponding factors. A word's factors can be anything, including the word itself, its morphological class, its root, and any other linguistic features. An example is shown in Table 1[2].

In the input layer, the extracted factors are encoded into the factor vectors using the 1-of-n coding. Assume, for example, that the factor extracted by function $f^k(w_{i-1})$ is the m-th element in the k-th factor vocabulary, which is then encoded to $|f^k|$-dimension vector f^k_{i-1} by setting the m-th element of the vector to 1 and all the other elements to 0. Here, $|f^k|$ stands for the size of the k-th factor vocabulary. The K factor vectors are concatenated into $F(w_{i-1})$ as expressed in Eq. (1). Finally, the input layer is formed by concatenating factor vectors $F(w_{i-1})$ of the preceding word w_{i-1} and hidden layer s_{i-1} at the preceding time step, as shown in Eq. (2).

$$F(w_{i-1}) = [f^1_{i-1},\ f^2_{i-1},\ ...,\ f^K_{i-1}] \tag{1}$$

[2] http://www.cis.upenn.edu/~treebank/

$$x_i = [F(w_{i-1}),\ s_{i-1}] \tag{2}$$

Using the concatenation operation, our factored RNNLM can simultaneously integrate all factors and the entire history in stead of backing-off to fewer factors and a shorter context. The weight of each factor is represented in connection weight matrix U. Therefore, it can address the optimization problem well in factored n-gram LM [4, 7]. In the special case that f^1_{i-1} is a surface word factor vector and f^k_{i-1} $(k = 2, ..., K)$ are dropped, our proposed factored RNNLM goes back to the RNNLM.

The hidden layer employs a sigmoid activation function:

$$\begin{aligned} s^m_i &= f(\sum_j (x^j_i \times u_{mj})) \quad \forall m \in [1, H] \\ f(z) &= \frac{1}{1 + e^{-z}} \end{aligned} \tag{3}$$

where H is the number of hidden neurons in the hidden layer and u_{mj} is an element in matrix U denoting the corresponding connection weight.

Like [10, 16], we assume that each word belongs to exactly one class and divide the output layer into two parts: the first estimates the posterior probability distribution over all classes,

$$y^l_c = g(\sum_j (s^j_i \times w_{lj})) \quad \forall l \in [1, C] \tag{4}$$

where C is the number of predefined classes. The second computes the posterior probability distribution over the

words that belong to class $c(w_i)$, the one that contains predicted word w_i:

$$y_w^o = g(\sum_j (s_i^j \times w_{oj})) \quad \forall o \in [1, nc(w_i)] \tag{5}$$

where $nc(w_i)$ is the number of words belonging to class $c(w_i)$ and w_{lj} and w_{oj} are the corresponding connection weights.

To ensure that all outputs are between 0 and 1, and their sum equals to 1, the output layer employs a softmax activation function shown below:

$$g(z_d) = \frac{e^{z_d}}{\sum_x e^{z_x}} \tag{6}$$

Finally, probability $P(w_i|F(w_{i-1}), s_{i-1})$ is the product of two posterior probability distributions:

$$\begin{aligned} P(w_i|F(w_{i-1}), s_{i-1}) &= P(c(w_i)|F(w_{i-1}), s_{i-1}) \ \times \\ &\quad P(w_i|F(w_{i-1}), s_{i-1}, c(w_i)) \\ &= y_c^l|_{l=classid(c(w_i))} \times y_w^o|_{o=wordid(w_i)} \end{aligned} \tag{7}$$

The architecture of splitting the output layer into two parts can greatly speedup the training and the test processes of RNNLM without sacrificing much performance. Many word clustering techniques can be employed. In this paper, we map words into classes with frequency binning [16], which proportionally assigns words to classes based on their frequencies.

2.2. Training

To use the factored RNNLM, connection weight matrixes U and W must be learned. To learn them, training is performed with the back-propagation through time (BPTT) algorithm [5] by minimizing an error function defined in Eq. (8).

$$L = \frac{1}{2} \times \sum_{i=1}^{N} (t_i - p_i)^2 + \gamma \times (\sum_{lk} u_{lk}^2 + \sum_{tl} w_{tl}^2) \tag{8}$$

where N is the number of training instances, t_i denotes the desired output; i.e., the probability should be 1.0 for the predicted word in the training sentence and 0.0 for all others. The first part of this equation is the summed squared error between the output and the desired probability distributions, and the second part is a regularization term that prevents RNNLM from over-fitting the training data. γ is the regularization term's weight, which is determined experimentally using a validation set.

The training algorithm randomly initializes the matrixes and updates them with Eq. (9) over all the training instances in several iterations. In Eq. (9), ψ stands for one of the connection weights in the neural network and η is the learning rate. After each iteration, it uses validation data for stopping and controlling the learning rate. Usually, the factored RNNLM needs 10 to 20 iterations.

$$\psi^{new} = \psi^{previous} - \eta \times \frac{\partial L}{\partial \psi} \tag{9}$$

3. Experiments

To evaluate our factored RNNLM in the context of large vocabulary speech recognition, we use the data sets for the IWSLT large vocabulary continuous speech recognition shared task [9] to recognize TED talks published on the TED website. TED talks touch on the environment, photography and psychology without adhering to a single genre. This task reflects the recent increase of interest in automatically transcribing lectures to make them either searchable or accessible.

The IWSLT evaluation campaign defines a closed set of publicly available English texts as training data for LM, including a small scale of in-domain corpus (TED transcriptions) and a large scale of general-domain corpora (English Gigaword Fifth Edition and News Commentary v7). All training data are preprocessed by a non-standard-word-expansion tool that converts non-standard words (such as CO2 or 95%) to their pronunciations (CO two, ninety five percent). The most frequent $32.6K$ words are extracted from the preprocessed in-domain corpora, which, with the CMU.v0.7a pronunciation dictionary[3], are used as the LM vocabulary. Our vocabulary contains $156.3K$ entries with an OOV rate of 0.8% on the dev2010 data set. Additionally, the IWSLT data sets of tests 2010, 2011 and 2012 are used. Their statistics are shown in Table 2.

Table 2: Summary of the IWSLT test data sets

LM training data		
corpora	#sentences	#words
in-domain	$142K$	$2{,}402K$
general-domain	$123.4M$	$2{,}726.6M$
Test sets		
data sets	#talks	#utterances	#words
dev2010	8	934	$17.5K$
test2010	11	1664	$27.0K$
test2011	8	818	$12.4K$
test2012	11	1124	$21.9K$

For the in-domain and general-domain corpora, modified Kneser-Ney smoothed 3- and 4-gram LMs are constructed using SRILM [21], and interpolated to form a baseline of 3- and 4-gram LMs by optimizing the perplexity of the dev2010 data set.

Acoustic models are trained on 170h speech segmented from 788 TED talks that were published prior to 2011. We

[3] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

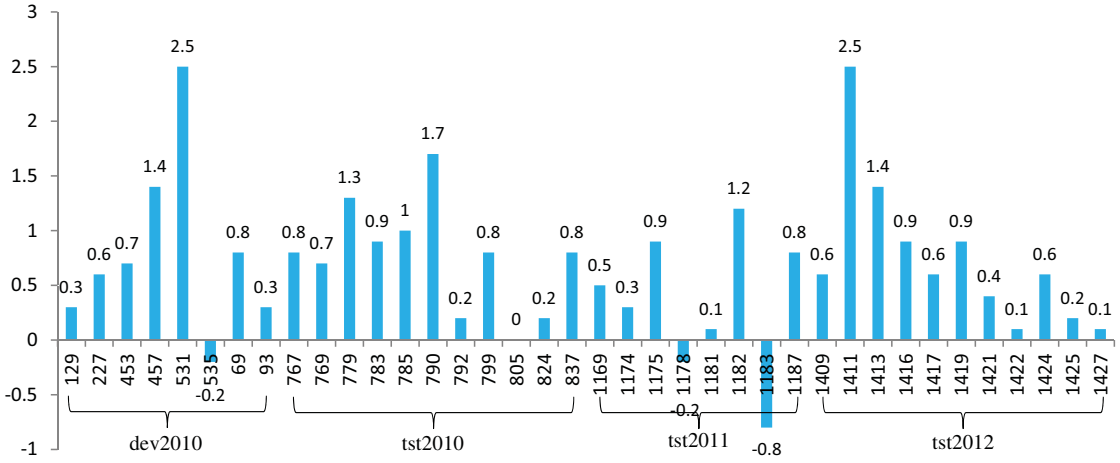


Figure 2: Absolute WER improvement on each talk.

employ two types of schemes, a Hidden Markov Model (HMM) and a Subspace Gaussian Mixture Model (SGMM) for each context-dependent phone and train them with the Kaldi toolkit [18]. HMM consists of 6.7K states and 240K Gaussians that are discriminatively trained using the boosted Maximum Mutual Information criterion. SGMM consists of 9.2K states. In addition, we apply speaker adaptive training with feature space maximum likelihood linear regression on top of the HMM and SGMM. The acoustic feature vectors have 40 dimensions. For each frame, we extract 13 static MFCCs, splice 9 adjacent frames, and apply LDA to reduce its dimension with maximum likelihood linear transform.

First, we employ a Kaldi speech recognizer [18] to decode each utterance using the trained AM and the 3-gram LM. Second, we use the 4-gram LM for lattice re-scoring and generate n-best lists. The n-best size is at most 100 for each utterance. Finally, we use RNNLM and factored RNNLM to re-score the n-best (n=100). Since it is very time consuming to train RNNLM and factored RNNLM on large data, the usual way is to train RNNLM on a small scale of in-domain corpus. This paper also employs this setting. The corpus is automatically tagged with parts-of-speech[4]. In the fRNNLM, we investigate three commonly used types of factors: word, stem[5] and part-of-speech (POS). We set the number of hidden neurons in the hidden layer and the number of classes in the output layer for both the RNNLM and factored RNNLM to 480 and 300.

3.1. Overall results

The best re-scoring results measured by word error rate (WER) are demonstrated in Table 3. Note that RNNLM and fRNNLMs are interpolated with 4-gram LM. The weight of 4-gram LM is empirically set to 0.8 to optimize the performance on the dev2010 set.

The results show that fRNNLM$_{wsp}$ and fRNNLM$_{wp}$ significantly improve upon 4-gram LM and RNNLM. The largest absolute improvements over the 4-gram LM and RNNLM are 0.8 points. However, no significant differences are found among the factored RNNLMs with various combinations of factors. Although the size of the parts-of-speech is the smallest (only 37), they have the largest impact on our factored RNNLM. The main reason may lie in that syntactic factor (POS) has stronger complementariness to the surface word factor, while morphological factors (stem and lemma) are too similar to the word itself, limiting such complementariness. Table 4 demonstrates the re-scoring results sampled from RNNLM and fRNNLM$_{wsp}$. This table shows that the results of fRNNLM$_{wsp}$ are more grammatically fluent. Fig. 2 illustrates the absolute improvements of fRNNLM$_{wsp}$ over RNNLM for each talk in the sets of tests 2010 and 2011. Our approach improves most talks, expect talks 535, 1178 and 1183.

Table 3: n-best re-scoring performance in WER. Subscript numbers are the absolute improvements over 4-gram LM. fRNNLM$_{wsp}$ denotes the factored RNNLM incorporating the word, stem and POS.

	dev2010	test2010	test2011	test2012
4-gram LM	16.5	13.8	12.3	13.9
RNNLM	$16.3_{0.2}$	$14.0_{-0.2}$	$12.2_{0.1}$	$13.9_{0.0}$
fRNNLM$_{wp}$	15.8	13.1	11.9	13.4
fRNNLM$_{wsp}$	$15.7_{0.8}$	$13.2_{0.6}$	$11.8_{0.5}$	$13.3_{0.6}$

3.2. Free parameter & time complexity

The number of free parameters, i.e., the size of matrices U and W in Fig. 1, in the RNNLM and factored RNNLM are $(|V| + H) \times H + H \times (C + |V|)$ and $(|f^1| + ... + |f^K| + H) \times H + H \times (C + |V|)$, respectively. That means, our factored RNNLM has $(|f^1| + ... + |f^K| - |V|) \times H$ addi-

[4] http://www.nactem.ac.uk/tsujii/software.html
[5] http://tartarus.org/~martin/PorterStemmer/

Table 4: Re-scoring results sampled from RNNLM and fRNNLM$_{wp}$. * denotes deletion errors, capitalized words denote substitution errors, and underlined words show their differences. #e stands for the number of errors.

model	#e	**result**
Reference		or we'll be here all day with my childhood stories
RNNLM	5	THE WORLD we * * ARE all day with my childhood stories
fRNNLM$_{wp}$	1	or * be here all day with my childhood stories
Reference		she's painting here a mural of his horrible final weeks in the hospital
RNNLM	2	she's painting * HERO mural of his horrible final weeks in the hospital
fRNNLM$_{wp}$	0	she's painting here a mural of his horrible final weeks in the hospital
Reference		and so you are standing there and everything else is dark but there's this portal that you wanna jump in
RNNLM	7	and so you are * STAYING IN ANYTHING else is dark but there's THE SPORT ALL that you WANT TO jump in
fRNNLM$_{wp}$	5	and so you are * STAYING IN ANYTHING else is dark but there's this HORRIBLE that you WANT TO jump in
Reference		my worlds of words and numbers blur with color emotion and personality
RNNLM	4	my WORLD SO FLOATS and numbers BELAIR with color emotion and personality
fRNNLM$_{wp}$	1	my worlds of words and numbers BELAIR with color emotion and personality

tional free parameters. If the factored RNNLM only employs word factor (f^1) and POS factors (f^2), then, it has $39 \times H$ additional free parameters. In experiments, H is usually set to $300 - 1000$, $|V|$, the word vocabulary's size, is usually set to several hundreds of thousands.

The time complexities in the RNNLM and factored RNNLM are $(1 + H) \times H \times \tau + H \times |V|$ and $(K + H) \times H \times \tau + H \times |V|$, respectively. That means, the factored RNNLM has $(K - 1) \times H \times \tau$ additional computational complexity. τ is usually set to 4 or 5. This means that $H \times |V| \gg (K - 1) \times H \times \tau$, and the increased complexity can be neglected. On the contrary, our factored RNNLM converges faster and reduces training time due to the additional free parameters. Table 5 shows the training time of an iteration, the training time of all iterations, and the test time on a PC with 1006GB of memory and 24 2660MHz processors. From this table, we observe the following: (1) No significant difference of elapsed time is found between RNNLM and fRNNLM$_{wsp}$ during an iteration of training and test stage. (2) For the time of all iterations, RNNLM takes more time than fRNNLM$_{wsp}$ because it takes 16 iterations to reach a convergence and fRNNLM$_{wsp}$ uses 13 iterations. This experiment shows that although fRNNLM has more free parameters and time complexities, it saves time owing to its fast convergence.

Table 5: Elapsed time during training and test. #1 and #2 denote time of an iteration and time of all iterations during training, m=minute, s=second.

	#1	#2	time on testing tst2010
RNNLM	120m	1923m	35.7s
fRNNLM$_{wsp}$	141m	1843m	43.4s

Figure 3 demonstrates the convergence progress of RNNLM, fRNNLM$_{wp}$ and fRNNLM$_{wsp}$. From this figure, we can observe that fRNNLM$_{wsp}$ outperforms RNNLM at all iterations, however, the relative improvements decrease with increasing iterations.

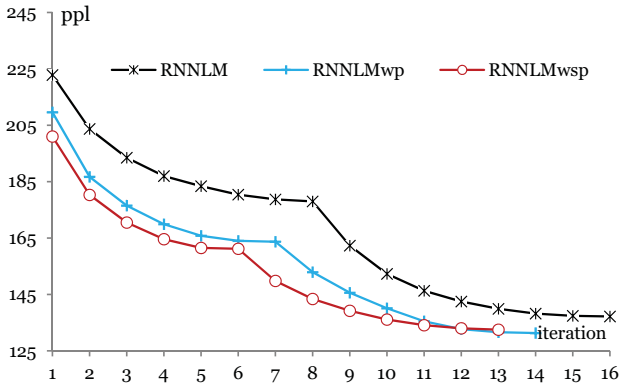


Figure 3: Convergence curve on the dev2010 set.

3.3. Training corpus size

This subsection analyzes the influence of training corpus size to RNNLM and fRNNLM$_{wsp}$. The training corpus is gradually increased by selecting sentences from the general-domain corpus [17, 20]. Note that, we change the order of the training data as follows, the training starts with the sentences selected from the general-domain data, and ends with the in-domain data. The selected sentences are also sorted in descending order of perplexities.

The results are shown in Table 6. This experiment indicates that the impacts of morphological and syntactic information become smaller with increasing of training data. The largest improvement of fRNNLM$_{wsp}$ trained on the in-domain data (2.4M words) reaches 0.8 points. However, this improvement reduces to 0.2 points when the model is trained on the larger training data (30M words).

4. Related work

Neural network language models to LVCSR were first presented in [3], which was a feed-forward NNLM with a fixed-length context consisting of projection, input, hidden, and output layers. Arisoy et al. [2] proposed a deep NNLM that uses multiple hidden layers instead of single hidden layer in

# of words in training data	dev2010		test2010		test2011		test2012	
	RNNLM	fRNNLM	RNNLM	fRNNLM	RNNLM	fRNNLM	RNNLM	fRNNLM
2.4M	16.3	15.7	14.0	13.2	12.2	11.8	13.9	13.3
9.0M	15.5	15.4	13.0	13.1	11.4	11.3	13.1	13.0
19.4M	15.4	15.3	12.9	12.9	11.3	11.2	13.0	13.0
30M	15.2	15.0	12.9	12.7	11.1	11.2	12.9	12.8

Table 6: Impact of training corpus size.

feed-forward NNLMs. Furthermore, several speedup techniques such as shortlists, regrouping and block models have been proposed [19]. Feed-forward NNLMs, which predict following word w_i based on any possible context of length n-1 history, remain a kind of n-gram language model.

Recurrent NNLM (RNNLM) [15, 16], which has different architecture at the input and output layers, can be considered as a deep neural network LMs because of its recurrent connections between input and hidden layers, which enable RNNLMs to use their entire history. Compared with feed-forward NNLMs, recurrent NNLMs reduce computational complexity and have relatively fast training due to the factorization of the output layer. Other experiments [2, 14, 13] demonstrated that RNNLM significantly outperforms feed-forward NNLM. Therefore, this paper uses RNNLM as a baseline and improves it by incorporating additional information other than surface words, such as morphological or syntactic features.

Although few studies incorporate morphological and syntactic features into RNNLM, using multiple features in language modeling is not novel. For example, Bilmes and Kirchhoff [4] presented a factored back-off n-gram LM (FLM) that assumes each word is equivalent to a fixed number (K) of factors, i.e., $W \equiv f^{1:K}$, and produces a statistic model of the following form: $p(f_i^{1:K}|f_{i-n+1:i-1}^{1:K})$. The standard back-off in an n-gram LM first drops the most distant word (w_{i-n+1} in the case of Eq. (1)), and then the second most distant word etc. until the unigram is reached. However, the factors in FLM occur simultaneously, i.e., without forming a temporal sequence, so the order in which they should be dropped is not immediately obvious. In this case, FLM creates a large space of back-off graphs that cannot be exhaustively searched. Duh and Kirchhoff [7] employed a genetic algorithm (GA) that, however, provides no guarantee of finding the optimal back-off graph. Our factored RNNLM addresses this optimization problem well, as described in Section 3. In addition, some studies [1, 2, 8, 12] introduced various syntactic features into their feed-forward NNLMs and discriminative language models.

5. Conclusion

In this paper we follow the architecture of a state-of-the-art recurrent neural network language model (RNNLM) and present a factored RNNLM by integrating additional morphological and syntactic information into RNNLM. In experiments, we investigate the impacts of three commonly used types of features on our factored RNNLM: word, stem and part-of-speech. We carry out extensive experiments to evaluate the factored RNNLM performance. Our experimental results prove that factored RNNLM consistently outperforms n-gram LM and RNNLM in terms of the IWSLT 2010∽2012 development and test data sets.

6. References

[1] Alexandrescu, A. and Kirchhoff, K. (2006). Factored neural language models. In *Proceedings of the NAACL 2006*, pages 1–4, New York, USA.

[2] Arisoy, E., Sainath, T. N., Kingsbury, B., and Ramabhadran, B. (2012). Deep neural network language models. In *Proceedings of NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, pages 20–28, Montreal, Canada.

[3] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, pages 1137–1155.

[4] Bilmes, J. A. and Kirchhoff, K. (2003). Factored language models and generalized parallel backoff. In *Proceedings of NAACL 2003*, pages 4–6, USA.

[5] Boden, M. (2002). A guide to recurrent neural networks and backpropagation. In *The Dallas Project, Sics Technical Report*.

[6] Chelba, C. and Jelinek, F. (1998). Exploiting syntactic structure for language modeling. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 225–231, Montreal, Canada.

[7] Duh, K. and Kirchhoff, K. (2004). Automatic learning of language model structure. In *Proceedings of COLING 2004*, pages 148–154, Geneva, Switzerland.

[8] Emami, A. and Jelinek, F. (2004). Exact training of a neural syntactic language model. In *Proceedings of ICASSP 2004*, pages 245–248, Montreal, Canada.

[9] Federico, M., Bentivogli, L., Paul, M., and Stuker, S. (2011). Overview of the iwslt 2011 evaluation campaign. In *Proceedings of IWSLT 2011*, pages 11–27, San Francisco, USA.

[10] Goodman, J. (2001). Classes for fast maximum entropy training. In *Proceedings of ICASSP 2001*, Utah, USA.

[11] Khudanpur, S. and Wu, J. (2000). Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling. *Computer Speech and Language*, pages 355–372.

[12] Kuo, H.-K. J., Mangu, L., Emami, A., Zitouni, I., and Lee, Y.-S. (2009). Syntactic features for arabic speech recognition. In *Proceedings of Automatic Speech Recognition & Understanding (ASRU) 2009*, pages 327–332, Merano, Italy.

[13] Kuo, H.-K. J., Arisoy, E., Emami, A., and Vozila, P. (2012). Large scale hierarchical neural network language models. In *Proceedings of Interspeech 2012*.

[14] Mikolov, T., Anoop, D., Stefan, K., Burget, L., and Cernocky, J. (2011a). Empirical evaluation and combination of advanced language modeling techniques. In *Proceedings of INTERSPEECH 2011*, pages 605–608, Florence, Italy.

[15] Mikolov, T., Karafiat, M., Burget, L., Cernocky, J. H., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Proceedings of INTERSPEECH 2010*, pages 1045–1048, Makuhari, Japan.

[16] Mikolov, T., Kombrink, S., Burget, L., Cernocky, J., and Khudanpur, S. (2011b). Extensions of recurrent neural network language model. In *Proceedings of ICASSP 2011*, pages 5528–5531, Prague, Czech Republic.

[17] Moore, C., Lewis, W. (2010). Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220C224, Hawaii, USA.

[18] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

[19] Schwenk, H. (2007). Continuous space language models. *Computer Speech and Language*, 21(3):492–518.

[20] Schwenk, H., Rousseau A., and Attik, M. (2012) Large, Pruned or Continuous Space Language Models on a GPU for Statistical Machine Translation. In *Proceedings of NAACL workshop on the Future of Language Modeling*, pages 11–19, Canada.

[21] Stolcke, A. (2002). Srilm - an extensible language modeling toolkit. In *Proceedings of INTERSPEECH 2002*, pages 901–904, Colorado, USA.

[22] Xu, P., Chelba, C., and Jelinek, F. (2002). A study on richer syntactic dependencies for structured language modeling. In *Proceedings of ACL 2002*, pages 191–198, Philadelphia, USA.

[23] Xu, P. and Jelinek, F. (2004). Random forests in language modeling. In *Proceedings of EMNLP 2004*, pages 325–332, Barcelona, Spain.

Incremental Adaptation Using Translation Information and Post-Editing Analysis

Frédéric Blain[*†], *Holger Schwenk*[*] *Jean Senellart*[†]

[*] LIUM
Université du Maine, Avenue Laennec
72085 Le Mans, France
lastname@lium.univ-lemans.fr

†Systran SA
5, rue Feydeau
75002 Paris, France
lastname@systran.fr

Abstract

It is well known that statistical machine translation systems perform best when they are adapted to the task. In this paper we propose new methods to quickly perform incremental adaptation without the need to obtain word-by-word alignments from GIZA or similar tools. The main idea is to use an automatic translation as pivot to infer alignments between the source sentence and the reference translation, or user correction. We compared our approach to the standard method to perform incremental re-training. We achieve similar results in the BLEU score using less computational resources. Fast retraining is particularly interesting when we want to almost instantly integrate user feed-back, for instance in a post-editing context or machine translation assisted CAT tool. We also explore several methods to combine the translation models.

1. Introduction

Due to multiplication of resources and the diversity of languages, Machine Translation (MT) systems are widely used as a precious help for human translators. Most of the systems used today are based on the statistical approach. Those systems extract all the knowledge from the provided data. Nevertheless, these systems have some limits: first, the specific resources available at *t* time could be less appropriate at *t+1*. Consequently, they need to be regularly re-trained in order to be updated, which is usually computationally demanding. The goal of incremental adaptation is then twofold: to adapt the system on the fly when new resources are available without re-training the entire system.

Post-Editing (PE) the output of SMT systems is widely used, amongst others, by professional translators of localization services which need for example to translate technical data in specific domains into several languages. However, the use of PE is restricted by some aspects that must be taken into consideration. As resumed by [1], the time spent by the post-editor is a commonly used measure of the PE effort, which should not to be, in case of poor translation quality, more important than translation from scratch. Even if this temporal aspect can be see as the most important, PE effort can be evaluated using automatic metrics based on the edit distance. These metrics commonly use the number of required edits of the MT system output to reach a reference translation. From then, the combination of PE and incremental adaptation can be seen as a way to reduce the task effort by allowing a MT system to gradually learn from its own errors. Especially considering the repetitive nature of the task highlighted by [2].

However, incremental adaptation is still a tricky task: how to adapt the system correctly? Adaptation should not degrade system performance and accuracy. Some approaches are possible and we will try to see the impact of several of them in the second part of this article.

First of all, we present a new experimental approach for incremental adaptation of a MT system using PE analysis. Starting from a generic baseline, we have gradually adapted our system by translating an in-domain corpora which was split beforehand. Each part of the corpora was translated using the translation model adapted at the previous step, *i.e.* updated with new extracted phrases. These phrases are the result of a word-to-word alignment combination we present afterward.

1.1. Similar work

The most similar approach in the literature is proposed in [3] who present an incremental re-training algorithm to simulate a post-editing situation. It is proposed to extract new phrases from *approximate alignments* which were obtained by a *modified* version of Giza-pp [4]. An initial alignment with one-to-one links between the same sentence positions is created and then iteratively updated as long as improvements are observed. In practice, a greedy search algorithm is used to find the locally optimal word alignment. All source positions carrying only one link are tried, and the single link change which produces the highest probability increase according to the Giza-pp model 4 is kept. The resulting alignment is improved with two simple post-processing steps. First, each unknown word in source side is aligned with the first non-aligned unknown word on the target side. Second, unaligned pairs of positions surrounded by corresponding alignments are automatically aligned.

In this paper, we present a very fast word-to-word align-

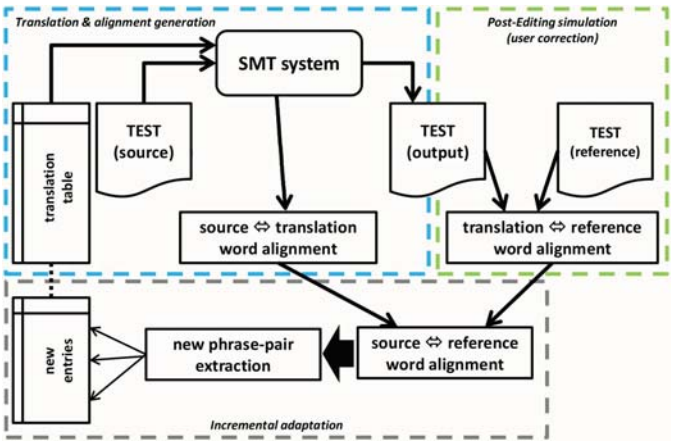


Figure 1: Incremental adaptation workflow in three steps protocol: 1. Translation and source-translation alignment: source sentences are translated using the SMT system Moses. Alignment links are generated during the translation step; 2. Edit distance on translation-reference: MT system output and its reference translation are aligned using edit distance algorithm of TER; 3. Source-reference alignment: the alignment links are deduced from combination of alignments of both step 1 and 2. Phrase pairs are then extracted, scored and added to translation model which is finally re-trained.

ment algorithm which is partially based on the edit-distance algorithm. As argued in [3], *"to be practical, incremental retraining must be performed in less than one second"*. For comparison, our entire alignment process takes few hundredths of second for 1500 sentences, in comparison to several seconds per sentences as reported in [3].

[5] present stream based incremental adaptation using an on-line version of the EM algorithm. This approach designed for large amounts of incoming data is not really adapted for the post-editing context. Like [3], we propose an incremental adaptation workflow that is more oriented to real time processing.

As part of our experiments, we have compared our approach with the use of the freely available tool named Inc-Giza-pp,[1] an incremental version of Giza-pp. It is precisely intended to inject new data into an SMT system without having to restart the entire word alignment procedure. To our knowledge, this is the standard method currently used in the field. In our experiments, we achieve similar results with respect to the BLEU score using less time.

The reminder of this paper is organized as follows. In the next section we first describe our incremental adaptation workflow and more particularly the word-to-word alignment methodology based on the edit distance. Section 3 is dedicated to the experimental protocols and compares the performance of our approach with the standard method using Inc-Giza-pp. The paper concludes with a discussion of perspectives of this work.

[1] http://code.google.com/p/inc-giza-pp/

2. Incremental Adaptation Workflow

In this paper, we present a new methodology to perform incremental training and domain adaptation. Starting with a generic phrase-based MT baseline system (PBMT), we have sequentially translated the source side of an in-domain corpus. At each step, like [3], we have simulated a human post-editing the translations by using the corresponding reference translations of the data. At the sentence level, the source and its reference translation are aligned in order to subsequently retrieve the corresponding phrase pairs. The extracted phrase pairs are then scored and used to retrain (i.e. adapt) the translation model of our PBMT system.

We have developed an aligning protocol which operates in three steps, named *"translation"*, *"analysis"* and *"adaptation"*. These three steps are linked together by a word-to-word alignment algorithm which allows us to align a source and its reference translation and then, to extract new phrase pairs with which the MT system will be adapted. This algorithm is illustrated in Figure 1 and explained in details in the next section.

2.1. Word-to-word alignment combination

Our approach to align the source and its corresponding reference translation could be seen as a combination of the source to hypothesis word alignments and an analysis of the edit distance between the hypothesis and the reference. The central element of this approach is an automatic translation of the source sentence into the target language. The principle of this idea is illustrated in Figure 2.

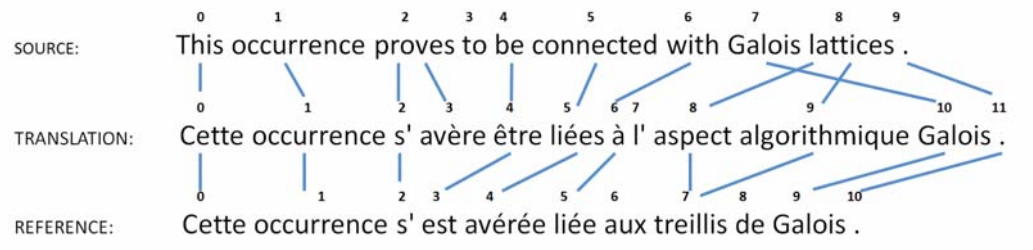


Figure 2: Example of a source-to-reference alignment using using the automatic translation as pivot. The alignment links between the source sentence and the translation are generated by the MT system. Those between the translation and its post-edited version (i.e. the reference) are calculated by TER. Finally, the source-to-reference alignment links are deduced by an alignment combination based on both alignment sets computed before.

2.1.1. Translation: source to translation alignment

The SMT system used to translate the source sentences is based on the Moses SMT toolkit [6]. Moses can provide the word-to-word alignments between the source sentence and the translation hypothesis. This aligning information represents the first part of our alignment combination. This automatic translation is "compared" with the reference translation using an edit distance algorithm.

2.1.2. Analysis: edit distance alignment

In this paper, we use the Translation Error Rate (TER) algorithm as proposed in [7]. TER is an extension of the Word Error Rate (WER) which is more suitable for machine translation since it can take into account word reorderings. TER uses the following edit types: *insertion, deletion, substitution* and *shift*.

The TER is computed between the output of our SMT system and the corresponding reference translation, and the word-to-word alignments are inferred. We only keep the aligned and substituted edit types in order to extract what we consider as the most interesting phrase pairs. Indeed, we argue that what is aligned correspond to what our system knows, while what is substituted correspond to what our system does not know.

Our approach can be extended to use TER-Plus [8], an extension of TER using paraphrases, stemming and synonyms in order to obtain better word-to-word alignments.

2.1.3. Adaptation: source to reference alignment

Considering the SMT translation hypothesis as a "pivot" for aligning both source and its reference sentence, we have designed the word-to-word alignment algorithm shown by Algorithm 1. It combines source-to-translation and translation-to-reference alignments, and then deduces the source-to-reference alignment path. From this path, the translation model is finally updated using the standard training phrase extraction and scoring script provided with Moses.

```
Data: src-to-tgt word alignments, tgt-to-ref edit-path
foreach src-to-tgt word alignment do
    alignment(src-word, tgt-word) = 1;
end
if edit-path has shift then
    foreach shift do
        updateWordPosition(tgt, shift);
    end
end
foreach edit-type of edit-path do
    if edit-type is 'align' or 'substitution' then
        alignment(tgt-word, ref-word) = 1;
    end
end
foreach ref-word of ref do
    foreach tgt-word aligned to ref-word do
        if isAligned?(src-word, tgt-word) then
            alignment(src-word, ref-word) = 1;
        end
    end
end
```

Algorithm 1: Source-to-reference alignment algorithm at word level. Using both source-to-translation alignments and translation-to-reference edit-path, the source-to-reference alignments path are build.

3. Experimental evaluation

The approach described in the previous section is compared to inc-Giza-pp which is considered as the state-of-the-art tool for incremental training. In our first experiments, each system uses a single translation model which is updated and entirely retrained after each iteration. For the results we present hereinafter, the system with inc-Giza-pp will be called "inc-Giza-pp" and the system with our approach will be called "noGizapp".

3.1. Training data

The experiments were performed on data which was made available by the French COSMAT project. The goal of this project is to provide task-specific automatic translations of scientific texts on the French HAL archive.[2] This archive contains a large amount of scientific publications and PhD Thesis. The MT system is closely integrated into the workflow of the HAL archive. In particular, the author has the possibility to correct the provided automatic translations. These translations will be then used to improve the system. In this paper, we consider the automatic translation from English into French.

Three corpora of parallel data are available to train the translation model: two generic corpora and an in-domain corpus for adaptation. The two first corpora are Europarl and News Commentary with 50 million and 3 million words, respectively. They were used to train our SMT baseline systems. The third corpus, named "absINFO", contains 500 thousand words randomly selected from abstracts of scientific papers in the domain of Computer Science. Information on the sub-domains is also available (networks, AI, data base, theoretical CS, ...), but was not used in this study. The corpus if freely available to support research in domain adaptation and was already used by the 2012 JHU summer workshop on this topic. A detailed description of this corpus can be found in [9].

This in-domain corpus was split into three sub-corpora:

- **absINFO.corr.train** is composed of 350k words and is used to simulate the user post-editing or corrective training.
- **absINFO.dev** is a set of 75k words and used for development.
- **absINFO.test** another set of 75k words used as a test corpus to monitor the performance of our adaptation workflow.

Moreover, in order to better simulate a sequential post-editing process, the absINFO.corr.train corpus was split into 10 sub-sets (about 1.5k sentences with 35k words each). This corresponds quite well to the update of an MT system after a post-correction of an entire document.

3.2. Baseline Training

The baseline SMT systems were constructed using the standard Moses pipeline and Giza-pp for word alignment. In order to later use Inc-Giza-pp, the incremental version of Giza-pp, we had to train a specific baseline system using the Hidden Markov Model (HMM) word alignment model option. However, to make a fair comparison of the two adaptation techniques, the baseline and following systems were trained on the same data and tuned with MERT [10] with the same initial parametrization. The inc-Giza-pp and noGizapp baseline SMT systems achieve a BLEU score of 35.27 and 35.32 BLEU points on the development corpus respectively, and 31.89 and 32.27 BLEU points on the test corpus.

3.3. Analysis of processing time and alignment quality

The two incremental training approaches are compared with respect to the BLEU score obtained by adding the additional aligned data. We also report the time needed to perform the word alignments. For inc-Giza-pp, the alignment protocol is composed of several steps (for more details, see "Incremental Training" of the "Advanced Features" section in Moses user documentation.[3]) First, one has to preprocess the data for use by Giza-pp. This involves updating the vocab files, converting the sentences into the *snt* format of Giza-pp, and then, updating the co-occurrence file. Then, Giza-pp is executed to update and compute the alignments for the new data. This is performed in both directions, source-to-translation and translation-to-source. For each iteration of our experiment, this process takes about 14 minutes.

For the noGizapp system, the required time to perform the source-to-translation alignment can be considered as null because it is implicitly achieved during the translation. The TER between the SMT translation and the reference translation is computed using a fast and freely available C++ implementation.[4] This tool can align about 35k words in about three seconds (corresponding to 1.5k sentences in the 10% subset of the absINFO.corr.train corpus). The alignment combination of the source and reference translation, described in algorithm 1, takes less than a second. Overall, we can obtain the source-to-reference alignments of 35k words in a few seconds only.

The BLEU scores on the development (left part) and test data (right part) are compared in Figure 3. The following systems were built:

Gizapp for each subcorpus of the absINFO.corr.train training data (10%, 20%, 30%...100%), all the available training data is concatenated and the full training pipeline is performed, including a new word alignment which considers **all** the training data. We consider this as the upper limit of the performance we could achieve by incremental training. This procedure is very time consuming.

inc-Giza-pp the subcorpora of of the absINFO.corr.train training data are added using the incremental version of Giza. This resulted in a slight decrease of the BLEU score on the development data and a quite unstable performance on the Test data.

noGizapp incremental training using the new approach described in this paper. We always used the same base-

[2] http://hal.archives-ouvertes.fr/

[3] Available online: http://www.statmt.org

[4] http://sourceforge.net/projects/tercpp/

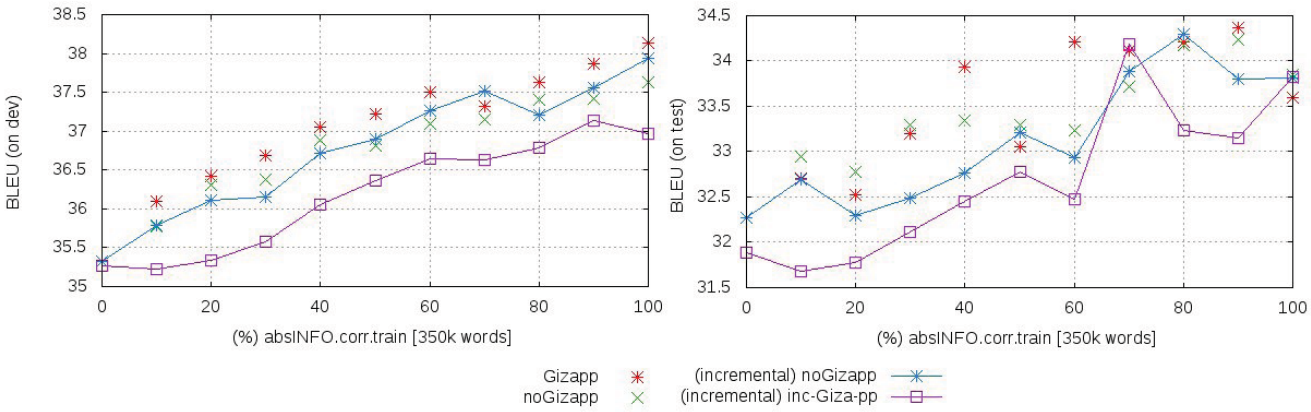


Figure 3: Incremental adaptation in BLEU score for our two PBMT systems on both development and test corpora. The Inc-Giza-pp system uses incremental version of Giza-pp for aligning sentence pair, while noGizapp system uses the approach we present in this paper, which is based on translation information and edit distance combination. The 'Gizapp' and 'noGizapp' curves represent the BLEU score obtained with a in-domain adaptation of our baseline systems, without incremental approach. While the curves 'Inc-Giza-pp' and '(incremental) noGizapp' represent the in-domain adaptation scores over an incremental process.

line SMT system to translate the additional adaptation data.

inc-noGizapp like *noGizapp*, but using the system adapted in the previous step to translate the additional adaptation data.

The proposed approach to obtain incremental word alignments achieves slightly better BLEU scores on both the development and the test corpus, but performs much faster.

The large variations on the test corpus could be explained by two potential reasons. The first one could be the characteristics of the absINFO.corr.train corpus. It was created from abstracts of (Computer Science) sub-domains which were randomly selected. Consequently, a sub-corpus predominantly represented in a sub-corpus of absINFO corpus could be not represented in the test corpus. The second reason could be the use of only one translation model. As explained above, this translation model is updated with new phrase pairs extracted from each iteration. Because we are only interested by edit types corresponding to 'align' and 'substitution' edit type during the edit distance analysis (see Section 2.1.2), the extracted phrase pairs could be generic or in-domain. Added to all entries already in the translation model, these new phrases disturb the probability distribution. This could also explain why our incremental systems are performing worse than the non incremental systems (what we have called "oracle systems") for which, the probability distribution is tuned in better way.

Another possibility could be to use two translation models like [3]. In this way, we can quickly create a phrase-table from the word alignments of the additional data.

3.4. Combination of translation models

In this section, we present results achieved by combining several translation models. The techniques described in the previous sections can significantly speed-up the word-alignment process, in comparison to running incremental Giza-pp, but we still need to create a new phrase table on all the data. Therefore, we propose to create a new phrase table on the newly added data only and to combine it with the original unadapted phrase table.

3.4.1. Back-off Models

Moses support several modes to use multiple phrase tables. We first explored the back-off mode which favors the principal phrase table: the second phrase table is only considered if the word or phrase is not found in the first one. Figure 4. The dotted curve represents the use of the incrementally trained in-domain translation model with the generic one as back-off. The crossed curve represents the use of these same models but in reverse order.

As we can see, we got very different results depending on which translation model is used first, but this can be easily explained by the nature of the back-off models. Our in-domain translation model is built with the incrementally added data only, i.e. very small amounts of data, in particular during the first iterations.

Figure 5 presents when jointly using both translation models. In this configuration, separate translation options are created for each occurrence, the score being combined if the same translation option is found in both translation models. Compared to the use of only one translation model, we

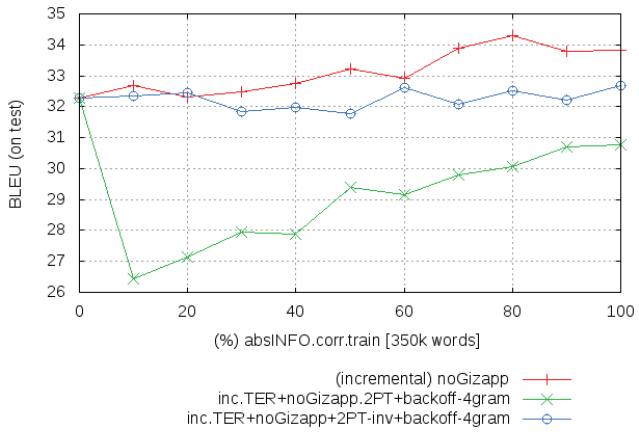


Figure 4: Results for use of "back-off" models. The crossed curve represents our PBMT system using only one translation model while the dotted and third curves represent respectively the impact of use two back-off models but in different order.

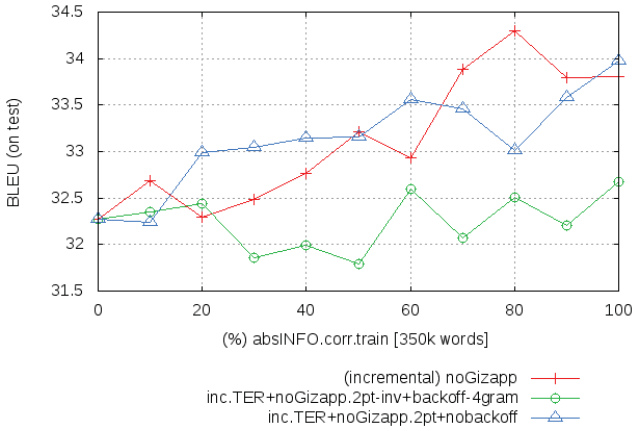


Figure 5: Comparison between use of back-off (dotted curve) and non back-off models. The crossed curve represents our PBMT system using only one translation model. The third curve represents a PBMT system using its both translation models for the decoding path while the dotted curve shows our results for using our translation models in back-off mode.

can observe a significant degradation near 80% of adaptation data before finally achieving a similar final BLEU score (up to +0.2 points) compared to inc-Giza-pp and noGizapp.

Once again, we believe that the nature of our absINFO corpus may explain the evolution of our score. When our SMT systems has to translate more generic sentences, it is likely that the translation options were provided by our generic translation rather than our in-domain model.

Based on this observation, we tried to limit edit distance analysis to substitutions only.

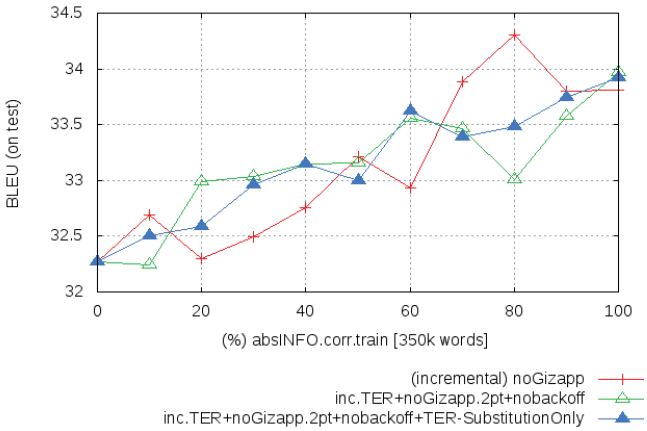


Figure 6: Use of 2 translations models with noback-off and only substitution were kept, or not.

3.4.2. Filtering by edit-distance type

The Figure 6 shows the results obtained with an in-domain translation model only trained from substitutions which were detected during the edit distance analysis. As we argued in section 2.1.2, we consider that the "substitution" edit type corresponds to what the MT system does not know since it was necessary to fix its output.

As we can see, the previous degradation is less important.Overall, the evolution of the BLEU score is smoother than for the other approaches tested so far. By keeping the phrase pairs corresponding to substitutions only (in the edit-path), we have also limited the contextual phrases in our in-domain translation model. It should also take into account the alignment errors that would have a more important impact in this configuration on the quality of the translation model.

3.4.3. N-best alignment generation

One of the key points presented in this paper is the use of the translations to generate the alignment links between a source sentence and its translation generated by the system. By default, our MT system returns the best translation candidate after decoding. This means that this translation has obtained the highest decoding score, but that does not necessarily mean that the alignment associated with it is the best one.

Based on this observation, we tried to explore the n most likely translations hypothesis (n-best list). Indeed, a source sentence could be translated into the same translation using different segmentations into phrase-pairs. With our approach, for the same sentence-translation pair, if we have multiple alignment candidates, we can generate more source-to-reference alignments and then, potentially reinforce our in-domain translation model. Using only the two best non distinct translation candidates, we obtained the results shown

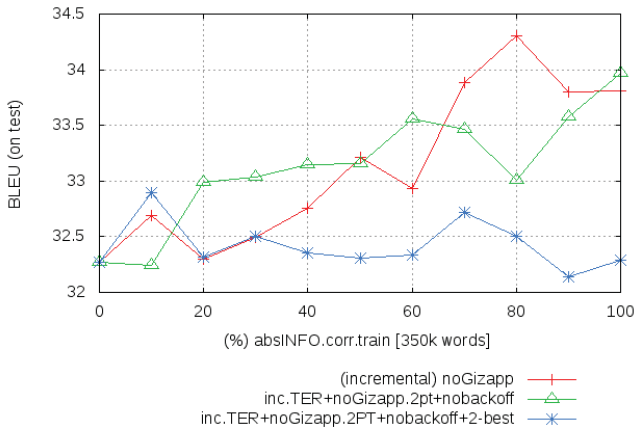


Figure 7: Use of n-best translation candidate to reinforce alignment possibilities and then, extend our phrase-pair generation. The starred curve presents our PBMT system for which we used the two first translation candidates in order to extract phrase pairs, while the second curve represents the same system but only the 1-best translation candidate is used.

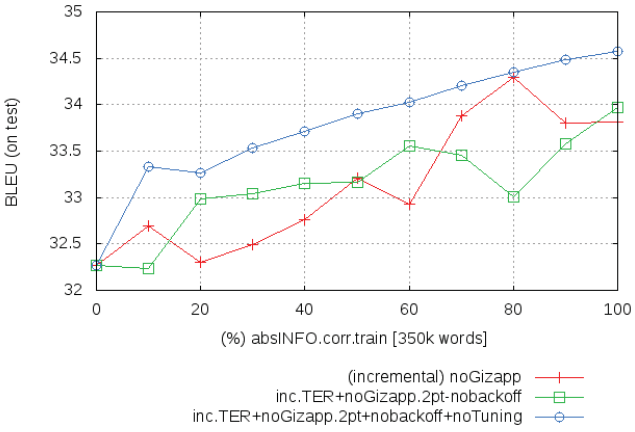


Figure 8: Results for incremental adaptation with no tuning step. The squared curve represents a PBMT system with normal tuning process achieved at each adaptation iteration, while the dotted curve represents the same system for which the tuning weights obtained at 10% level remain stable throughout the entire adaptation.

in Figure 7. Unfortunately, the results are worse than expected. In future work, we will investigate other options to use the information in the n-best lists.

3.4.4. No tuning step

In the final part of the paper, results from an incremental adaptation of a PBMT system without tuning step are presented. This procedure is very time-efficient and stable since we do not apply tuning at every adaptation step. We argue that we do not need to re-tune our models since adaptation only adds small amounts of information. Tuning is only applied at the creation of the model, and the resulting parameters are maintained during the adaptation process. The results of this procedure are shown in Figure 8.

First, we can observe a clear difference between the squared and the dotted curves for the 10% adaptation level, even though they result from the same approach. This is due to the baseline that we applied: By default, our PBMT system is a translation model using only one phrase table. We need to tune however on a "new baseline system" using two phrase tables (the one at the 10% level), for which the tuning weights obtained remain stable throughout adaptation.

Second, the resulting curve is rather smooth, indicating the instability of the tuning process.
To sum up, by applying our incremental adaptation, we obtain a clear improvement in BLEU scores (+0.5 points), however without the need to retune at every adaptation. Tuning can be performed in larger time intervals, for example - in an industrial post-editing context - every night or as soon as processing resources become available.

4. Conclusion and Future Work

In this paper, we have presented a new word-to-word alignment methodology for incremental adaptation using a phrase-based MT system. This method uses the information generated during the translation step and then relies on an analysis of a (simulated) post-editing step to infer a source-to-reference alignment at the word level.

Compared to incremental Giza, the standard method currently used in the field, the first part of our experiments show that our approach allows us to obtain similar performance in the BLEU score at an significantly improved speed. Incremental Giza needs several minutes to align two corpora of about 35k words while the approach proposed in this paper runs in some seconds. Our approach could be therefore integrated into an interface dedicated to post-editing which would exploit user feedback in real time.

The second part of this article was dedicated to experiments on translation model combination. These experiments show that we can get better results by jointly using two translation models instead of only one. The results of these experiments suggest some directions for future research. For example, the use of the TER algorithm for analyzing the post-editing result could be reinforced by the notion of "Post Edit Actions" introduced by [2], in order to better identify errors of the SMT system.

5. Acknowledgment

This research was partially financed by the DGA and the ANRT under CIFRE-Defense 7/2009, the french ANR project COSMAT under ANR-09-CORD-004, and the European Commission under the project MATECAT, ICT-

2011.4.2 – 287688.

6. References

[1] M. Koponen, "Comparing human perceptions of post-editing effort with post-editing operations," *Proceedings of the Seventh Workshop on Statistical Machine Translation*, p. 181–190, June 2012.

[2] F. Blain, J. Senellart, H. Schwenk, M. Plitt, and J. Roturier, "Qualitative analysis of post-editing for high quality machine translation," in *Machine Translation Summit XIII*, A.-P. A. for Machine Translation (AAMT), Ed., Xiamen (China), 19-23 sept. 2011.

[3] D. Hardt and J. Elming, *Incremental Re-training for Post-editing SMT.*, 2010.

[4] F. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[5] A. Levenberg, C. Callison-Burch, and M. Osborne, "Stream-based translation models for statistical machine translation," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 394–402.

[6] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Annual meeting-association for computational linguistics*, vol. 45, no. 2, 2007, p. 2.

[7] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of Association for Machine Translation in the Americas*, 2006, pp. 223–231.

[8] M. Snover, N. Madnani, B. Dorr, and R. Schwartz, "Fluency, adequacy, or hter? exploring different human judgments with a tunable mt metric," in *Proceedings of the Fourth Workshop on Statistical Machine Translation*, vol. 30. Association for Computational Linguistics, 2009, pp. 259–268.

[9] L. Patrik, H. Schwenk, and F. Blain, "Automatic translation of scientific documents in the hal archive," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012, pp. p.3933–3936.

[10] F. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 160–167.

Interactive-Predictive Speech-Enabled Computer-Assisted Translation

Shahram Khadivi, Zeinab Vakil

Human Language Technology Lab, Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
{Khadivi, Z.Vakil}@aut.ac.ir

Abstract

In this paper, we study the incorporation of statistical machine translation models to automatic speech recognition models in the framework of computer-assisted translation. The system is given a source language text to be translated and it shows the source text to the human translator to translate it orally. The system captures the user speech which is the dictation of the target language sentence. Then, the human translator uses an interactive-predictive process to correct the system generated errors. We show the efficiency of this method by higher human productivity gain compared to the baseline systems: pure ASR system and integrated ASR and MT systems.

1. Introduction

Nowadays, with the expansion of global communications, the need for the translation has become a basic and important requirement, especially for international institutions and news agencies. Consider the following example to illustrate the importance of the translation in today world. In 2003, after the enlargement of the European Union, with a population of 453 million, the cost of the translation at all institutions, once translators are operating at full speed, was estimated at 807 M€ per year.

Recently, significant improvements have been achieved in statistical machine translation (MT), but still even the best machine translation technology is far from replacing or even competing with human translators. In order to achieve high quality translations, translated texts by these systems need to be reviewed and corrected by a human translator.

Another way to increase the productivity of the translation process is computer-assisted translation (CAT) system. In a CAT system, the human translator begins to type the translation of a given source text; by typing each character the MT system interactively offers the choices to enhance and complete the translation. Human translator may continue typing or accept the whole completion or part of it.

Interactive machine translation (IMT), first appeared as part of Kay's MIND system [1], where the user's role was to help with source-text disambiguation by answering questions about word sense, pronominal reference, prepositional-phrase attachment, etc. Later work on IMT, eg [2,3,4], has followed in this vein, concentrating on improving the question/answer process by having less questions, more friendly ones, etc. Despite progress in these endeavors, the question/answer process remained in the systems of this sort. Finally these systems are only used where the cost of manually producing a translation is high enough to justify the extra effort, for example when the user's knowledge of the target language may be limited or non-existent, or when there are multiple target languages. With introducing TransType project by [5], a major change in how the user interacts with the machine had occurred. In such an environment, human translators interact with a translation system that acts as an assistance tool and dynamically provides a list of translations (suffixes) which complete the part of the source sentence already translated (prefix). Also from 1997 to 2004, most of the given papers related to the various versions of the TransType project such as [6,7,8,9].

Also one desired feature of a computer-assisted translation system is to provide an environment to accept the translator's target language speech signal to speed up the translation process; since professional translators can translate a given text faster by dictation rather than directly typing the translation [10]. In such a system, two sources of information are available to recognize the speech input; the target language speech and the given source language text. The target language speech is just a human-produced translation of the source language text. Machine translation models are used only to take into account the source text in order to increase the speech recognition accuracy. The overall schematic of automatic text dictation in computer-assisted translation is depicted in Figure 1.

The idea of incorporating statistical machine translation and speech recognition models was independently initiated about one decade ago by two groups: researchers at the IBM Thomas J. Watson Research Center [10] and researchers involved in the TransTalk project [11] and [12].

In [10], the authors described the statistical speech recognition models and statistical translation models. Then, they proposed a method for combining those models, but they did not report any recognition or translation results. Instead, they just reported the perplexity reduction when the translation models were combined to recognition models.

In the TransTalk project [11] and [12], the authors reported three different combination methods between translation and recognition models. The first method was capable only of isolated word recognition. In the second method, the speech recognition system generates a list of the most probable word sequence hypotheses. Then the statistical translation models rescore them and select the best word sequence hypothesis. The idea behind the third method was the dynamic vocabulary for a speech recognition system which translation models generated for each source language sentence. The best recognition results have been achieved with the second method, while the third method was faster. The authors have shown the promising results of combining the translation models to speech recognition models. However, they neither described the details of the utilized translation model nor studied the impact of different translation models. Also recently, some researcher in [13,14,15,16,17] have studied the integration of ASR and MT models but in the any of these works haven't been used from interactive framework. For the first time, in this paper, we enter interactive form into a speech enabled CAT and create a Speech-Enabled Interactive CAT. In this new system, the human translator uses an interactive-predictive process to correct the system generated errors.

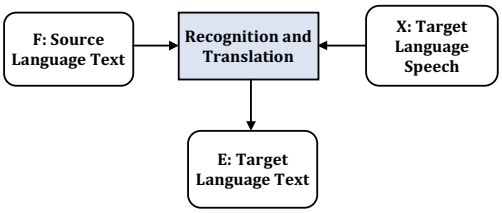


Figure 1: Schematic of automatic text dictation in computer-assisted translation

2. Models of interactive-predictive speech-enabled CAT

In a speech-enabled interactive-predictive computer-assisted translation system, we are given a source language sentence $F = f_1 \dots f_j \dots f_J$, an acoustic signal $X = x_1 \dots x_t \dots x_T$ that is the speech of the target language sentence, and the correct translated part of the target language sentence (prefix) $E_p = e_1 \dots e_i$. Then, we generate the best complement for the target sentence prefix (suffix) $E_s = e_{i+1} \dots e_I$. Among all possible target language sentence suffixes, we will choose the sentence with the highest probability:

$$\hat{E}_s = \text{argmax}_{E_s}\{P(E_s, E_p, F, X)\} \quad (1)$$

$$= \text{argmax}_{E_s}\{P(E_p, E_s, F).P(X|E_p, E_s, F)\} \quad (2)$$

$$= \text{argmax}_{E_s}\{P(E_s, E_p).P(F|E_p, E_s).P(X|E_p, E_s)\} \quad (3)$$

$$= \text{argmax}_{E_s}\{P(E_s|E_p).P(F|E_p, E_s).P(X|E_p, E_s)\} \quad (4)$$

Equation 2 is simplified into Equation 3 by assuming that there is no direct dependence between X and F. The decomposition into three knowledge sources in Equation 4 allows an independent modelling of the target language model $P(E_s|E_p)$, the translation model $P(F|E_p, E_s)$ and the acoustic model $P(X|E_p, E_s)$.

The target language model describes the well-formedness of the target language sentence. The translation model links the source language sentence to the target language sentence. The acoustic model links the acoustic signal to the target language sentence. The argmax operation denotes the search problem, i.e. the generation of the output sentences in the target language by maximization all possible target language sentences. Another approach for modelling the posterior probability $P(E_s|E_p, F, X)$ is direct modelling by the use of a log-linear model. The direct posterior probability is given by:

$$P(E_s|E_p, F, X) = \frac{\exp[\sum_{m=1}^{M} \lambda_m h_m(E_s, E_p, F, X)]}{\sum_{\hat{E}_s} \exp[\sum_{m=1}^{M} \lambda_m h_m(\hat{E}_s, E_p, F, X)]} \quad (5)$$

This approach has been suggested by Papineni et al. in [18.19], for natural language understanding task; by Beyerlein in [20], for automatic speech recognition; and in [21] for statistical machine translation. The time-consuming renormalization in Equation 5 is not needed in the search. Therefore we obtain the following decision rule:

$$\hat{E}_s = \text{argmax}_{E_s} \sum_{m=1}^{M} \lambda_m h_m(E_s, E_p, F, X) \quad (6)$$

Each of the terms $h_m(E_s, E_p, F, X)$ denotes one of the various models which are involved in the recognition process. Each individual model is weighted by its model scaling factor λ_m. As there is no direct dependence between F and X, the $h_m(E_s, E_p, F, X)$ can be in one of these two forms: $h_m(E_s, E_p, X)$ and $h_m(E_s, E_p, F)$.

This approach is a generalization of Equation (6). The direct modeling has the advantage that additional models or feature functions can be easily integrated into the overall system. Based on Equation (4), the principal models which will contribute to the final system are the acoustic model, the language model, and the translation model(s). We may use one or more translation models in the final system. A set of possible translation models consists of *HMM*, *IBM-1*, *IBM-2*, *IBM-3*, *IBM-4*, *IBM-5*, and *Alignment Template* models, which will be described in Section 3. The details of utilized acoustic and language models will be explained in Section 4.

The model scaling factors λ_1^M in Equation 5 are trained according to the maximum entropy principle, e.g. using the GIS algorithm. Alternatively, one can train them with respect to the final recognition quality measured by the word error rate [22]. The development of an efficient search algorithm for integrating automatic speech recognition and statistical machine translation models is very complicated. Thus, in order to facilitate the implementation of the above log-linear model, we use the principle of *N*-best rescoring instead of implementing a new search algorithm. The *N*-best rescoring approach helps us to quickly examine many different dependencies and models for the combination of automatic speech recognition and statistical machine translation.

The recognition process is performed in two steps. In the first step, the baseline speech recognition system creates an *N*-best list of length *N* for every utterance *X* of the given corpus. In the second step, the translation models rescore every sentence pair (the entries in the *N*-best list with their corresponding source sentence). For each utterance, the decision about the best recognized sentence is made according to the recognition and the translation models. Then the implementation approach is very similar to the second method explained in [12].

3. Translation models

A key issue in modeling the translation model probability $P(F|E_p, E_s)$ is the question of how we define the correspondence between the words of the target sentence and the words of the source sentence. In typical cases, we can assume a sort of pairwise dependence by considering all word pairs (f_j, e_i) for a given sentence pair $(f_1^J; e_1^I)$. A family of such *alignment models* (IBM-1,..., IBM-5) was developed in [23]. Using the similar principles as in Hidden Markov models (HMM) for speech recognition, we re-write the translation probability by introducing the *hidden alignments* $\boldsymbol{\mathcal{A}}$ for each sentence pair $(f_1^J; e_1^I)$:

$$Pr(f_1^J|e_1^I) = \sum_{\boldsymbol{\mathcal{A}}} Pr(f_1^J, \boldsymbol{\mathcal{A}}|e_1^I) \quad (7)$$

IBM-1,2 and Hidden Markov Models. The first type of alignment models is virtually identical to HMMs and is based on a mapping $j \rightarrow i = a_j$, which assigns a source position j to a target position $i = a_j$. Using suitable modeling assumptions [22,23], we can decompose the probability $Pr(f_1^J, \boldsymbol{\mathcal{A}}|e_1^I)$ with $\boldsymbol{\mathcal{A}} = a_1^J$:

$$Pr(f_1^J, a_1^J|e_1^I) = p(J|I).\prod_{j=1}^{J}\left[p(a_j|a_{j-1}, I, J).p\left(f_j\middle|e_{a_j}\right)\right] \quad (8)$$

With the length model $p(J|I)$, the alignment model $p(i|i',I,J)$ and the lexicon model $p(f_j|e_i)$. The alignment models IBM-1 and IBM-2 are obtained in a similar way by allowing only zero-order dependencies.

IBM-3, 4 and 5 Models. For the generation of the target sentence, it is more appropriate to use the concept of inverted alignments which perform a mapping from a target position *i* to a set of source positions *j*, i.e. we consider mappings $\mathcal{B}$ of the form:

$$\mathcal{B}: i \rightarrow \mathcal{B}_i \subset \{1, \ldots, j, \ldots, J\} \quad (9)$$

with the constraint that each source position j is covered exactly once. Using such an alignment $\mathcal{A} = \mathcal{B}_1^I$ we re-write the probability $Pr(f_1^J, \mathcal{A}|e_1^I)$:

$$Pr(f_1^J, \mathcal{B}_1^I|e_1^I) = p(J|I) \cdot \prod_{i=1}^{I}\left[p(\mathcal{B}_i|\mathcal{B}_1^{i-1}) \cdot \prod_{j \in \mathcal{B}_i} p(f_j|e_i)\right] \quad (10)$$

By making suitable assumptions, in particular first-order dependencies for the inverted alignment model $p(\mathcal{B}_i|\mathcal{B}_1^{i-1})$, we arrive at what is more or less equivalent to the alignment models IBM-3, 4 and 5 [24].

Alignment Template Model. In all the above models, the single words are taken into account. In [25,26], the authors showed significant improvement in translation quality by modeling *word groups* rather than *single words* in both the alignment and lexicon models. This method is known as the *alignment template* (AT) approach.

3.1. Training

The unknown parameters of the alignment and lexicon models are estimated from a corpus of bilingual sentence pairs. The training criterion is the maximum likelihood criterion. As usual, the training algorithms can guarantee only local convergence. In order to mitigate the problems with poor local optima, we apply the following strategy [23]. The training procedure is started with a simple model for which the problem of local optima does not occur or is not critical. The parameters of the simple model are then used to initialize the training procedure of a more complex model, in such a way that a series of models with increasing complexity can be trained. To train the above models except for the alignment template model, we use the GIZA++ software [24]. The alignment template model training scheme, and also the description of our translation system which is based on the alignment template approach is explained in [26].

4. Speech recognition system

The speech recognition system is trained on a large vocabulary, namely the European Parliament Plenary Sessions (EPPS) corpus. The corpus consists of: 67k training-sentences (87.5h) from 154 speakers. The other statistics of the speech recognition train corpus are shown in Table 1.

4.1. Experimental results

We rescore the ASR *N*-best lists with the standard HMM [27] and IBM [23] MT models. Then we use each the *N*-best list as *N*-best hypotheses in order to provide target suffixes for the CAT system.

Table 1: Statistics of the speech recognition train corpus.

	EPPS
Language	English
Acoustic data [h]	87.5
# Running words	705 K
Vocabulary size	58 K
# Segments	67 K
# Speaker	154

The size of the development and evaluation sets *N*-best lists is sufficiently large to achieve almost the best possible results. On average 1738 hypotheses per each source sentence are extracted from the ASR word graphs. The ASR and MT integration experiments are carried out on a large vocabulary task which is the Spanish–English parliamentary speech translation (EPPS). The corpus statistics is shown in Table 2.
To determine the performance of the speech-enabled interactive-predictive CAT system, we simulate a human translator who uses this system. The simulated human knows the correct translation and selects all or part of a suggested suffix whenever this suffix matches fully or partially with the correct translation. If suggested suffix doesn't match with the reference translation, simulated human will more complete the prefix, character by character, until whole or part of a suggested suffix matches with the reference translation.
See Figure 2 for the pseudo-code of the algorithm that simulates a human, matches prefix in the N-best lists and calculates the measure of user efforts.

Table 2: Statistics of the Spanish-English (EPPS) corpus.

		EPPS	
		Spanish	English
Train	Sentences	1 167 627	
	Running words	35.3 M	33.9 M
	Vocabulary size	159 080	110 636
	Singletons	63 045	46 121
Dev	Sentences	1 750	
	Running words	22 174	23 429
	OOVs	64	83
Test	Sentences	792	
	Running words	19 081	19 306
	OOVs	43	45

4.2. Evaluation metrics

In order to measure the performance of our CAT system, we need to determine quantity of effort the human translator for translating a sentence in the absence and presence of the CAT system. For this purpose, we use the summation of the keystroke ratio (KSR) and mouse-action ratio (MAR) which in follow are described.
KSR (Key-stroke ratio): The KSR is the number of key-strokes required to produce the single reference translation using the interactive machine translation system divided by the number of keystrokes needed to type the reference translation. Hence, the KSR is inversely related to the productivity increase which the system brings for the user.

```
Input: N_best_lists, Ref_Sentences, KSR=0, MAR=0
Output: KSMR
1: main()
2: {
3:    for (i=0; i< N_best_lists.size(); i++)
4:         Simulated_User (N_best_lists[i][0],i)
5:    KSMR=(KSR+MAR)/total_character*100
6: }

7: Simulated_User (char* Trans_offer ,int Id)
8: {
9:     Prefix=Find_biggest_prefix(Trans_offer
                                    , Ref_Sentences[Id])
10:    // Find_biggest_prefix compare two char*
11:    // and return the biggest identical substring
12:    if (Prefix== Ref_Sentences[Id])
13:    {
14:       KSR=KSR+1 // for accepting offer
15:       return ;
16:    }
17:    else
18:    {
19:       MAR=MAR+1 // for determining prefix by mouse
20:       Prefix= Prefix +Ref_Sentences[Id][ Prefix.size()]
21:       // the first non_match character is added to prefix.
22:       KSR=KSR+1 // for insert a character
23:       Simulated_User (Match_Prefix (Prefix,Id),Id)
24:    }
25: }

26: char* Match_Prefix(char* Prefix, int Id)
27: {
28:    min=1000
29:    index_min=-1
30:    for (i=0; i< N_best_lists[Id].size(); i++)
31:    {
32:       dis=Minimum_Edit_Distance(N_best_lists[Id][i]
                                    , Prefix)
33:    // Minimum_Edit_Distance is calculated by Levenshtein  Algorithm.
34:        if (dis<min )
35:       {
36:           min=dis
37:           index_min=i
38:       }
39:    }
40:    Suffix= N_best_lists[Id][ index_min] – Prefix
41:    return Suffix
42: }
```

Figure 2: The pseudo-code of the algorithm which simulates a human and matches prefix in the *N*-best list.

A KSR of 1 means that the interactive machine translation has never suggested an appropriate completion to the use sentence prefix, while a KSR value close to 0 means that the system has often suggested perfect completions.

MAR (Mouse-action ratio):

It is similar to KSR, but it measures the number of mouse pointer movements plus one more count per sentence (the user action needed to accept the final translation), divided by the total number of reference characters.

KSMR (Key-stroke and mouse-action ratio):

It is the summation of KSR and MAR, which is the amount of all required actions either by keyboard or by mouse to generate the reference translations using the interactive machine translation system divided by the total number of reference characters.

4.3. Experiments

In order to rescore the *N*-best list generated by the automatic speech recognizer, we make use of the translation models described in Section 3. The rescored *N*-best lists are used in the CAT system as N-best hypotheses lists. After human translator interact with the CAT and a prefix is formed, the CAT will search N-best hypotheses for founding a hypothesis which has minimum edit distance to the prefix and exactly includes the last (partial) word of the prefix. Then the CAT system returns remaining of target sentence to the user (from after last word to end of hypothesis). To study the effect of the N-best list size on the CAT results, we repeat the experiments with N-best lists which have a maximum of 1, 5, 10, 100, 1000 and 5000 hypotheses per sentence for the EPPS task. The results of the speech-enabled interactive-predictive CAT system are listed in Table 3 and 4.

Table 3: KSMR result for Test and Dev in percent. For each translation model, translation probability is calculated in one direction.

			Test	Dev
ASR		n=1	9.2330	12.4844
		n=5	7.8893	10.3986
		n=10	7.3995	9.7566
		n=100	6.3681	8.4446
		n=1000	5.7882	7.9736
		n=5000	5.6361	7.8683
SAR+MT	IBM1	n=1	8.5129	11.751
		n=5	7.1701	9.7380
		n=10	6.7058	9.1292
		n=100	5.7490	7.9496
		n=1000	5.3794	7.5926
		n=5000	5.2884	7.5205
	HMM	n=1	8.9872	12.247
		n=5	7.6180	10.152
		n=10	7.1501	9.5327
		n=100	6.0740	8.2896
		n=1000	5.5724	7.8164
		n=5000	5.4413	7.7057
	IBM3	n=1	8.4091	11.651
		n=5	7.1583	9.6807
		n=10	6.7623	9.0812
		n=100	5.7781	7.9456
		n=1000	5.3858	7.5879
		n=5000	5.3139	7.4903
	IBM4	n=1	8.1488	11.285
		n=5	6.9270	9.3283
		n=10	6.4764	8.7420
		n=100	5.5269	7.7808
		n=1000	5.2319	7.4292
		n=5000	5.1646	7.3556
	IBM5	n=1	7.9867	11.152
		n=5	6.7522	9.2268
		n=10	6.3872	8.7063
		n=100	5.4313	7.6987
		n=1000	5.2082	7.3951
		n=5000	5.1254	7.3308

Table 4: KSMR result for Test and Dev in percent. For each translation model, translation probability is calculated in two directions.

			Test	Dev
SAR+MT	IBM1 & IBM1-I	n=1	7.3686	9.8767
		n=5	6.4200	8.5220
		n=10	6.1487	8.0550
		n=100	5.4286	7.3339
		n=1000	5.1828	7.1325
		n=5000	5.1008	7.0582
	HMM & HMM-I	n=1	7.9385	11.014
		n=5	6.7395	9.2593
		n=10	6.4436	8.6382
		n=100	5.5842	7.6971
		n=1000	5.2702	7.4253
		n=5000	5.2046	7.3564
	IBM3 & IBM3-I	n=1	8.3099	11.248
		n=5	7.1146	9.4592
		n=10	6.6922	8.8450
		n=100	5.7472	7.8304
		n=1000	5.3566	7.4965
		n=5000	5.2884	7.4090
	IBM4 & IBM4-I	n=1	6.6749	9.2780
		n=5	5.8646	8.0410
		n=10	5.6088	7.6445
		n=100	5.0471	7.0489
		n=1000	4.8832	6.8506
		n=5000	4.8450	6.8212
	IBM5 & IBM5-I	n=1	6.7504	9.3662
		n=5	5.8974	8.1115
		n=10	5.6443	7.7606
		n=100	5.0872	7.1194
		n=1000	4.8960	6.8955
		n=5000	4.8678	6.8793

In spite of Table3 that shows the translation probability in one direction ($p(e_1^I|f_1^J)$). Additionally, in Table 4, for each translation model, we calculate the translation probability in both directions: $p(e_1^I|f_1^J)$ and$p(f_1^J|e_1^I)$. Both tables are shown the KSMR measure of the CAT.

4.4. Discussion

As the results show, there is a clear and significant accuracy improvement in all cases when moving from single-best to N-best translations. The best results obtained on the test and development sets are 5.13 % and 7.33 %, respectively. Both of results are produced by the IBM translation Model 5 and the N-best lists with maximum size 5000 hypotheses. According to these results, user of our CAT would only need an effort equivalent to typing about 5.13% and 7.33% of the characters in order to produce the correct translations for the test and development sets, respectively. These results are very ideal for CAT systems.

Also we could improve these results by using the translation models in both directions. These results are shown in Table 4. In this case, the best results obtained on the test and development sets are 4.87% and 6.88%, respectively. For better and easier comparing of the results, consider Figure 3 to Figure 6.

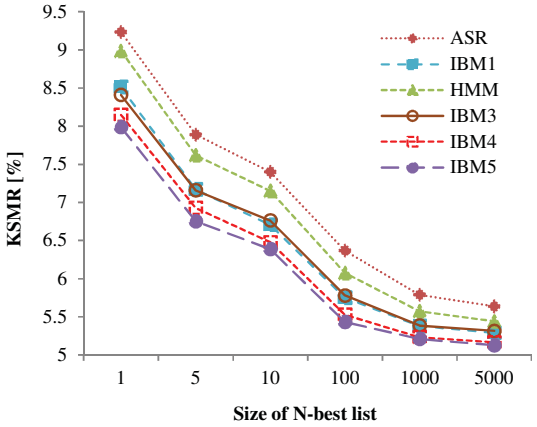


Figure 3: Results of the Interactive-predictive Speech-enabled CAT on the EPPS Test set.

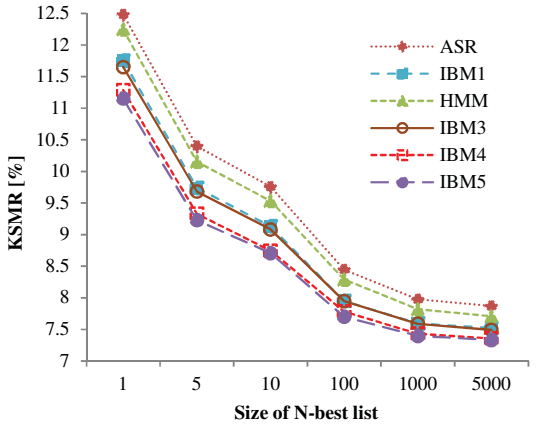


Figure 4: Results of the Interactive-predictive Speech-enabled CAT on the EPPS Dev set.

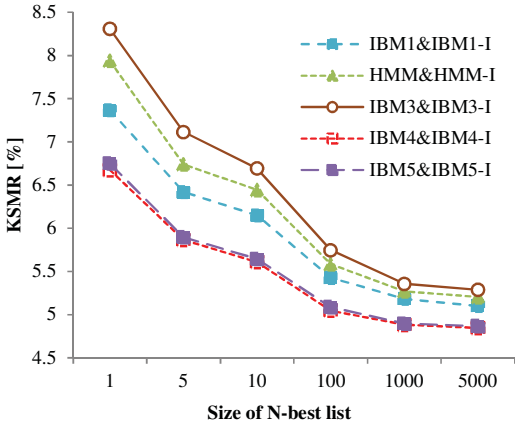


Figure 5: Results of the Interactive-predictive Speech-enabled CAT on the EPPS Test set.

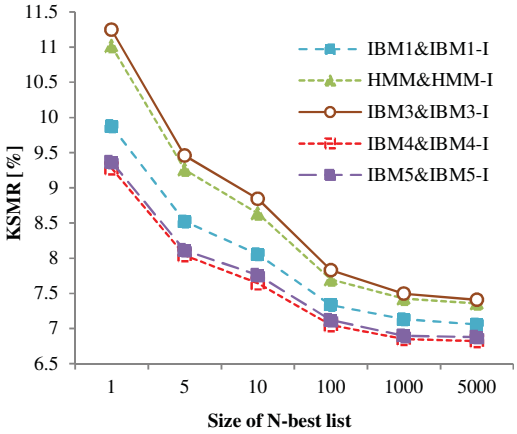


Figure 6: Results of the Interactive-predictive Speech-enabled CAT on the EPPS Dev set.

The successes obtained in these experiments are due to the quality of translations produced by the integrated ASR and MT systems and size of the N-best lists. With larger n-best list, the probability that the CAT system can suggest a better extension will increase.

5. Conclusion

The goal of this paper was to evaluate whether the accuracy of a speech-enabled interactive-predictive CAT system could be improved by using the N-best lists which are obtained by ASR and are rescored by translation models.

We introduced a general framework for integrating the speech recognition and translation models for automatic text dictation in the context of computer-assisted translation. We used the N-best lists which were produced by integrated ASR and MT systems, as N-best hypotheses in the CAT system and we achieved significantly better results.

References

[1]. Kay, M. (1973). *The MIND system,* in Natural Language Processing, pp. 155-188.

[2]. Brown, R.D., Nirenburg, S. (1990). *Human-computer interaction for semantic disambiguation,* In Processing of the International Conference on Computational Linguistics (COLING), PP. 42-47.

[3]. Maruyama, H., Watanabe, H. (1990). *An interactive Japanese parser for machine translation,* In Processing of the International Conference on Computational Linguistics (COLING), pp. 257-262.

[4]. Whitelock, P. J., McGee Wood, M., Chandler, B. J., Holden, N. and Horsfall, H. J. (1986). *Strategies for interactive machine translation: the experience and implications of the UMIST Japanese project*, In Proceedings of the International Conference on Computational Linguistics (COLING), pages 329-334.

[5]. Foster, G., Isabelle, P. and Plamondon, P. (1997). *Target-Text Mediated Interactive Machine translation*, in Kluwer Academic Publishers, pp. 175–194.

[6]. Langlais, P., Foster, G., and Lapalme, G. (2000). *TransType: a computer-aided translation typing system*, In Proceedings of the NAACL/ANLP Workshop on Embedded Machine Translation Systems, pp. 46–52.

[7]. Langlais, P., Lapalme G. and Loranger, M. (2002). *TRANSTYPE: Development–Evaluation Cycles to Boost Translator's Productivity*, in Kluwer Academic Publishers, pp. 77–98.

[8]. Foster, G. (2002). *Text Prediction for Translators,* Ph.D. thesis, Universit´e de Montr´eal, Canada.

[9]. Cubel, E., González, J., Lagarda, A. L., Casacuberta, F., Juan, A. and Vidal, E. (2004). *Adapting finite-state translation to the TransType2 project,* Proceedings of the Joint Conference combining the 8th International Workshop of the European Association for Machine Translation.

[10]. P. F. Brown, S. F. Chen, S. A. D. Pietra, V. D. Pietra, A. S.Kehler, and R. L. Mercer, "Automatic speech recognition in machine-aided translation", *Computer Speech and Language*, vol. 8, no. 3, pp. 177–187,1994.

[11]. M. Dymetman, J. Brousseau, G. Foster, P. Isabelle, Y. Normandin, and P. Plamondon, "Towards an automatic dictation system for translators: the TransTalk project", *in Proceedings of ICSLP-94*, pp. 193–196, 1994.

[12]. J. Brousseau, C. Drouin, G. Foster, P. Isabelle, R. Kuhn, Y. Normandin, and P. Plamondon, "French speech recognition in an automatic dictation system for translators: the transtalk project", *in Proceedings of Eurospeech*, pp. 193–196, 1995.

[13]. S. Khadivi, R. Zens and H. Ney, "Integration of Speech to Computer-Assisted Translation Using Finite-State Automata", *In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 467–474, 2006.

[14]. S. Khadivi and H. Ney. 2, "Integration of Speech Recognition and Machine Translation", *in IEEE Transactions On Audio, Speech, And Language Processing*, VOL. 16, pp. 1551-1564, 2008.

[15]. Reddy, R. Rose and A. Désilets, "Integration of ASR and Machine Translation Models in a Document Translation Task", *In IEEE Transactions on Audio*, Speech, and Language Processing, Canada, 2007.

[16]. M. Paulik and A. Waibel, "Extracting clues from human interpreter speech for spoken language translation", *in Proc. ICASSP*, pp. 5097-5100, 2008.

[17]. E.Vidal, F. Casacuberta, L. Rodríguez, J. Civera, and C. Martínez. Computer-assisted translation using speech recognition. *IEEE Transaction on Audio, Speech and Language Processing*, 14(3):941-951, 2006.

[18]. K. A. Papineni, S. Roukos, and R. T. Ward, "Feature based language understanding, in EUROSPEECH", Rhodes, Greece, September, pp. 1435–1438, 1997.

[19]. K. A. Papineni, S. Roukos, and R. T. Ward, "Maximum likelihood and discriminative training of direct translation models", *in Proc. IEEE Int. Conf. on Acoustics*, Speech, and Signal Processing (ICASSP), vol. 1, Seattle, WA, pp. 189–192, 1998.

[20]. P. Beyerlein, "Discriminative model combination, in Proc. IEEE Int. Conf. on Acoustics", *Speech, and Signal Processing (ICASSP)*, vol. 1, Seattle, WA, pp.481 – 484, 1998.

[21]. F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation", *in Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, pp. 295–302, 2002.

[22]. F. J. Och, "Minimum error rate training in statistical machine translation", *in Proc. of the 41th Annual Meeting*

of the Association for Computational Linguistics (ACL), Sapporo, Japan, pp. 160–167, 2003.

[23]. P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L.Mercer, "The mathematics of statistical machine translation: Parameter estimation", *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[24]. F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models", *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[25]. F. J. Och, C. Tillmann, and H. Ney, "Improved alignment models for statistical machine translation*", in Proc. Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, University of Maryland, College Park, MD, pp. 20-28, 1999.

[26]. F. J. Och and H. Ney, "The alignment template approach to statistical machine translation", *Computational Linguistics*, vol. 30, no. 4, pp. 417–449, 2004.

[27]. Vogel, H. Ney, and C. Tillmann. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics*, pages 836–841, Morristown, NJ, USA, 1996.

MDI Adaptation for the Lazy: Avoiding Normalization in LM Adaptation for Lecture Translation

Nick Ruiz, Marcello Federico

FBK - Fondazione Bruno Kessler
Via Sommarive 18, 38123 Povo (TN), Italy
{nicruiz,federico}@fbk.eu

Abstract

This paper provides a fast alternative to Minimum Discrimination Information-based language model adaptation for statistical machine translation. We provide an alternative to computing a normalization term that requires computing full model probabilities (including back-off probabilities) for all n-grams. Rather than re-estimating an entire language model, our Lazy MDI approach leverages a smoothed unigram ratio between an adaptation text and the background language model to scale only the n-gram probabilities corresponding to translation options gathered by the SMT decoder. The effects of the unigram ratio are scaled by adding an additional feature weight to the log-linear discriminative model. We present results on the IWSLT 2012 TED talk translation task and show that Lazy MDI provides comparable language model adaptation performance to classic MDI.

1. Introduction

Topic adaptation is used as a technique to adapt language models based on small contexts of information that may not necessarily reflect an entire domain or genre. In scenarios such as lecture translation, it is advantageous to perform language model adaptation on the fly to reflect topical changes in a discourse. In these scenarios, general purpose domain adaptation techniques fail to capture the nuances of discourse; while domain adaptation works well in modeling newspapers and government texts which contain a limited number of subtopics, the genres of lectures and speech may cover a virtually unbounded number of topics that change over time. Instead of general purpose adaptation, adaptation should be performed on smaller windows of context.

Most domain adaptation techniques require the re-estimation of an entire language model to leverage the use of out-of-domain corpora in the construction of robust models. While efficient algorithms exist for domain adaptation, they are in practice intended to adapt language models globally over a new translation task. Topic adaptation, on the other hand, intends to adapt language models as relevant contextual information becomes available. For a speech, the relevant contextual information may come in sub-minute intervals. Well-established and efficient techniques such as Minimum Discrimination Information adaptation [1, 2] are unable to perform topic adaptation in real-time scenarios for large order n-gram language models. In practice, new contextual information is likely to be available before techniques such as MDI have finished LM adaptation from earlier contexts. Thus spoken language translation systems are typically unable to use the state-of-the-art techniques for the purpose of topic adaptation.

In this paper, we seek to apply MDI adaptation techniques in real-time translation scenarios by avoiding the computation of the normalization term that requires all n-grams to be re-estimated. Instead, we only wish to adapt n-grams that appear within an adaptation context. Dubbed "Lazy MDI", our technique uses the same unigram ratios as MDI, but avoids normalization by applying smoothing transformations based a sigmoid function that is added as a new feature to the conventional log-linear model of phrase-based statistical machine translation (SMT). We observe that Lazy MDI performs comparably to classic MDI in topic adaptation for SMT, but possesses the desired scalability features for real-time adaptation of large-order n-gram LMs.

This paper is organized as follows: In Section 2, we discuss relevant previous work. In Section 3, we review MDI adaptation. In Section 4, we describe Lazy MDI adaptation for machine translation and review how unigram statistics of adaptation texts can be derived using bilingual topic modeling. In Section 5, we report adaptation experiments on TED talks[1] from IWSLT 2010 and 2012, followed by our conclusions and suggestions for future work in Section 6.

2. Previous Work

This paper is based on the work of [3], which combines MDI adaptation with bilingual topic modeling on small adaptation contexts for lecture translation. Adaptation texts are drawn from source language input and leveraged for language model adaptation. A bilingual Probabilistic Latent Semantic Analysis (PLSA) [4] model is constructed by combining parallel training texts, allowing for inference on monolingual source texts for MDI adaptation by removing source language unigram statistics.

[1] http://www.ted.com/talks

A similar approach is considered by [5] in domain adaptation by constructing two hierarchical LDA models from parallel document corpora and enforcing a one-to-one correspondence between the models by learning the hyperparameters of the variational Dirichlet posteriors in one LDA model and bootstrapping the second model by fixing the hyperparameters. The bilingual LSA framework is also applied to adapt translation models. Other bilingual topic modeling approaches include Hidden Markov Bilingual Topic AdMixtures [6] and Polylingual Topic Models [7].

The literature focuses primarily on domain adaptation, using techniques such as information retrieval to select similar sentences in training corpora for adaptation, either through interpolation [8] or corpora filtering [9], or mixture model adaptation approaches [10, 11].

An alternative to MDI adaptation is proposed by [12], which uses a log-linear combination of binary features $f_i(h, w)$ to scale LM probabilities $P(w \mid h)$:

$$\hat{P}(w \mid h) = \exp\left(\sum_i f_i(h, w)\lambda_i\right) P(w \mid h).$$

Normalization is avoided by simply dividing $\hat{P}(w \mid h)$ by $\hat{P}(w \mid h) + 1$.

3. MDI Adaptation

MDI adaptation was originally presented in [1] as a means for domain adaptation on language models. MDI adaptation scales the probabilities of a background language model, $P_B(h, w)$, by a factor determined by a ratio between the unigram statistics observed in an adaptation text A versus the same statistics observed in the background corpus B:

$$\alpha(w) = \left(\frac{\hat{P}_A(w)}{P_B(w)}\right)^\gamma, \quad 0 < \gamma \leq 1. \tag{1}$$

As such, the adapted language model $P_A(h, w)$ is constructed as follows:

$$P_A(h, w) = P_B(h, w)\alpha(w), \tag{2}$$

where h is the n-gram history of word w. As outlined in [13], the adapted language model can also be written recursively in an interpolated conditional form with discounted frequencies $f^*(w|h)$ and reserved probabilities for out-of-vocabulary words $\lambda(h)$:

$$P_A(w|h) = f_A^*(w|h) + \lambda_A(h)P_A(w|h'), \tag{3}$$

with:

$$f_A^*(w|h) = \frac{f_B^*(w|h)\alpha(w)}{z(h)}, \tag{4}$$

$$\lambda_A(h) = \frac{\lambda_B(h)z(h')}{z(h)}, \tag{5}$$

and

$$z(h) = \Big(\sum_{w:N_B(h,w)>0} f_B^*(w|h)\alpha(w)\Big) + \lambda_B(h)z(h'), \tag{6}$$

which efficiently computes the normalization term for high order n-grams recursively by just summing over observed n-grams. The recursion ends with the following initial values for the empty history ϵ:

$$z(\epsilon) = \sum_w P_B(w)\alpha(w), \tag{7}$$

$$P_A(w|\epsilon) = P_B(w)\alpha(w)z(\epsilon)^{-1}. \tag{8}$$

While MDI has been applied in domain adaptation both for language models [2] and translation models [5], its re-estimation requires the computation of the normalization term outlined in (6). In topic adaptation scenarios, it is desirable to rapidly adapt a background language model using small adaptation contexts consisting of few sentences. One method of inferring unigram statistics for MDI adaptation given sparse data is to perform bilingual topic modeling [3, 5, 7]. While it has been shown that the combination of topic modeling and MDI adaptation yield a significant improvement in translation adequacy, the approach of adapting non-overlapping contexts of size C requires M/C full LM re-estimations on a translation task with M sentences, with each re-estimation requiring the expensive computation of the normalization term.

4. Lazy MDI Alternative for SMT

The goal of MDI adaptation is to construct an adapted language model that minimizes its Kullback-Leibler divergence from the background LM, which is effectively performed via the unigram ratio scaling method described in (1) and (2). We seek to loosely approximate this KL divergence in statistical machine translation by adapting only n-grams that appear as translation options for a given sentence. As such, we seek to avoid computing a normalization term that requires observing the probabilities of all high- and lower-order n-grams in the LM. Since the ratio of unigram probabilities is defined across the range $[0, +\infty]$, we explore smoothing functions that bind the ratio to a finite range.

4.1. Smoothing unigram ratios

In machine learning, sigmoid activation functions are typically used to constrain functions in the range of $[0, a]$ or $[-a, a]$ to reduce the bias of a few data points within a training set. Likewise we explore the use of sigmoid functions to reward n-gram probabilities across the range of $[0, a]$. However, since we are scaling ratios in general, we desire the following properties of our smoothing function f:

$$f(0) = 0; \quad \lim_{x\to+\infty} f(x) = a$$

$$f(1) = 1; \quad \lim_{x\to-\infty} f(x) = -a$$

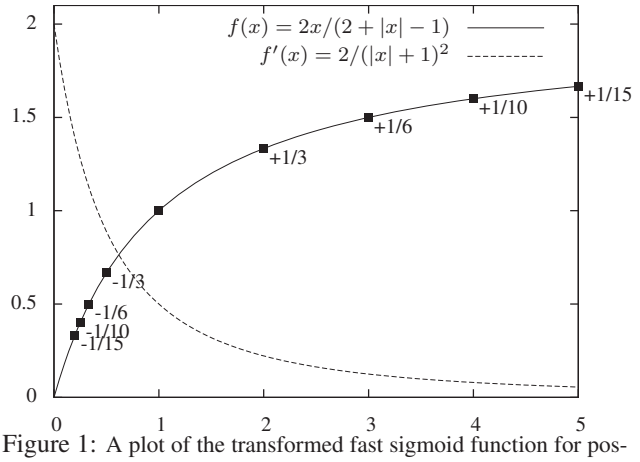


Figure 1: A plot of the transformed fast sigmoid function for positive ratios in (10) and its first derivative, evaluated at $a = 2$. The relative changes in $f(x)$ are labeled, centered at $f(1)$. The changes in $f(x)$ are symmetric with respect to each ratio and inverse ratio.

In particular the $f(1) = 1$ constraint ensures that background LM probabilities remain fixed when the ratio is balanced.

Staple sigmoid functions such as the logistic function or the hyperbolic tangent unfortunately cannot satisfy the property $f(1) = 1$ for any magnitude a. However, a *fast sigmoid* approximation was proposed in [14], defined as:

$$f(x) = \frac{x}{1 + |x|}. \tag{9}$$

With some simple transformations, we arrive at our desired function:

$$f(x, a) = \frac{ax}{a + |x| - 1}, \quad a > 1. \tag{10}$$

Figure 1 contains a plot of (10) at $a = 2$ and its first derivative. A useful property of the fast sigmoid in (10) is that the change in slope is symmetric with respect to inverted ratios, relative to the center at $x = 1$. For example, for the fast sigmoid outlined in Figure 1, a ratio of 2:1 yields a scale of $1 + \frac{1}{3}$, while a ratio of 1:2 yields a scale of $1 - \frac{1}{3}$.

4.2. Log-linear feature

Since we are no longer normalizing n-gram probabilities, we can consider the smoothed unigram probabilities as a function that rewards or penalizes translation options based on the likelihood that the words composing the target phrase should appear in the translation. We treat the smoothed unigram probabilities as a new feature in the discriminative log-linear model of the decoder. While our new feature is independent from any language model features, we can logically consider the adaptation of a background language model as a log-linear combination of the LM feature and the Lazy MDI feature as:

$$\hat{P}_{LM}(E \mid F) = P_{LM}(E \mid F)^{\gamma_1} \cdot \prod_{i=1}^{|E|} \hat{\alpha}(e_i)^{\gamma_2}, \tag{11}$$

where $P_{LM}(E \mid F)$ computes the language model probabilities of target sentence E, given a source sentence F; though we only consider language models that score the target sentence, independent from F. $\hat{\alpha}(e_i)$ is the Lazy MDI adaptation on the ith target word in E, defined as:

$$\hat{\alpha}(w) = f\left(\frac{P_A(w)}{P_B(w)}\right). \tag{12}$$

By rearranging terms, we arrive at our unnormalized log-linear approximation of (2):

$$\hat{P}_{LM}(E) = \prod_{i=1}^{|E|} P_{LM}(e_i \mid h_i)^{\gamma_1} \cdot \hat{\alpha}(e_i)^{\gamma_2}. \tag{13}$$

In practice, only translation hypotheses suggested by the translation model are scored by the language model, thus limiting the number of unigram ratios to consider. Additionally, for computational efficiency, calculations are performed in log space. For $a = 2$, our fast sigmoid function can be rewritten as:

$$f(x, 2) = 2 \cdot \left(1 + e^{-\ln(x)}\right)^{-1}, \quad x > 0, \tag{14}$$

which allows us to compute log probability ratios as $\ln P_A(w) - \ln P_B(w)$.

4.3. Sparsity considerations

If we treat the background and adaptation unigram statistics as unigram language models, we can use smoothing to reserve probability for out-of-vocabulary words. However, due to the sparsity of unigram features in adaptation texts, it is possible that the adapted unigram statistics are missing words that appear in the background LM. Assuming that there are insufficient adaptation statistics to reliably scale the probabilities of n-grams containing these words, we instead leave the background probabilities intact by fixing the unigram probability ratio to 1.

A similar problem can arise in the scenario that the adaptation text contains unigrams that are not observed in the background LM. One possible solution is to limit the vocabulary of the adaptation statistics to the same as that of the background.

4.4. Inferring unigrams via bilingual topic modeling

Since an adaptation text is in practice too small to directly compute reliable unigram statistics, we resort to topic modeling approaches to infer full unigram probabilities. One such approach is Probabilistic Latent Semantic Analysis (PLSA) [4], which computes the probability of unigrams in a document d by marginalizing over a collection of latent topics Z:

$$P(w \mid d) = \sum_{z \in Z} P(w \mid z) P(z \mid d). \tag{15}$$

Following the exposition of [3], we construct a bilingual topic model by combining source and target parallel

sentences into "monolingual" documents with vocabulary $V_{FE} = V_F \cup V_E$.[2] During inference, we infer unigram probabilities of V_{FE} using only documents containing only the source language. Removing words $f \in V_F$ from the probability distribution and normalizing yields a probability distribution for all words in V_E.

5. Experiments

We conduct experiments on the IWSLT TED talk translation tasks from 2010 and 2012. In Section 5.1, we evaluate the utility of Lazy MDI using lowercased unigram statistics on a lowercased MT system trained only on TED data. We compare the performance of smoothed and unsmoothed Lazy MDI against classic MDI.

In Section 5.2, we evaluate the logical adaptation of cased language models with uncased unigram statistics from both the adaptation text and the background text. Due to the small size of the adaptation texts, we are not guaranteed a reliable unigram probability estimations on a vocabulary that is likely to double in size. We evaluate the utility of Lazy MDI on a state-of-the-art system against a domain-adapted mixture LM.

5.1. IWSLT 2010

We replicate the experimental settings of [3] and provide a comparison of classic MDI against Lazy MDI, using the same data set of English-French translations of TED talks, downloaded from the TED website as it was on March 30, 2011 and split into training, dev and test sets according to indexes used for IWSLT 2010[3] evaluation. The data set is segmented at the clause level, rather than at the level of sentences. The TED training data consists of 329 parallel talk transcripts with approximately 84k sentences. The TED test data consists of transcriptions created via 1-best ASR outputs from the KIT Quaero Evaluation System. It consists of 2381 clauses and approximately 25,000 English and French words, respectively.

Lowercased SMT systems are built upon the Moses open-source SMT toolkit [15][4]. The translation and lexicalized reordering models have been trained on parallel data. One 5-gram background LM was constructed with the IRSTLM toolkit [16] on the French side of the TED training data (740k words), and smoothed via the improved Kneser-Ney technique [17]. The weights of the log-linear interpolation model were optimized via minimum error rate training (MERT) [18] on the TED development set, using 200 best translations at each tuning iteration.

As in [3], online adaptation is simulated by splitting the training corpus into small non-overlapping contexts of 5 lines (41,847 "documents" in total) and performing bilingual PLSA training using IRSTLM. The PLSA model consists of 250 topics and is trained for 20 EM iterations. Ten inference iterations are performed on the English side of the development and test sets to generate French unigram probabilities for each 5-line context.

MDI adaptation is performed on the test set contexts using the 5-gram TED language model described above as the background. For each 5-line context in the test set, the background LM is replaced with the adapted LM for SMT decoding, preserving the same feature weight as the background LM.

In the case of Lazy MDI, adaptation is integrated into the Moses decoder using the same context unigrams. MERT is performed on the development set with simultaneous adaptation for each context. We experiment with both adaptation via unsmoothed unigram ratios and smoothing via our transformed fast sigmoid function. Words not in the adaptation unigram LM are fixed with a 1:1 ratio to prevent their effect on the global translation hypothesis score.

We ran 3 MERT instances for each system and evaluated using MultiEval 0.3 [19]. Evaluation results in terms of BLEU, METEOR (French), TER, and segment length are listed in Table 1. We observe similar results between MDI and smoothed Lazy MDI – both of which yield an average improvement of 0.2 and 0.3 BLEU, respectively. As predicted, unsmoothed Lazy MDI adaptation performs poorly as the unigram ratios between the background and context LMs often diverge greatly. This can also be observed in the weight associated with the feature, as shown in Table 2. For unsmoothed Lazy MDI, the associated feature weight has divergent values across each MERT instance, implying the un-

Metric	System	Avg	$\bar{s}_{\text{sel}}$	s_{Test}	p
BLEU ↑	Baseline	28.0	0.5	0.3	-
	MDI	28.2	0.5	0.2	0.01
	Lazy MDI (unsmoothed)	24.4	0.5	5.8	0.00
	Lazy MDI (smoothed)	28.3	0.5	0.1	0.00
METEOR ↑	Baseline	50.4	0.4	0.1	-
	MDI	50.6	0.5	0.2	0.09
	Lazy MDI (unsmoothed)	47.7	0.4	4.3	0.00
	Lazy MDI (smoothed)	50.5	0.4	0.1	0.18
TER ↓	Baseline	57.3	0.6	0.4	-
	MDI	56.9	0.6	0.4	0.00
	Lazy MDI (unsmoothed)	61.9	0.6	8.0	0.00
	Lazy MDI (smoothed)	56.9	0.6	0.1	0.00
Length	Baseline	104.1	0.5	1.1	-
	MDI	103.5	0.5	0.9	0.00
	Lazy MDI (unsmoothed)	106.2	0.5	4.5	0.00
	Lazy MDI (smoothed)	103.5	0.5	0.2	0.00

Table 1: Lowercased evaluation of MDI and Lazy MDI adaptation techniques on the IWSLT 2010 TED test set. Metric scores averaged across three MERT runs. p-values are relative to the baseline. s_{sel} indicates the variance due to test set selection. Significant improvements in terms of BLEU and TER are observed for both MDI and smoothed Lazy MDI (via a fast sigmoid transformation of unigram ratios). Unsmoothed Lazy MDI yields unpredictable results during optimization.

[2] To avoid overlapping types in the topic mdoel, we annotate the source and target vocabularies to track their provenance.

[3] http://iwslt2010.fbk.eu/

[4] http://www.statmt.org/moses/

predictability of unbounded ratios.

System	Metric	Opt 1	Opt 2	Opt 3
Baseline	BLEU	27.64	28.20	28.20
MDI	BLEU	28.49	28.07	28.16
Lazy MDI (unsmoothed)	BLEU	27.14	*17.80*	28.40
	weight	0.1537	*0.4096*	0.0445
Lazy MDI (smoothed)	BLEU	28.27	28.39	28.17
	weight	0.0132	0.0177	0.0138

Table 2: Lowercased evaluation runs for the TED baseline and Lazy MDI adaptations for the IWSLT 2010 test set across three tuning instances. Unsmoothed Lazy MDI yields unstable adaptation feature weights across each run. "Opt 2" overpowers the log-linear model, causing a large overfitting to the development set. "Opt 3" provides the best generalization to the test set by reducing the effects of the adaptation. For fast sigmoid-smoothed Lazy MDI, the adaptation weights remain consistent across all runs.

5.2. IWSLT 2012

We also evaluate the performance of our fast sigmoid-smoothed Lazy MDI setting on a state-of-the-art SMT system submitted for the IWSLT 2012 TED English-French MT shared task[5]. In this experiment, we build cased translation systems using Moses and evaluate the effects of Lazy MDI adaptation from lowercased unigram context statistics. Our baseline system consists of translation and reordering models trained from the in-domain TED[6] corpus, as well as out-of-domain Giga French-English[7] and Europarl v7 [21] corpora. Each out-of-domain corpus was domain-adapted by aggressive filtering using a cross-entropy difference scoring technique described by [22] on the French side and optimizing the perplexity against the (French) TED training data by incrementally adding sentences. The corresponding parallel English sentences were preserved to provide compact parallel corpora. A single phrase and reordering table were constructed using the fill-up technique described in [23] in a cascaded fashion in the order of TED, Giga French-English, and Europarl.

A domain-adapted 5-gram mixture language model was constructed with IRSTLM from the TED, Giga French-English, Gigaword French v2 AFP[8], and WMT News Commentary v7 corpora. The same filtering technique [22] was applied to the LM corpora. For Lazy MDI, we again use the bilingual PLSA model constructed from the IWSLT 2010 training data, with 250 topics and 20 EM iterations. MERT is again performed on the development set with simultaneous Lazy MDI adaptation for each context.

Topic adaptation results against the domain-adapted baseline are shown in Table 3. The evaluation results are averaged over three MERT optimizations of the baseline and Lazy MDI-adapted systems. We observe that performing Lazy MDI adaptation yields a BLEU improvement of 0.2 against the already-adapted baseline, suggesting a cumulative gain of domain adaptation and topic adaptation. We also observe a 0.2 improvement in terms of TER, while METEOR remains more or less the same. The tuning weights obtained across three MERT iterations are averaged to control optimizer instability. We list the evaluation results of each system run in Table 4.

Metric	System	Avg	$\bar{s}_{\text{sel}}$	s_{Test}	p
BLEU ↑	Mix LM	32.4	0.5	0.0	-
	+Lazy MDI	32.6	0.5	0.1	0.07
METEOR ↑	Mix LM	52.0	0.4	0.0	-
	+Lazy MDI	52.1	0.4	0.1	0.18
TER ↓	Mix LM	49.5	0.5	0.1	-
	+Lazy MDI	49.3	0.5	0.2	0.05
Length	Mix LM	97.3	0.4	0.3	-
	+Lazy MDI	97.2	0.4	0.2	0.12

Table 3: Evaluation of Lazy MDI adaptation on the IWSLT 2010 TED test set provided in the IWSLT 2012 TED translation task. Metric scores averaged across three MERT runs. Lazy MDI p-values are relative to the domain-adapted baseline, described in Section 5.2. s_{sel} indicates the variance due to test set selection. Significant improvements in terms of BLEU and TER are observed for smoothed Lazy MDI (via a fast sigmoid transformation of unigram ratios).

System	Metric	Opt 1	Opt 2	Opt 3	Avg
Mix LM	BLEU	32.37	32.44	32.44	32.42
	NIST	7.463	7.438	7.438	7.443
+Lazy MDI	BLEU	32.63	32.55	32.52	32.70
	NIST	7.473	7.480	7.440	7.448

Table 4: Lowercased evaluation runs for the mixture LM baseline and Lazy MDI adaptations for the 2010 test set in the IWSLT 2012 translation task, across three tuning instances. The weights from the tuning instances are averaged to control optimizer instability. Performing Lazy MDI adaptation on the mixture LM baseline yields a 0.28 BLEU improvement and marginal NIST improvements.

We evaluate the impact of Lazy MDI adaptation by computing TER on the translation of each individual line from the 2010 test set by each system. We observe that of the 1,664 transcript lines, 247 lines yield a TER improvement, while 175 result in a higher error rate. We show three examples of segments yielding a TER improvement in Table 5. For ID #364, Lazy MDI yields a slight increase in fluency, while adequacy remains more or less the same. The baseline suggests that white pills are worse than blue pills – a subtle difference from the intent of the reference. The Lazy-adapted hypothesis corrects this difference, but makes common mistakes in translating "good" and "as". Lazy MDI yields a shorter translation in ID #1055 that moves away from a literal translation in the first half of the sentence that closely matches the reference. ID #1059 results in a very minor article change from

[5]http://hltc.cs.ust.hk/iwslt/index.php/evaluation-campaign/ted-task

[6]https://wit3.fbk.eu/mt.php?release=2012-03-test

[7]10^9 French-English data set provided by the WMT 2012 translation task [20].

[8]http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2009T28

"the" to "our". In this context, this subtle difference is important because the speaker is comparing the water at his fish farm to other farms.

ID	Text	TER
364	But a white pill is not as good as a blue pill .	
	Mais un comprimé blanc n' est pas aussi bon qu' une comprimé bleu	(0.154)
	Mais une pilule blanche est moins bonne qu' une pilule bleue .	0.769
	Mais une pilule blanche n' est pas aussi bien comme une pilule bleue .	0.615
1055	I mentioned that to Miguel , and he nodded .	
	J' ai dit ça à Miguel , et il a acquiesçé .	(0.167)
	J' ai mentionné que de Miguel , et il a fait un signe .	0.500
	J' ai dit à Miguel , et il a fait un signe .	0.333
1059	And then he added , " But our water has no impurities . "	
	Et puis il a ajouté : " Mais notre eau n' a pas d' impuretés . "	(0.058)
	Et puis il a ajouté : " Mais l' eau n' a pas impuretés . "	0.176
	Et puis il a ajouté : " Mais notre eau n' a pas impuretés . "	0.118

Table 5: Three examples of improvement in MT results: the first translation in each collection corresponds to the reference translation, the second utilizes a mixture LM, and the third adds Lazy MDI adaptation. The sentence-level TER scores are listed by each hypothesis and the difference is listed in parentheses by the reference.

We also outline three examples of diminished performance after performing Lazy MDI in Table 6. The Lazy MDI example in ID #858 demonstrates an attempt to literally translate the word "space" as "espace", which can ambiguously refer either to outer space, or a domain (as in the reference translation). This surface word is likely to have been chosen above "domaine" due to its topic similarity to "nucléaire". While the TER on this sentence is higher than the baseline, it should be noted that the baseline didn't provide a translation for "space". ID #895 is an example where the topic adaptation attempts to literally translate "I think", but adds an additional "that" afterward. The sentence becomes a bit awkward to read. The baseline, however, leaves out the hedge phrase "I think" and comes across as factual. It is likely that a human translator would prefer the topic-adapted sentence. In ID #1358, synonyms for "globe" are selected, correctly implying that the speaker refers to a globe as the world. While the reference and baseline select the word "planet", the topic-adapted sentence prefers "world" – an equally acceptable word. It is likely that "world" was selected due to collocations with "trash" and "pollution". With only one reference translation, it is hard to detect when Lazy MDI adaptation actually worsens the translation hypothesis.

6. Conclusions

We have presented a simplified framework for approximating MDI adaptation in an online manner for lecture translation. We avoid normalization computations that prevent

ID	Text	TER
858	In the nuclear space , there are other innovators .	
	Dans le domaine nucléaire , il y a d' autres innovateurs .	(-0.167)
	Dans le nucléaire , il y a d' autres innovateurs .	0.083
	En l' espace nucléaire , il y a d' autres innovateurs .	0.250
895	And so there is a thread of something that I think is appropriate .	
	Mais là-dedans , il y a quelque chose qui ne me semble pas faux .	(-0.267)
	Et il y a un fil de quelque chose qui est approprié .	0.600
	Et il y a un fil de quelque chose que je pense que c' est approprié .	0.867
1358	and not only that , we 've used our imagination to thoroughly trash this globe .	
	Pire , nous avons utilisé notre imagination pour polluer profondément cette planète .	(-0.154)
	Et non seulement ça , nous avons utilisé notre imagination à ordures soigneusement cette planète .	0.538
	Et non seulement ça , nous avons utilisé notre imagination à ordures soigneusement ce monde .	0.692

Table 6: Three examples of decreased TER performance in MT results: the first translation in each collection corresponds to the reference translation, the second utilizes a mixture LM, and the third adds Lazy MDI adaptation. The sentence-level TER scores are listed by each hypothesis and the difference is listed in parentheses by the reference.

classic MDI from being used in speech translation scenarios. Lazy MDI adaptation acts as a separate log-linear feature that doesn't directly adapt LM probabilities – instead, it rewards or penalizes the scores of each translation hypothesis by observing the unigram probabilities inferred an adaptation context and compares it to the background in a smoothed ratio. The smoothing is performed by a conservative fast sigmoid function that favors 1:1 ratios and prevents ratios from growing above a magnitude a.

We conducted adaptation experiments on TED talk data from IWSLT 2010 and 2012 and demonstrate a significant improvement in terms of BLEU, NIST, and TER over two baselines: a lowecased TED-only system, and a state-of-the-art cased system that combines in-domain and out-of-domain data. We demonstrate that Lazy MDI adaptation has cumulative adaptation effects on already-adapted language models.

For future work, we intend to compare our fast sigmoid function against non-sigmoidal smoothing functions for Lazy MDI. We additionally intend to explore log-linear alternatives that do not rely on the computation of unigram ratios – for example, inferring context from semantically-rich resources, such as Wikipedia or WordNet.

As it currently stands, Lazy MDI adaptation scales unigram ratios from data sources with differing vocabularies. It is likely that we can gain more reliable ratios by filtering the background unigram LM vocabulary to match the adaptation text and renormalizing the probabilities.

Another potential weakness in our approach is the use of topic models that do not filter stop-words and perform unigram adaptation on the surface level. For morphologically-

rich languages, such as German or Arabic, the vocabulary sizes can increase greatly due to word splitting. We intend to test our adaptation approach using word stems.

7. Acknowledgements

This work was supported by the T4ME network of excellence (IST-249119), funded by the DG INFSO of the European Commission through the Seventh Framework Programme.

8. References

[1] S. A. Della Pietra, V. J. Della Pietra, R. Mercer, and S. Roukos, "Adaptive language model estimation using minimum discrimination estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. I, San Francisco, CA, 1992, pp. 633–636.

[2] M. Federico, "Efficient language model adaptation through MDI estimation," in *Proceedings of the 6th European Conference on Speech Communication and Technology*, vol. 4, Budapest, Hungary, 1999, pp. 1583–1586.

[3] N. Ruiz and M. Federico, "Topic adaptation for lecture translation through bilingual latent semantic models," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, July 2011, pp. 294–302. [Online]. Available: http://www.aclweb.org/anthology/W11-2133

[4] T. Hofmann, "Probabilistic Latent Semantic Analysis," in *Proceedings of the 15th Conference on Uncertainty in AI*, Stockholm, Sweden, 1999, pp. 289–296.

[5] Y.-C. Tam, I. Lane, and T. Schultz, "Bilingual LSA-based adaptation for statistical machine translation," *Machine Translation*, vol. 21, pp. 187–207, December 2007. [Online]. Available: http://portal.acm.org/citation.cfm?id=1466799.1466803

[6] B. Zhao and E. P. Xing, "HM-BiTAM: Bilingual Topic Exploration, Word Alignment, and Translation," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008, pp. 1689–1696.

[7] D. Mimno, H. M. Wallach, J. Naradowsky, D. A. Smith, and A. McCallum, "Polylingual Topic Models," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, August 2009.

[8] B. Zhao, M. Eck, and S. Vogel, "Language Model Adaptation for Statistical Machine Translation via Structured Query Models," in *Proceedings of Coling 2004*. Geneva, Switzerland: COLING, Aug 23–Aug 27 2004, pp. 411–417.

[9] A. Sethy, P. Georgiou, and S. Narayanan, "Selecting relevant text subsets from web-data for building topic specific language models," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. New York City, USA: Association for Computational Linguistics, June 2006, pp. 145–148. [Online]. Available: http://www.aclweb.org/anthology/N/N06/N06-2037

[10] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," in *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 128–135. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0217

[11] P. Koehn and J. Schroeder, "Experiments in Domain Adaptation for Statistical Machine Translation," in *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 224–227. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0233

[12] S. F. Chen, K. Seymore, and R. Rosenfeld, "Topic adaptation for language modeling using unnormalized exponential models," in *IEEE ICASSP-98*. IEEE, 1998, pp. 681–684.

[13] M. Federico, "Language Model Adaptation through Topic Decomposition and MDI Estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. I, Orlando, FL, 2002, pp. 703–706.

[14] G. M. Georgiou, "Parallel distributed processing in the complex domain," Ph.D. dissertation, Tulane University, New Orleans, LA, USA, 1992, uMI Order No. GAX92-29796.

[15] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, Prague, Czech Republic, 2007, pp. 177–180. [Online]. Available: http://aclweb.org/anthology-new/P/P07/P07-2045.pdf

[16] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an Open Source Toolkit for Handling Large Scale Language Models," in *Proceedings of Interspeech*, Melbourne, Australia, 2008, pp. 1618–1621.

[17] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” *Computer Speech and Language*, vol. 4, no. 13, pp. 359–393, 1999.

[18] F. J. Och, “Minimum Error Rate Training in Statistical Machine Translation,” in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, E. Hinrichs and D. Roth, Eds., 2003, pp. 160–167. [Online]. Available: http://www.aclweb.org/anthology/P03-1021.pdf

[19] J. Clark, C. Dyer, A. Lavie, and N. Smith, “Better hypothesis testing for statistical machine translation: Controlling for optimizer instability,” in *Proceedings of the Association for Computational Lingustics*, ser. ACL 2011. Portland, Oregon, USA: Association for Computational Linguistics, 2011, available at http://www.cs.cmu.edu/ jhclark/pubs/significance.pdf.

[20] C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, “Findings of the 2012 workshop on statistical machine translation,” in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 10–51. [Online]. Available: http://www.aclweb.org/anthology/W12-3102

[21] P. Koehn, “Europarl: A multilingual corpus for evaluation of machine translation,” Unpublished, `http://www.isi.edu/~koehn/europarl/`, 2002.

[22] R. C. Moore and W. Lewis, “Intelligent selection of language model training data,” in *ACL (Short Papers)*, 2010, pp. 220–224.

[23] A. Bisazza, N. Ruiz, and M. Federico, “Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation,” in *International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011.

Segmentation and Punctuation Prediction in Speech Language Translation Using a Monolingual Translation System

Eunah Cho, Jan Niehues and Alex Waibel

International Center for Advanced Communication Technologies - InterACT
Institute of Anthropomatics
Karlsruhe Institute of Technology, Germany
firstname.lastname@kit.edu

Abstract

In spoken language translation (SLT), finding proper segmentation and reconstructing punctuation marks are not only significant but also challenging tasks. In this paper we present our recent work on speech translation quality analysis for German-English by improving sentence segmentation and punctuation.

From oracle experiments, we show an upper bound of translation quality if we had human-generated segmentation and punctuation on the output stream of speech recognition systems. In our oracle experiments we gain 1.78 BLEU points of improvements on the lecture test set. We build a monolingual translation system from German to German implementing segmentation and punctuation prediction as a machine translation task. Using the monolingual translation system we get an improvement of 1.53 BLEU points on the lecture test set, which is a comparable performance against the upper bound drawn by the oracle experiments.

1. Introduction

With increased performance in the area of automatic speech recognition (ASR), a large number of applications arise, which use the output of ASR systems as input. It is critical for these applications to have a clean, well-constructed input.

Especially for an application such as statistical machine translation (SMT), it is expected to have sentence-like segments in the input. As a first reason, most MT systems are trained using text data with well-defined sentence boundaries. Therefore, it is necessary to have proper segmentation before the translation to match the translation models in order to achieve better translation quality. Moreover, there are algorithmic constraints as well as user preferences, such as readability. When a sentence is excessively long, it either consumes a great deal of resources and time, or readability suffers.

If the input is already augmented with punctuation in the source language, it is advantageous to the training procedure of MT. In this case, there is no need to retrain the translation system with modification on the training data, in order to match the ASR output [1]. Nevertheless, most of the current ASR systems do not provide punctuation marks.

It is one of the challenging tasks to restore segmentation and punctuation in the output of an ASR system, especially for speech translation. Sentence segmentation in the ASR system is often generated using prosodic features (pause duration, pitch, etc.) and lexical cues (e.g. language model probability). However, the performance of sentence segmentation degrades in spontaneous speech. This is because a large amount of the spontaneous utterance is less grammatical compared to written texts [2] and there are fewer sentence-like-units (SU). Moreover, the presence of disfluencies in casual and spontaneous speech increases the difficulty of this task.

In this work we aim at recovering sentence segmentation and punctuation before translation as a preprocessing step and analyze its impact on the translation quality. The first goal of this paper is to investigate the upper bound of possible improvement on the translation quality when proper sentence segmentation and punctuation are achieved. For this we implement an oracle experiment, in which the human-generated segmentation and punctuation of manual transcripts are applied to ASR output before the translation process. In the second part of the oracle experiments, we insert the segmentation according to the ASR system into manual transcripts. As a second goal of this work, we build a monolingual translation system as a method to generate segments and punctuation marks. We will evaluate the performance of our monolingual translation system against the oracle experiment.

This paper is organized as follows. In Section 2, a brief overview of past research on segmentation and punctuation prediction is given. In Section 3, we present our baseline translation system used for this work. The oracle experiments and their results are described in Section 4, followed by Section 5 which contains the strategy to recover segmentation and punctuation and its results. Section 6 concludes our discussions.

2. Related Work

In previous work, the punctuation prediction problem was addressed to improve the readability as well as subsequent natural language processing [3]. In order to annotate ASR output with punctuation marks, they developed a maximum-entropy based approach. In this approach the insertion of punctuation was considered a tagging task. A maximum entropy tagger using both lexical and prosodic features was applied and the model was used to combine the different features. Their work showed that it is hard to distinguish between commas and default tags, and periods and question marks, since there is little prosodic information (similarly short or similarly long pause durations) and the features can cover a span longer than bigrams. They achieved a good F-measure for both reference transcriptions and transcriptions produced by a speech recognition system.

In [1] the authors made an extensive analysis on how to predict punctuation using a machine translation system. In this work, it was assumed that the ASR output already has the proper segmentation, which is sentence-like units. They investigated three different approaches to restore punctuation marks; prediction in the source language, implicit prediction, and prediction in the target language. Using a translation system to translate from unpunctuated to punctuated text, they showed significant improvements in the evaluation campaign of IWSLT 2011.

Among different motivations for the sentence segmentation, [4] split long sentence pairs in the bilingual training corpora to make full use of training data and improved model estimation for statistical machine translation (SMT). For the splitting they used the lexicon information to find splitting points. They showed that splitting sentences improved the performance for Chinese-English translation task. Similarly, to improve the performance of Example-based machine translation (EMBT) systems, [5] suggested a method to split sentences using sentence similarity based on edit-distance.

Combining prosodic and lexical information to detect sentence boundaries and disfluencies was demonstrated in the work of [6], where decision trees are used to model prosodic cues and N-grams for the language model. The authors suggested that having large amounts of recognizer output as training data for the models can improve the prediction task as it lowers the mismatch between training data and test set. The necessity of resegmentation for the ASR output was investigated in [2]. They trained a sentence segmenter based on pause duration and language model probabilities. It was emphasized that it is important to have commas in addition to periods within a sentence boundary, as it defines independently translatable regions and eventually improves translation performance.

Segmentation and punctuation issues are addressed together in [7]. The authors modified phrase tables so that the target side contains commas, but the source side does not contain any. Thus, when this modified phrase table was applied during translation, it recovered commas on the target side. For the segmentation and periods after each new line, they used a sentence segmenter based on a decision tree on the source side. They applied this method to three language pairs and achieved a significantly improved translation performance.

Table 1: *Information on the preprocessed source side of the test set*

ASR output	Sentences	2393
	Words without punctuation marks	27173
	WER	20.79%
Manual Transcript	Sentences	1241
	Words	29795
	Words without punctuation marks	26718
	Periods	1186
	Commas	1834
	Question marks	55

3. System Description

In this section we briefly introduce the statistical MT system that we use in this experiment.

As we work on translating speech in this experiment, we use the parallel TED[1] data and manual transcripts of lecture data containing 63k sentences as indomain data and adapt our models at the domain. The lecture data is collected internally at our university, and the domain of each lecture differs from the others. To better cope with domain-specific terminologies in university lectures, Wikipedia[2] title information is used as presented in [8].

For development and testing, we use the lecture data from different speakers. These are also collected internally from university classes and events. They consist of talks of 30 to 45 minutes and the topic varies from one speech to the other. For the development set we use manual transcripts of lectures, while for testing we use the transcripts generated by an ASR system. The development set consists of 14K parallel sentences, with 30K words on the source side and 33K words on the target side including punctuation marks. Detailed information on the source side of the test set, including the word error rate (WER) of the recognition output, can be found in Table 1.

The translation system is trained on 1.8 million sentences of German-English parallel data including the European Parliament data and News Commentary corpus. Before the training, the data is preprocessed and compound splitting for the German side is applied. Preprocessing consists of text normalization, tokenization, smartcasing, conversion of German words written according to the old spelling conventions into the new form of spelling.

[1] http://www.ted.com
[2] http://www.wikipedia.org

The Moses package [9] is used to build the phrase table. The 4-gram language model is trained on the English side of the above data with nearly 425 million words using the SRILM toolkit [10]. To extend source word context, a bilingual language model [11] is used. The POS-based reordering model as described in [12] is used for word reordering in order to account for the different word orders in source and target language. To cover long-range reorderings, we apply the modified reordering model as described in [13]. The translation hypotheses are generated using an in-house phrase-based decoder [14] and the optimization is performed using minimum error rate training (MERT) [15].

Translation models are built using the punctuated source side. Also for the other experiments, where there are no punctuation marks on the source side available, phrase tables are prepared in the same way.

4. Oracle Experiments

To investigate the impact of segmentation and punctuation marks on the translation quality, we conduct two experiments.

In the first experiment, we apply human-transcribed segments and punctuation marks to the output of the speech recognition system. Thus, words are still from an ASR system, but the segments and punctuation marks are reused from a human-generated transcript. In the second experiment, the segments in the output of the speech recognition system are applied to the human-generated transcripts. In this case, words are transcribed by human transcribers, but segmentation and punctuation are from an ASR system.

From these experiments we can observe how much impact the better segmentation and punctuation have for the performance of ASR output translation. We can also find how the segmentation according to an ASR system affects manual transcripts.

4.1. Oracle 1: Insertion of manual segments and punctuation marks into ASR output

Applying manual segments to the output of an ASR system requires the time stamp information for each utterance. We use this information from manual transcripts and segment the output stream generated by the ASR system according to it. The alignment information between ASR test sets and their manual transcripts is learned in order to insert punctuation marks. As punctuation marks, we consider period, comma, question mark, and exclamation mark. Punctuation marks such as period, question mark, and exclamation mark are usually followed by a new segment in manual transcripts, and commas are useful to define independently translatable regions [2].

Depending on which punctuation marks are inserted, three hypotheses are considered in this experiment.

- MTSegment: correct segments from a manual transcript are applied to the ASR test set.
- MTSegmentFullStop: correct segments and ".?!" from a manual transcript are applied to the ASR test set.
- MTSegmentAllPunct: correct segments and ".,?!" from a manual transcript, including commas, are applied to the ASR test set.

Therefore, the results in the hypothesis MTSegment show the boundary of performance improvement when the proper segmentation is given, while the hypothesis MTSegmentAllPunct shows the scenario when we also have good punctuation marks additionally. With the hypothesis MTSegmentFullStop, we intend to investigate how helpful it is for the translation quality to have commas or not.

To show the impact of the difference of the segmentation according to the ASR system and according to the hypothesis MTSegmentAllPunct, several consecutive segments are extracted from our test set. The translation of these two texts with different segmentation is presented in Table 2. The two source texts contain the same recognized words from an ASR system, but different segmentation and punctuation are applied. We can observe that when the text is with manual transcripts' segmentation, the translated text conveys the meaning of the sentence substantially better, as well as it provides improved readability. For example, the German participle *gesprochen*, which was translated into *spoken* using MTSegmenatAllPunct, is lost in the first segment in the ASR system and segmented into the next line. This leads to the loss of the

Table 2: *Translation using different segmentation according to ASR output and MTSegmentAllPunct hypothesis*

Segmentation	Translation
ASR	> We see here is an example from the European Parliament, the European Parliament 20 languages > And you try simultaneously by help human translator translators the > Talk to each of the speaker in other languages to translate it is possible to build computers > The similar to provide translation services
MTSegment-AllPunct	> We see here is an example from the European Parliament. > The European Parliament 20 languages are spoken, and you try by help human translator to translate simultaneously translators the speeches of the speaker in each case in other languages. > It is possible to build computers that are similar to provide translation services?

Table 3: *Disfluency and its affect on the automatic segmentation*
(Reference translation: Thus we consequently also have a third foot hold in Asia, in the Chinese region, in Hong Kong.)

System	
ASR output	> wir haben somit also auch ein drittes Standbein in Asien in > in chinesischen Raum in Hongkong
reference	> wir haben somit also auch ein drittes Standbein in Asien, im chinesischen Raum, in Hongkong.

information about this participle during the translation. An article and its following noun, *die Reden*, are also split using the original segmentation of the ASR system. It becomes the reason why the more suitable word *(the) speeches* in this context is not chosen, but *Talk*.

4.2. Oracle 2: Insertion of ASR output segments into manual transcripts

In addition to the insertion of proper segmentation and punctuation into the output of the ASR system, we perform another experiment where the segmentation in the output of the ASR system is applied to manual transcripts.

Although the segmentation from ASR output is obtained by incorporating language model probability and prosodic information such as pause duration, it is often not the best segmentation especially for spontaneous speech. This is caused by its nature of having less organized sentences and more disfluencies.

Table 3 depicts an example of incorrect automatic segmentation caused by disfluencies. As the speaker stutters, the automatic segmenter of the ASR system based on pause duration and a language model trained on clean texts inserts a new line.

In this experiment, we analyze the following three scenarios.

- ASRSegment: a manual transcript was segmented according to the segmentation of the ASR output.
- ASRSegmentComma: a manual transcript was segmented according to the segmentation of the ASR output, and commas are removed.
- ASRSegmentAllPunct: a manual transcript was segmented according to the segmentation of the ASR output, and all four punctuation marks are removed.

The four punctuation marks correspond to ".,?!" as in the first oracle experiment. To segment a manual transcript as in the ASR output, we use an algorithm which is commonly used for evaluating machine translation output with automatic sentence segmentation [16]. This method is based on the Levenshtein edit distance algorithm [17]. By backtracing the decisions of the Levenshtein edit distance algorithm, we can find the Levenshtein alignment between the reference words and the words in the ASR output.

In this work, the ASR output plays the role of a reference and using this algorithm we are able to find a resegmentation of the human reference transcript based on the original segmentation of the ASR output.

4.3. Results

Table 4 depicts the results of the two experiments in numbers. The scores are reported as case-insensitive BLEU [18] scores, without considering punctuation marks. This aims at analyzing the impact of the segmentation and punctuation solely on the translation quality.

Table 4: *Influence of oracle segmentation and punctuation on the speech translation quality*

System		BLEU
ASR		**20.70**
Oracle 1	MTSegment	21.42
	MTSegmentFullStop	22.18
	MTSegmentAllPunct	22.48
Transcripts		**27.99**
Oracle 2	ASRSegment	26.38
	ASRSegmentComma	26.36
	ASRSegmentAllPunct	25.54

For the hypotheses MTSegment, ASRSegmentAllPunct and tests on the ASR output, we create phrase tables removing punctuation marks on the source side in order to make a better match between the test set and the phrase table. To evaluate the translation hypotheses of ASR output and the ASRSegmentation experiments, we resegmented our translation hypotheses to have the same number of segments as the reference as shown in [16].

From this table we observe that having the correct segmentation and punctuation improves the translation quality significantly. When the human-transcribed segmentation and punctuation are available, an improvement of 1.78 BLEU is observable on the test set.

Another interesting point is when we compare MTSegmentAllPunct to MTSegmentFullStop, we see the steady improvement of 0.3 BLEU in translation from having commas on the source side. This is congruent with the findings in [2], that inserting commas in addition to periods improves translation quality. In our case, the scores are evaluated ignoring punctuation marks. Thus, the improvement on BLEU means

that by having proper punctuation marks the translation quality itself can be improved.

On the other hand, we can observe from Table 4 that by simply changing the segmentation of the transcripts we lose 1.6 BLEU scores in translation performance. As shown in Table 1, there are almost twice as many segments in the ASR output compared to the manual transcript. This can be one reason of the drastic drop of the translation quality. We also observed from this translation that incorrect reordering of words occasionally happens within a segment, when the segment is not a sentence-like unit but a part of a sentence.

Removing commas from ASRSegment does not result in a big performance drop in ASRSegmentComma. Often, the segments from the ASR system do not match with the phrase boundaries learned in the text translation system, which results in having fewer independently translatable regions separated by commas. In addition to this, losing all punctuation information leads to a further performance drop of 0.84 BLEU scores.

5. Monolingual Translation System

In this section we introduce our monolingual translation system that we used to predict the segmentation and punctuation.

Inspired by [1], we build a monolingual translation system to predict segmentation and punctuation marks in the translation process. This monolingual translation system translates non-punctuated German into punctuated German. Using this system we predict punctuation marks as well as segmentation before the actual translation of the test sets. The output of this system becomes the input to our regular text translation system which is trained using training data with punctuation marks.

When translating the output of the monolingual translation system, no preprocessing is applied as the test set is already preprocessed before going through the monolingual translation system. The monolingual translation system does neither alter any words nor reorder words, but it is used solely for changing segments and inserting punctuation marks.

In order to build this system, we first process the training data to make the source side not contain any punctuation marks, but the target side contain all punctuation marks. The training data statistics on the target side is shown in Table 5.

Table 5: *Information on the preprocessed punctuated German side of the training data*

Words	46.32M
Periods	1.76M
Commas	2.88M
Question marks	0.10M
Exclamation marks	0.07M

For a language model, we use 4-grams and it is trained on the punctuated German data. Also, no reordering model is used as we use the monotone alignment.

The difference of our monolingual translation system to the work in [1] is that in our work the monolingual translation system is used to predict sentence segmentation additionally. In their work, it was assumed that the segmentation of the speech recognition output was given and corresponded to at least sentence-like units. Therefore, their monolingual translation system was used to reconstruct punctuation marks only with using three different strategies.

It was shown in the previous section that the segmentation generated from an ASR system is not necessarily the best segmentation, especially when the recognized text is spontaneous speech with less grammatical sentences and more disfluencies. In this work, we aim at improving segmentation in addition to inserting punctuation marks using this monolingual translation system. To perform this it is required to modify the training data as well as development and test sets.

5.1. Data preparation

Usually training data for conventional text translation systems is segmented by human transcribers so that it has punctuation such as a full stop, a question mark, or an exclamation mark at the end of each line. Therefore, if we use this training data to translate the ASR test sets, translation models would more likely insert a punctuation mark at the end of every line of the ASR test set during translation. From this observation, we resegment training corpora randomly so that every segment is not necessarily one proper sentence-like unit. The development set is modified in the same way.

The test sets for this monolingual translation system are also prepared differently, using the idea of a sliding window. Examplary sentences from our test set are shown in Table 6. In this table, each line contains 8 words and the first line starts with a word *der*. In the second line, we have the next starting word *bildet*, which was the second word in the first line. At the same time, we have a new encountering word *gesehen* at the end of the line. When the length of a sliding window is *l*, each line consists of *l-1* words from the previous line and 1 new word. Thus, the *n*th line contains the *n*th to *n+l-1*th word of a test set. The test set prepared in this way has the same length as the number of words in the original test set. In this way we can have up to *l* spaces between words. For those spaces we want to investigate how probable it is to have a punctuation mark in that word space. In this experiment, we constrain the length of sliding window *l* to 10.

This differently formatted test set enters the monolingual translation process in a normal way, line by line. The translation of the test set shown in Table 6 using our monolingual translation system is illustrated in Table 7. We see that words such as *Normalform* and *gesehen* are followed by certain punctuation marks.

Table 6: *Test set preparation for the monolingual translation system*

der	bildet	die	sogenannte	konjunktive	Normalform	wir	haben
bildet	die	sogenannte	konjunktive	Normalform	wir	haben	gesehen
die	sogenannte	konjunktive	Normalform	wir	haben	gesehen	dass
sogenannte	konjunktive	Normalform	wir	haben	gesehen	dass	wir
konjunktive	Normalform	wir	haben	gesehen	dass	wir	diese
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 7: *Translation using the monolingual translation system*

der	bildet	die	sogenannte	konjunktive	**Normalform.**	Wir	haben
bildet	die	sogenannte	konjunktive	**Normalform.**	Wir	haben	**gesehen,**
die	sogenannte	konjunktive	**Normalform.**	Wir	haben	**gesehen,**	dass
sogenannte	konjunktive	**Normalform.**	Wir	haben	**gesehen,**	dass	wir
konjunktive	**Normalform.**	Wir	haben	**gesehen,**	dass	wir	diese
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

5.2. Punctuation prediction criteria

A punctuation mark is chosen if the same punctuation mark is found same or more often than a given threshold. If more than one punctuation mark appears more than the threshold in the same word space, the most frequent one is chosen. There are some cases where we have the same frequency for multiple punctuation marks; in this case we put a different priority on punctuation marks. For example, in this experiment we put higher priority for a period over a comma.

In this experiment, we evaluate the translation quality over a varying threshold, from 1 to 9. We exempt the case when the threshold is 10, the length of the sliding window. In this case, one punctuation mark has to appear all the 10 word spaces after a word in order to be inserted. This condition is so restrictive that only few full stops are generated, which causes unaffordable computational time consumption for the translation procedure.

In the same way as in the oracle experiment, we consider four punctuation marks here: period, comma, question mark, and exclamation mark. A new segment is introduced when either a period, question mark, or exclamation mark is predicted, in order to have congruence with the manual transcripts.

To make the hypotheses comparable with the oracle experiments, we considered three different hypotheses of reconstructing segmentation and punctuation.

- MonoTrans-Segment: monolingual translation system is used for segmentation prediction only.
- MonoTrans-FullStop: monolingual translation system is used for segmentation and full stop prediction.
- MonoTrans-AllPunct: monolingual translation system is used for segmentation and all punctuation marks prediction.

5.3. Results

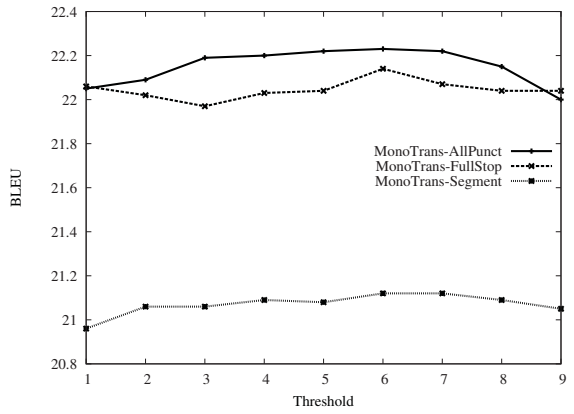
In order to analyze the effect of the varying threshold for the monolingual translation system, first we use the same threshold value for all punctuation marks. The number of punctuation marks predicted using the same threshold are shown in Table 8. As shown in the table we could predict periods and commas, but we could not generate question marks and exclamation marks. A reason might be that question mark and exclamation mark are already rare in the manual transcript. In addition, we do not have many of them appearing in the training corpora, compared to the frequency of the other punctuation marks. The number of periods in Table 8, therefore, is the same as the number of segments predicted.

Figure 1 presents the translation performance of the three hypotheses in BLEU over different threshold values. In this experiment as well, the same threshold value is used for all the different punctuation marks. Even though we ob-

Table 8: *Punctuation marks predicted using the monolingual translation system, with a different threshold. The number of punctuation marks in the manual transcript is also given as a comparison.*

Threshold	1	2	3	4	5	6	7	8	9	Manual Transcript
Periods	1,273	970	881	861	851	841	817	736	464	1,186
Commas	2,741	2,190	1,973	1,915	1,904	1,889	1,857	1,773	1,486	1,834

Figure 1: *Translation performance with varying threshold values*



tain more segments the lower we set the threshold value, each hypothesis still outperforms the translation of ASR output (20.70 in BLEU). The threshold value can go down to 1 without any significant loss in BLEU. As shown by the curve of MonoTrans-FullStop, the performance is already good when having segments from periods only. When we compare MonoTrans-AllPunct and MonoTrans-FullStop, the performance of MonoTrans-AllPunct fluctuates relatively more while that of MonoTrans-FullStop stays more stagnant. From this observation we notice the necessity of another experiment where different threshold values for period and commas are used, as the performance can be improved with fewer commas when there are more segments.

Table 9 presents how close we can get toward the oracle experiments when using the segmentation and punctuation predicted output from the monolingual translation system. The numbers from an oracle experiment and ASR output are also shown for comparison. The condition Test1 represents the results where the threshold 6 was used for both period and comma.

As depicted in this table, all three hypotheses of our monolingual translation system beat the translation quality using the ASR output with a significant difference. When both segmentation and punctuation are predicted using our monolingual translation system, we gain 1.53 BLEU points on our test set, which is only 0.25 BLEU points less than a result from the oracle experiment.

In order to maintain a similar number of segments to the manual transcript, but still have the "helpful" number of commas for translation, we separate the threshold value for period and comma. Test2 in Table 9 depicts the translation performance when we use the threshold value 1 for period and 6 for comma. Thus, a comma is chosen when it is found more than 5 times at the space between words. Compared to the case where the same threshold value of 6 for both punctuation marks is used, we obtain more than 150% of the original number of segments. However, we can still maintain a similar translation performance, showing only a drop of 0.06 BLEU points in the hypothesis MonoTrans-AllPunct.

Table 9: *Results of using monolingual translation system to reconstruct segmentation and punctuation, compared to the oracle experiment*

System	BLEU	
	Test1	Test2
ASR	20.70	
MonoTrans-Segment	21.12	20.97
Oracle 1: MTSegment	21.42	
MonoTrans-FullStop	22.14	22.06
Oracle 1: MTSegmentFullStop	22.18	
MonoTrans-AllPunct	22.23	22.17
Oracle 1: MTSegmentAllPunct	22.48	
Number of segments	851	1,292

Predicting a new line only after a period performs well for the translation. However, the numbers shown in Table 1 indicate that inserting a new line only after a period provides half of the number of segments that our ASR system produced for the test set. Therefore, to compare the performance of the ASR segmenter in a fair condition, we conduct another experiment where a new line is inserted whenever a punctuation mark, including comma, is predicted. For this experiment we use the same threshold 8 for all punctuation marks, so that we can have similar number of segments as in the ASR output. By doing so we could obtain 2,509 segments, which is nearly 200 segments more than the ASR output. From this we gained 21.67 BLEU points for the MonoTrans-AllPunct hypothesis. Although the score of the hypothesis MonoTrans-AllPunct is 0.5 BLEU points lower than previous two tests, the score is still around 1 BLEU point higher than the translation quality of raw ASR output.

6. Conclusion

In this paper, we first presented the impact of segmentation and punctuation on the output of speech recognition systems by implementing oracle experiments. Experiments have shown that we can gain up to 1.78 BLEU points of improvement on the translation quality if we apply the manual segmentation and punctuation to the ASR output. On the other hand, when we apply the segmentation and punctuation of speech recognition output to the manual transcripts, we have an overall loss of 2.45 BLEU points on the translation quality. Therefore we show that the segmentation produced by ASR systems may not assure the best translation performance, but a separate process to segment the ASR stream before the translation can help the translation performance.

In the second part of the paper, the monolingual translation system is used to predict segmentation and punctuation

in ASR output. In order to implement this system, we change the format of the training corpora as well as the development and test set. By using the monolingual translation system, we gain more than 1.5 BLEU points on the ASR test set.

In future work, we would like to pursue on developing the monolingual translation system with different ways to extract relevant phrases for the task. Furthermore, the analysis on disfluencies in speech is necessary to improve the segmentation and punctuation prediction.

7. Acknowledgements

This work was achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

8. References

[1] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling Punctuation Prediction as Machine Translation," in *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011.

[2] S. Rao, I. Lane, and T. Schultz, "Optimizing Sentence Segmentation for Spoken Language Translation," in *Proc. of Interspeech*, Antwerp, Belgium, 2007.

[3] J. Huang and G. Zweig, "Maximum Entropy Model for Punctuation Annotation from Speech." in *Proc. of ICSLP*, Denver, CO, USA, 2002.

[4] J. Xu, R. Zens, and H. Ney, "Sentence Segmentation using IBM Word Alignment Model," in *EAMT 2005*, Budapest, Hungary, 2005.

[5] T. Doi and E. Sumita, "Splitting Input Sentence for Machine Translation Using Language Model with Sentence Similarity," in *Coling 2004*.

[6] A. Stolcke, E. Shriberg, R. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tür, and Y. Lu, "Automatic Detection of Sentence Boundaries and Disfluencies Based on Recognized Words," in *Proc. of ICSLP*, Sidney, Australia, 1998.

[7] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence Segmentation and Punctuation Recovery for Spoken Language Translation," in *ICASSP*, Las Vegas, Nevada, USA, April 2008.

[8] J. Niehues and A. Waibel, "Using Wikipedia to Translate Domain-specific Terms in SMT," in *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, CA, 2011.

[9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation," in *ACL 2007, Demonstration Session*, Prague, Czech Republic, June 23 2007.

[10] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *Proc. of ICSLP*, Denver, Colorado, USA, 2002.

[11] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, Edinburgh, UK, 2011.

[12] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *TMI*, Skövde, Sweden, 2007.

[13] J. Niehues and M. Kolss, "A POS-Based Model for Long-Range Reorderings in SMT," in *Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece, 2009.

[14] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *Int. Conf. on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[15] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *Workshop on Data-drive Machine Translation and Beyond (WPT-05)*, Ann Arbor, MI, 2005.

[16] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating Machine Translation Output with Automatic Sentence Segmentation." in *Internat. Workshop on Spoken Language Translation*, Pittsburgh, USA, 2005.

[17] V. I. Levenshtein, *Binary Codes Capable of Correcting Deletions, Insertions and Reversals.*, ser. 10(8). Soviet Physics Doklady, 1966, vol. pp. 707-710.

[18] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation." IBM Research Division, T. J. Watson Research Center, Tech. Rep. RC22176 (W0109-022), 2002.

Sequence Labeling-based Reordering Model for Phrase-based SMT

Minwei Feng, Jan-Thorsten Peter, Hermann Ney

Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
Aachen, Germany
<surname>@cs.rwth-aachen.de

Abstract

For current statistical machine translation system, reordering is still a major problem for language pairs like Chinese-English, where the source and target language have significant word order differences. In this paper, we propose a novel reordering model based on sequence labeling techniques. Our model converts the reordering problem into a sequence labeling problem, i.e. a tagging task. For the given source sentence, we assign each source token a label which contains the reordering information for that token. We also design an unaligned word tag so that the unaligned word phenomenon is automatically implanted in the proposed model. Our reordering model is conditioned on the whole source sentence. Hence it is able to catch the long dependency in the source sentence. Although the learning on large scale task requests notably amounts of computational resources, the decoder makes use of the tagging information as soft constraints. Therefore, the training procedure of our model is computationally expensive for large task while in the test phase (during translation) our model is very efficient. We carried out experiments on five Chinese-English NIST tasks trained with BOLT data. Results show that our model improves the baseline system by 1.32 BLEU 1.53 TER on average.

1. Introduction

The systematic word order difference between two languages, pose a challenge for current statistical machine translation (SMT) systems. The system has to decide in which order to translate the given source words. This problem is known as the reordering problem. As shown in [1], if arbitrary reordering is allowed, the search problem is NP-hard.

In this paper, we propose a novel tagging style reordering model. Our model converts the reordering problem into a sequence labeling problem, i.e. a tagging task. For the given source sentence, we assign each source token a label which contains the reordering information for that token. We also design an unaligned word tag so that the unaligned word phenomenon is automatically implanted in the proposed model. Our model is conditioned on the whole source sentence. Hence it is able to capture the long dependency in the source sentence. We compare two training methods: conditional random fields (CRFs) and recurrent neural network (RNN). Although the learning on large scale task requests notably amounts of computational resources, the decoder makes use of the tagging information as soft constraints. Therefore, the training procedure of our model is computationally expensive while in the test phase (during translation) our model is very efficient.

The remainder of this paper is organized as follows: Section 2 reviews the related work for solving the reordering problem. Section 3 introduces the basement of this research: the principle of statistical machine translation. Section 4 describes the proposed model. Section 5 provides the experimental configuration and results. Conclusion will be given in Section 6.

2. Related Work

Many ideas have been proposed to address the reordering problem. Early work focuses on reordering constraints, e.g. using ITG constraints [2] and IBM constraints [3] to model the sequence permutation. Within the phrase-based SMT framework there are mainly three stages where improved reordering could be integrated:

1. Reorder the source sentence. So that the word order of source and target sentences is similar. Usually it is done as the preprocessing step for both training data and test data.

2. In the decoder, add models in the log-linear framework or constraints in the decoder to reward good reordering options or penalize bad ones.

3. In the reranking framework.

For the first point, [4] used manually designed rules to reorder parse trees of the source sentences as a preprocessing step. Based on shallow syntax, [5] used rules to reorder the source sentences on the chunk level and provide a source-reordering lattice instead of a single reordered source sentence as input to the SMT system. Designing rules to reorder

the source sentence is conceptually clear and usually easy to implement. In this way, syntax information can be incorporated into phrase-based SMT systems. However, one disadvantage is that the reliability of the rules is often language pair dependent.

In the second category, researchers try to inform the decoder on what a good reordering is or what a suitable decoding sequence is. [6] used a discriminative reordering model to predict the orientation of the next phrase given the previous phrase. [7] presents a translation model that constitutes a language model of a sort of bilanguage composed of bilingual units. From the reordering point of view, the idea is that the correct reordering is to find the suitable order of translation units. [8] puts the syntactic cohesion as a soft constraint in the decoder to guide the decoding process to choose those translations that do not violate the syntactic structure of the source sentence. Adding new features in the log-linear framework has the advantage that the new feature has access to the whole search space. Another advantage of methods in this category is that we let the decoder decide the weights of features, so that even if one model gives wrong estimation sometimes, it can still be corrected by other models. Our work in this paper belongs to this category.

In the reranking step, the system has the last opportunity to choose a good translation. [9] describe the use of syntactic features in the rescoring step. They report the most useful feature is IBM Model 1 score. The syntactic features contribute very small gains. Another disadvantage of carrying out reordering in reranking is the representativeness of the N-best list is often a question mark.

3. Translation System Overview

In this section, we are going to describe the phrase-based SMT system we used for the experiments.

In statistical machine translation, we are given a source language sentence $f_1^J = f_1 \dots f_j \dots f_J$. The objective is to translate the source into a target language sentence $e_1^I = e_1 \dots e_i \dots e_I$. The strategy is among all possible target language sentences, we will choose the one with the highest probability:

$$\hat{e_i^I} = \arg\max_{I, e_1^I} \{Pr(e_1^I | f_1^J)\} \tag{1}$$

We model $Pr(e_1^I | f_1^J)$ directly using a log-linear combination of several models [10]:

$$Pr(e_1^I | f_1^J) = \frac{\exp\Big(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\Big)}{\sum_{I', {e'}_1^{I'}} \exp\Big(\sum_{m=1}^{M} \lambda_m h_m({e'}_1^{I'}, f_1^J)\Big)} \tag{2}$$

The denominator is to make the $Pr(e_1^I | f_1^J)$ to be a probability distribution and it depends only on the source sentence f_1^J. For search, the decision rule is simply:

$$\hat{e_i^I} = \arg\max_{I, e_1^I} \Big\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \Big\} \tag{3}$$

The model scaling factors λ_1^M are trained with Minimum Error Rate Training (MERT).

In this paper, the phrase-based machine translation system is utilized [11, 12, 13]. The translation process consists in segmenting the source sentence according to the phrase table which is built from the word alignment. The translation of each of these segments consists in just extracting the target side from the phrase pair. With the corresponding target side, the final translation is the composition of these translated segments. In this last step, reordering is allowed.

4. Tagging-style Reordering Model

In this section, we describe the proposed model. First we will describe the training process. Then we explain how to use the model in the decoder.

4.1. Modeling

Figure 1 shows the modeling steps. The first step is word alignment training. Figure 1(a) is an example after GIZA++ training. If we regard this alignment as a translation result, i.e. given the source sentence f_1^7, the system translates it into the target sentence e_1^7. The alignment link set $\{a_1 = 3, a_3 = 2, a_4 = 4, a_4 = 5, a_5 = 7, a_6 = 6, a_7 = 6\}$ reveals the decoding process, i.e. the alignment implies the order in which the source words should be translated, e.g. the first generated target word e_1 has no alignment, we can regard it as a translation from a NULL source word; then the second generated target word e_2 is translated from f_3. We reorder the source side of the alignment to get Figure 1(b). Figure 1(b) implies the source sentence decoding sequence information, which is depicted in Figure 1(c). Using this example we describe the strategies we used for special cases in the transformation from Figure 1(b) to Figure 1(c):

- ignore the unaligned target word, e.g. e_1
- the unaligned source word should follow its preceding word, the unaligned feature is kept with a $*$ symbol, e.g. f_2^* is after f_1
- when one source word is aligned to multiple target words, only keep the alignment that links the source word to the first target word, e.g. f_4 is linked to e_5 and e_6, only $f_4 - e_5$ is kept. In other words, we use this strategy to guarantee that every source word appears only once in the source decoding sequence.
- when multiple source words aligned to one target word, put together the source words according to their original relative positions, e.g. e_6 is linked to f_6 and f_7. So in the decoding sequence, f_6 is before f_7.

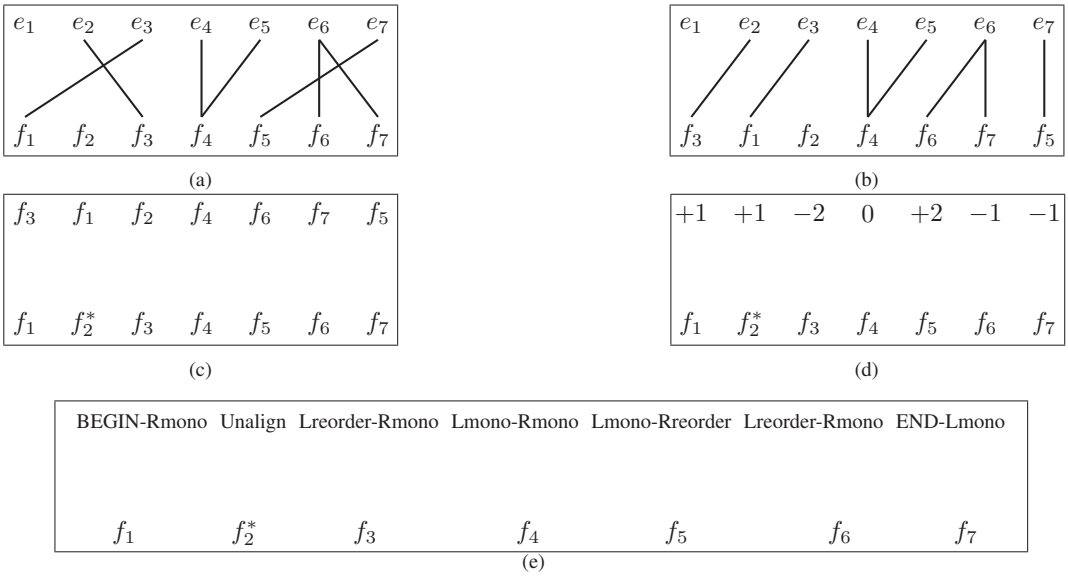


Figure 1: Modeling process illustration.

Now Figure 1(c) shows the original source sentence and its decoding sequence. By using the strategies above, it is guaranteed that the source sentence and its decoding sequence has the exactly same length. Hence the relation can be modeled by a function $F(f)$ which assigns a value for each of the source word f. Figure 1(d) manifests this function. The positive function values mean that compared to the original position in the source sentence, its position in the decoding sequence should move right. If the function value is 0, the word's position in original source sentence and its decoding sequence is same. For example, f_1 is the first word in the source sentence but it is the second word in the decoding sequence. So its function value is $+1$ (move right one position).

Now Figure 1(d) converts the reordering problem into a sequence labeling or tagging problem. To make the computational cost to a reasonable level, we do a final step simplification in Figure 1(e). Suppose the longest sentence length is 100, then according to Figure 1(d), there are 200 tags (from -99 to +99 plus the unalign tag). As we will see later, this number is too large for our task. We instead design nine tags. For a source word f_j in one source sentence f_1^J, the tag of f_j will be one of the following:

Unalign f_j is an unaligned source word

BEGIN-Rmono $j = 1$ and f_{j+1} is translated *after* f_j (Rmono for right monotonic)

BEGIN-Rreorder $j = 1$ and f_{j+1} is translated *before* f_j (Rreorder for right reordered)

END-Lmono $j = J$ and f_{j-1} translated *before* f_j (Lmono for left monotonic)

END-Lreorder $j = J$ and f_{j-1} translated *after* f_j (Lreorder for left reordered)

Lmono-Rmono $1 < j < J$ and f_{j-1} translated *before* f_j and f_j translated *before* f_{j+1}

Lreorder-Rmono $1 < j < J$ and f_{j-1} translated *after* f_j and f_j translated *before* f_{j+1}

Lmono-Rreorder $1 < j < J$ and f_{j-1} translated *before* f_j and f_j translated *after* f_{j+1}

Lreorder-Rreorder $1 < j < J$ and f_{j-1} translated *after* f_j and f_j translated *after* f_{j+1}

Up to this point, we have converted the reordering problem into a tagging problem with nine tags. The transformation in Figure 1 is conducted for all the sentence pairs in the bilingual training corpus. After that, we have built an "annotated" corpus for the training. For this supervised learning task, we choose the approach conditional random fields (CRFs) [14, 15, 16] and recurrent neural network (RNN) [17, 18, 19].

For the first method, we adopt the linear-chain CRFs. However, even for the simple linear-chain CRFs, the complexity of learning and inference grows quadratically with respect to the number of output labels and the amount of structural features which are with regard to adjacent pairs of labels. Hence, to make the computational cost as low as possible, two measures have been taken. Firstly, as described above we reduce the number of tags to nine. Secondly, we add source sentence part-of-speech (POS) tags to the input. For features with window size one to three, both source words and its POS tags are used. For features with window size four and five, only POS tags are used.

As the second method, we use recurrent neural network (RNN). RNN is closely related with Multilayer Perceptrons (MLP) [20, 21], but the output of one ore more hidden layers is reused as additional inputs for the network in the next time step. This structure allows the RNN to learn whole sequences without restricting itself to a fixed input window. A plain RNN has only access to the previous events in the input sequence. Hence we adopt the bidirectional RNN (BRNN) [22] which reads the input sequence from both directions before making the prediction. The long short-term memory (LSTM) [23] is applied to counter the effects that long distance dependencies are hard to learn with gradient descent. This is often referred to as vanishing gradient problem [24].

4.2. Decoding

Once the model training is finished, we make inference on develop and test corpora. After that we get the labels of the source sentences that need to be translated. In the decoder, we add a new model which checks the labeling consistency when scoring an extended state. During the search, a sentence pair (f_1^J, e_1^I) will be formally splitted into a segmentation S_1^K which consists of K phrase pairs. Each $s_k = (i_k; b_k, j_k)$ is a triple consisting of the last position i_k of the kth target phrase $\tilde{e}_k$. The start and end position of the kth source phrase $\tilde{f}_k$ are b_k and j_k. Suppose the search state is now extended with a new phrase pair $(\tilde{f}_k, \tilde{e}_k)$:

$$\tilde{f}_k := f_{b_k} \dots f_{j_k} \tag{4}$$

$$\tilde{e}_k := e_{i_{k-1}+1} \dots e_{i_k} \tag{5}$$

We have access to the old coverage vector, from which we know if the new phrase's left neighboring source word f_{b_k-1} and right neighboring source word f_{j_k+1} have been translated. We also have the word alignment within the new phrase pair, which is stored during the phrase extraction process. Based on the old coverage vector and alignment, we can repeat the transformation in Figure 1 to calculate the labels for the new phrase. The added model will then check the consistence between the calculated labels and the labels predicted by the reordering model. The number of source words that have inconsistent labels is the penalty and is then added into the log-linear framework as a new feature.

5. Experiments

In this section, we describe the baseline setup, the CRFs training results, the RNN training results and translation experimental results.

5.1. Experimental Setup

Our baseline is a phrase-based decoder, which includes the following models: an n-gram target-side language model (LM), a phrase translation model and a word-based lexicon model. The latter two models are used for both directions: $p(f|e)$ and $p(e|f)$. Additionally we use phrase count features, word and phrase penalty. The reordering model for the baseline system is the distance-based jump model which uses linear distance. This model does not have hard limit. We list the important information regarding the experimental setup below. All those conditions have been kept same in this work.

- lowercased training data (Table 1) from the BOLT task alignment trained with GIZA++
- tuning corpus: NIST06
  test corpora: NIST02 03 04 05 and 08
- 5-gram LM (1 694 412 027 running words) trained by SRILM toolkit [25] with modified Kneser-Ney smoothing
  training data: target side of bilingual data.
- BLEU [26] and TER [27] reported
  all scores calculated in lowercase way.
- Wapiti toolkit [16] used for CRFs; RNN is built by the RNNLIB [28] toolkit.

	Chinese	**English**
Sentences	5 384 856	
Running Words	115 172 748	129 820 318
Vocabulary	1 125 437	739 251

Table 1: training data statistics

Table 1 contains the data statistics used for translation model and LM. For the reordering model, we take two further filtering steps. Firstly, we delete the sentence pairs if the source sentence length is one. When the source sentence has only one word, the translation will be always monotonic and the reordering model does not need to learn this. Secondly, we delete the sentence pairs if the source sentence contains more than three contiguous unaligned words. When this happens, the sentence pair is usually low quality hence not suitable for learning. The main purpose of the two filtering steps is to further lay down the computational burden. The label distribution is depicted in Figure 2. From the figure we can see that most words are monotonic. We then divide the corpus to three parts: train, validation and test. The source side data statistics for the reordering model training is given in Table 2 (target side has only nine labels).

	train	**validation**	**test**
Sentences	2 973 519	400 000	400 000
Running Words	62 263 295	8 370 361	8 382 086
Vocabulary	454 951	149686	150 007

Table 2: reordering model training data statistics

5.2. CRFs Training Results

The toolkit Wapiti [16] is used in this paper. We choose the classical optimization algorithm limited memory BFGS (L-BFGS) [29]. For regularization, Wapiti uses both the ℓ^1 and ℓ^2 penalty terms, yielding the elastic-net penalty of the form

$$\rho_1 \cdot \parallel \theta \parallel_1 + \frac{\rho_2}{2} \cdot \parallel \theta \parallel_2^2 \quad (6)$$

In this work, we use as many features as possible because ℓ^1 penalty $\rho_1 \parallel \theta \parallel_1$ is able to yield sparse parameter vectors, i.e. using a ℓ^1 penalty term implicitly performs the feature selection. The computational costs are given here: on a cluster with two AMD Opteron(tm) Processor 6176 (total 24 cores), the training time is about 16 hours, peak memory is around 120G. Several experiments have been done to find the suitable hyperparameter ρ_1 and ρ_2, we choose the model with lowest error rate on validation corpus for translation experiments. The error rate of the chosen model on test corpus (the test corpus in Table 2) is 25.75% for token error rate and 69.39% for sequence error rate. The feature template we set initially will generate 722 999 637 features. After training 36 902 363 features are kept.

5.3. RNN Training Results

We also applied RNN to the task as an alternative approach to CRFs. The here used RNN implementation is RNNLIB [28] which has support for long short term memory (LSTM) [30]. We used a one of k encoding for the input word and also for the labels. After testing several configurations over the validation corpus we used a network with LSTM 200 nodes in the hidden layer. The RNN has a token error rate of 27.31% and a sentence error rate of 77.00% over the test corpus in Table 2. The RNN is trained on a similar computer as above. RNNLIB utilizes only one thread. The training time is about three and a half days and peak memory consumption is 1G .

5.4. Comparison of CRFs and RNN errors

From machine learning point of view, CRFs performs better than RNN (token error rate 25.75% vs 27.31%). Both error rate values are much higher than what we usually see in part-of-speech tagging task. The main reason is that the "annotated" corpus is converted from word alignment which contains lots of error. However, as we will show later, the model trained with both CRFs and RNN help to improve the translation quality.

Table 3 and Table 4 demonstrate the confusion matrix of the CRFs and RNN errors over the test corpus. The rows represent the correct tag that the classifier should have predicted and the columns are the actually predicted tags. E.g. the nunber 687724 in first row and first column of Table 3 tells that there are 687724 correctly labeled **Unalign** tags. The number 15084 in first row and second column of Table 3 represents that there are 15084 **Unalign** tags labeled incorrectly to **Begin-Rmono**. Therefore, numbers on the diagonal from the upper left to the lower right corner represent the amount of correctly classified tags and all other numbers show the amount of false labels. The many zeros show that both classifier rarely make mistake for the label "**BEGIN-**$*$" which only occur at the beginning of a sentence. The same is true for the "**END-**$*$" labels.

5.5. Translation Results

Results are summarized in Table 6. Automatic measure BLEU and TER scores are provided. Also we report significance testing results on both BLEU and TER. We perform bootstrap resampling with bounds estimation as described in [31]. We use the 95% confidence threshold (denoted by ‡ in the table) to draw significance conclusions. Besides the five test corpora, we add a column **avg.** to show the average improvements. We also add a column **Index** for score reference convenience.

From Table 6 we see that our proposed reordering model using CRFs improves the baseline by 0.98 BLEU and 1.21 TER on average, while the proposed reordering model using RNN improves the baseline by 1.32 BLEU and 1.53 TER on average. For the CRFs-based model, the largest BLEU improvement 1.15 is from NIST05 and the largest TER improvement 1.57 is from NIST03. The improvements are even larger with the tags created by the RNN with a BLEU improvement of 1.70 and a TER improvement 1.98 for NIST02. For line 3 and 6, all the scores are better than their corresponding baseline values with more than 95% confidence. For line 2 and 5, three out of the five scores are better than their corresponding baseline values with more than 95% confidence. The results show that our proposed idea improves the baseline system and RNN trained model performs better than CRFs trained model, in terms of both automatic measure and significance test.

To investigate why RNN has lower performance for the tagging task but achieves better BLEU, we build a 5-gram LM on the source side of the training corpus in Table 2. Perplexity values are provided in Table 5. We see clearly that the perplexity of the test corpus for reordering model comparison is much lower than those NIST corpora for translation experiments. In other words, there exists mismatch of the data for reordering model training and actual MT data. This could explain why CRFs is superior to RNN for labeling problem while RNN is better for MT tasks.

	Running Words	**OOV**	**Perplexity**
Test in Table 2	8 382 086	0	6.665
NIST02	22 749	391	234.494
NIST03	24 180	518	346.242
NIST04	49 612	700	223.492
NIST05	29 966	511	342.925
NIST08	32 502	998	473.975

Table 5: Perplexity

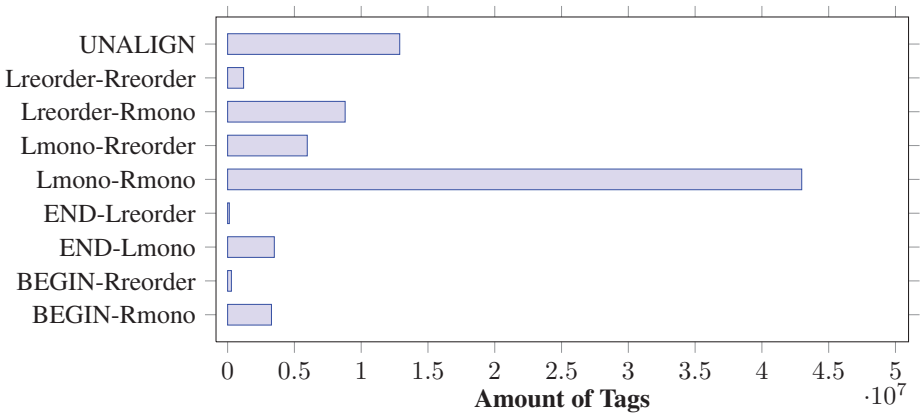


Figure 2: Tags distribution illustration.

Reference \ Prediction	Unalign	BEGIN-Rm	BEGIN-Rr	END-Lm	END-Lr	Lm-Rm	Lr-Rm	Lm-Rr	Lr-Rr
Unalign	687724	15084	850	7347	716	493984	107364	43457	9194
BEGIN-Rmono	3537	338315	6209	0	0	0	0	0	0
BEGIN-Rreorder	419	12557	17054	0	0	0	0	0	0
END-Lmono	1799	0	0	365635	3196	0	0	0	0
END-Lreorder	510	0	0	5239	7913	0	0	0	0
Lmomo-Rmono	188627	0	0	0	0	4032738	176682	150952	13114
Lreorder-Rmono	88177	0	0	0	0	369232	433027	27162	15275
Lmomo-Rreorder	32342	0	0	0	0	268570	24558	296033	10645
Lreorder-Rreorder	9865	0	0	0	0	34746	20382	16514	45342
Recall	50.36%	97.20%	56.79%	98.65%	57.92%	88.40%	46.42%	46.83%	35.74%
Precision	67.89%	92.45%	70.73%	96.67%	66.92%	77.56%	56.83%	55.42%	48.46%

Table 3: CRF Confusion Matrix. Abbreviations: Lmono(Lm) Lreorder(Lr) Rmono(Rm) Rreorder(Rr)

Reference \ Prediction	Unalign	BEGIN-Rm	BEGIN-Rr	END-Lm	END-Lr	Lm-Rm	Lr-Rm	Lm-Rr	Lr-Rr
Unalign	589100	17299	901	7870	1000	639555	82413	24277	3305
BEGIN-Rmono	1978	339686	6397	0	0	0	0	0	0
BEGIN-Rreorder	186	13812	16032	0	0	0	0	0	0
END-Lmono	2258	0	0	364121	4251	0	0	0	0
END-Lreorde	699	0	0	4693	8269	1	0	0	0
Lmomo-Rmono	142777	1	0	0	0	4232113	105266	78692	3264
Lreorder-Rmono	96278	0	1	0	0	491989	323272	14635	6698
Lmomo-Rreorder	31118	0	0	0	0	380483	18144	198068	4335
Lreorder-Rreorder	12366	0	1	0	0	50121	25196	17008	22157
Recall	43.13%	97.59%	53.39%	98.24%	60.53%	92.77%	34.65%	31.33%	17.47%
Precision	67.19%	91.61%	68.71%	96.66%	61.16%	73.04%	58.32%	59.54%	55.73%

Table 4: RNN Confusion Matrix. Abbreviations: Lmono(Lm) Lreorder(Lr) Rmono(Rm) Rreorder(Rr)

6. Conclusion

In this paper, a novel tagging style reordering model has been proposed. By our modeling method, the reordering problem is converted into a sequence labeling problem so that the whole source sentence is taken into consideration for reordering decision. By adding an unaligned word tag, the unaligned word phenomenon is automatically implanted in the proposed model. Although the training phase of our model needs large computational costs, its usage for decoding is quite simple. In practice, we do not experience decoding memory increase nor speed slow down.

We choose CRFs and RNN to accomplish the sequence labeling task. The CRFs learning task takes huge amount of features and significant computational costs. Both ℓ^1 and ℓ^2 penalty are used in regularization. Hence the feature selection is automatically conducted. For test corpus, the token error rate is 25.75% and the sequence error rate is 69.39%. For

Systems	NIST02	NIST03	NIST04	NIST05	NIST08	avg.	Index
			BLEU scores				
baseline	33.60	34.29	35.73	32.15	26.34	-	1
baseline+CRFs	34.53	35.19	36.56‡	33.30‡	27.41‡	0.98	2
baseline+RNN	35.30‡	35.34‡	37.03‡	33.80‡	27.23‡	1.32	3
			TER scores				
baseline	61.36	60.48	59.12	60.94	65.17	-	4
baseline+CRFs	60.14‡	58.91‡	57.91‡	59.77‡	64.30‡	1.21	5
baseline+RNN	59.38‡	58.87‡	57.60‡	59.56‡	63.99‡	1.53	6

Table 6: Experimental results. ‡ means the value is better than its corresponding baseline with more than 95% confidence.

RNN training, we adopt the bidirectional RNN with LSTM. For test corpus, the token error rate is 27.31% and the sequence error rate is 77.00%.

We utilize our model as soft constraints in the decoder. Experimental results show that our model is stable and improves the baseline system by 0.98 BLEU and 1.21 TER (trained by CRFs) and 1.32 BLEU and 1.53 TER (trained by RNN). Most of the scores are better than their corresponding baseline values with more than 95% confidence.

The two main contributions are: propose the tagging-style reordering model and prove its ability to improve the translation quality; compare two sequence labeling techniques CRFs and RNN. To our best knowledge, this is the first experimental comparison of the CRFs and RNN.

7. Acknowledgements

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. 4911028154.0. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

8. References

[1] K. Knight, "Decoding complexity in word-replacement translation models," *Comput. Linguist.*, vol. 25, no. 4, pp. 607–615, 1999.

[2] D. Wu, "Stochastic inversion transduction grammars with application to segmentation, bracketing, and alignment of parallel corpora," in *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1328–1335.

[3] A. Berger, P. F. Brown, S. A. Pietra, V. J. Pietra, A. S. Kehler, and R. L. Mercer, "Language translation apparatus and method of using context-based translation models," United States Patent 5510981, April 1996.

[4] C. Wang, M. Collins, and P. Koehn, "Chinese syntactic reordering for statistical machine translation," in *Proceedings of the EMNLP/CoNLL-07*, Prague, Czech Republic, June 2007, pp. 737–745.

[5] Y. Zhang, R. Zens, and H. Ney, "Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation," in *Proceedings of the NAACL-HLT-07/AMTA Workshop on Syntax and Structure in Statistical Translation*, Morristown, NJ, USA, April 2007, pp. 1–8.

[6] R. Zens and H. Ney, "Discriminative reordering models for statistical machine translation," in *Proceedings of the Workshop on Statistical Machine Translation at HLT-NAACL-06*, New York City, NY, June 2006, pp. 55–63.

[7] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. R. Fonollosa, and M. R. Costa-Jussà, "*N*-gram-based machine translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[8] C. Cherry, "Cohesive phrase-based decoding for statistical machine translation," in *Proceedings of ACL-08: HLT*, Columbus, Ohio, June 2008, pp. 72–80.

[9] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev, "A smorgasbord of features for statistical machine translation," in *Proceedings of NAACL-HLT-04*, Boston, Massachusetts, USA, May 2004, pp. 161–168.

[10] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proceedings of ACL-02*, Philadelphia, Pennsylvania, USA, July 2002, pp. 295–302.

[11] F. J. Och, C. Tillmann, and H. Ney, "Improved alignment models for statistical machine translation," in *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP99)*, University of Maryland, College Park, MD, USA, June 1999, pp. 20–28.

[12] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *German Conference on Artificial Intelligence*. Springer Verlag, 2002, pp. 18–32.

[13] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 48–54.

[14] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[15] C. Sutton and A. Mccallum, *Introduction to Conditional Random Fields for Relational Learning*. MIT Press, 2006.

[16] T. Lavergne, O. Cappé, and F. Yvon, "Practical very large scale CRFs," in *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, July 2010, pp. 504–513.

[17] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[18] M. I. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," *IEEE Computer Society Neural Networks Technology Series*, pp. 112–127, 1990.

[19] K. J. Lang, A. H. Waibel, and G. E. Hinton, "A time-delay neural network architecture for isolated word recognition." *Neural networks*, vol. 3, no. 1, pp. 23–43, 1990.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: explorations in the microstructure of cognition, vol. 1," D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986, ch. Learning internal representations by error propagation, pp. 318–362.

[21] C. M. Bishop, *Neural Networks for Pattern Recognition*, 1st ed. Oxford University Press, USA, Jan. 1996.

[22] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *Trans. Sig. Proc.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1109/78.650093

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[24] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult." *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[25] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *Proceedings of ICSLP-02*, Denver, Colorado,USA, September 2002, pp. 901–904.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," no. RC22176 (W0109-022), September 2001.

[27] M. Snover, B. Dorr, R. Schwartz, J. Makhoul, L. Micciulla, and R. Weischedel, "A study of translation error rate with targeted human annotation," University of Maryland, College Park, MD, Tech. Rep. LAMP-TR-126, CS-TR-4755, UMIACS-TR-2005-58, 2005.

[28] A. Graves, "Rnnlib: A recurrent neural network library for sequence learning problems," http://sourceforge.net/projects/rnnl/.

[29] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Math. Program.*, vol. 45, no. 3, pp. 503–528, Dec. 1989.

[30] A. Graves, "Supervised Sequence Labelling with Recurrent Neural Networks," Ph.D. dissertation, 2008.

[31] P. Koehn, "Statistical significance tests for machine translation evaluation," Barcelona, Spain, July 2004, pp. 388–395.

Sparse Lexicalised Features and Topic Adaptation for SMT

Eva Hasler, Barry Haddow, Philipp Koehn

University of Edinburgh
Edinburgh, United Kingdom
e.hasler@ed.ac.uk, {pkoehn,bhaddow}@inf.ed.ac.uk

Abstract

We present a new approach to domain adaptation for SMT that enriches standard phrase-based models with lexicalised word and phrase pair features to help the model select appropriate translations for the target domain (TED talks). In addition, we show how source-side sentence-level topics can be incorporated to make the features differentiate between more fine-grained topics within the target domain (topic adaptation). We compare tuning our sparse features on a development set versus on the entire in-domain corpus and introduce a new method of porting them to larger mixed-domain models. Experimental results show that our features improve performance over a MIRA baseline and that in some cases we can get additional improvements with topic features. We evaluate our methods on two language pairs, English-French and German-English, showing promising results.

1. Introduction

In the field of statistical machine translation, domain adaptation is the task of tuning machine translation systems to produce optimal translations for a particular target domain by making the best possible use of the training data, given that we have, usually, a small amount of in-domain data and a larger amount of out-of-domain data. Most approaches to domain adaptation concentrate on either the language model or the translation model and ways to get more appropriate estimates for the respective probability distributions. Other approaches focus on acquiring more in-domain data as opposed to trying to make better use of existing training data.

In this paper, we focus on enhancing standard phrase-based machine translation systems with sparse features in order to bias our systems for the vocabulary and style of the target domain, the TED talks domain. We explore and compare several discriminative training approaches to include sparse features into small in-domain and larger mixed-domain systems. The idea is that sparse features can be added on top of baseline systems that are trained in the usual fashion, overlapping with existing features in the phrase table. This gives us flexibility to explore new feature sets which is particularly useful for training large systems from mixed-domain data. We show experimental results on data provided for the IWSLT 2012 shared task.

2. Training sparse features for domain adaptation

Adding sparse, lexicalised features to existing translation systems trained on in-domain or mixed-domain data is one way to bias translation systems towards translating a particular domain, in our case the TED talks domain. Our features are trained with the MIRA algorithm which is explained briefly in the following subsection. We compare the standard approach, e.g. tuning on a rather small development set, to the less common jackknife approach, details of which are given in subsection 2.4.

2.1. Training features with MIRA

Recently, the Margin Infused Relaxed Algorithm (MIRA) [6] has gained popularity as an alternative training method to Minimum Error Rate Training (MERT) [16], because it can deal with an arbitrary number of features. MIRA is an online large margin algorithm that enforces a margin between different translations of the same sentence. This margin can be tied to a loss function like BLEU [17] or another quality measure. Given that we can provide the learning algorithm with good oracle translations, the model learns to score hypothesis translations with higher BLEU scores better than translations with lower BLEU scores. MIRA updates the feature weights of a translation model by iterating though the training data, decoding one sentence at a time and performing weight updates for pairs of good and bad translation examples. Details about MIRA can be found in [12] or [3], for example.

We use a slightly modified version of the implementation described in [12] that selects hope and fear translations from a 30best list instead of running the decoder with hope and fear objectives. This has the effect that there is no need for dynamically computed sentence-level BLEU scores anymore because real sentence-level BLEU scores can be computed on the 30best list. [5] mentions that certain features, e.g. the language model, are very sensitive to larger weight changes and so we introduce a separate learning rate for core features (translation model, language model, word penalty and so on) in order to reduce fluctuations and keep MIRA training more stable. This learning rate is independent of the C parameter in the objective function solved by MIRA and is set to 0.1 for core features (1.0 for sparse features).

2.2. Feature sets

We experiment with two classes of indicator features, sparse phrase pair features and sparse word pair (or word translation) features. Word pair features capture translations of single source words to single target words, whereas phrase pair features capture translations of several words on the source side into several words on the target side. The class of phrase pair features depends on the decoder segmentation and can also include phrase pairs of length 1 on each side if such a phrase pair was extracted from the training data. Word pair features on the other hand depend on word alignment information and only contain word pairs that were connected by an alignment point in the training data.

Both of these feature classes were also extended with topic information acquired from topic models trained on the source side of the training corpus. The topic information is integrated as a source side trigger for a particular word or phrase pair, given a topic. Details about how these topic models were trained are given in section 2.3. Table 1 shows a pair of source sentence and hypothesis translation taken from a MIRA training run and examples of the features extracted from that sentence pair. The feature values indicate the number of times a feature occurred in a given sentence pair. The features in the first column capture general word or phrase translations while the features in the second column capture translations given a particular topic (here: topic 10). The features without topic information simply indicate whether a particular word or phrase translation should be favoured or avoided by the decoder, depending on whether they receive positive or negative weights during training. The features with topic information are triggered by the topic of the source sentence, that is, for a particular source sentence to be translated, only the features that were seen with the topic of that sentence will fire.

The TED domain is an interesting domain to try out these classes of features, because we can distinguish two different adaptation tasks: (1) adapting to the general vocabulary of TED talks as opposed to the vocabulary of out-of-domain texts (details in the experiments section), and (2) adapting to the vocabulary of subsets of TED talks that can be grouped into more fine-grained topics which we try to capture with topic models.

2.3. Training topic models

The topic models used for building enhanced word pair and phrase pair features are Hidden Topic Markov Models (HTMMs) [11] and were trained with a freely available toolkit. While topic modelling approaches like Latent Dirichlet Allocation assume that each word in a text was generated by a hidden topic and the topics of all words are assumed to be independent, HTMMs model the topics of words in a document as a Markov chain where all words in a sentence are assigned the same topic. This makes intuitively more sense than assigning several different topics within the same sentence and [11] show that HTMMs also yield lower model perplexity than LDA. The former characteristic makes HTMMs particularly suitable for our purpose. We are guaranteed that each word in a source phrase is assigned the same topic and therefore we do not have to figure out how to assign phrase topics given word topics.

Table 1: *Examples of en-fr word pair (wp) and phrase pair (pp) features, with and without topic information. Brackets indicate the phrase segmentation during decoding.*

input (topic 10): "[a language] [is a] [flash of] [the human spirit] [.]"	
hypothesis: "[une langue] [est une] [flash de] [l' esprit humain] [.] "	
reference: "une langue est une étincelle de l' esprit humain ."	
wp_a∼une=2	wp_10_a∼une=2
wp_language∼langue=1	wp_10_language∼langue=1
wp_is∼est=1	wp_10_is∼est=1
wp_flash∼ flash=1	wp_10_flash∼ flash=1
wp_of∼de=1	wp_10_of∼de=1
...	...
pp_a,language∼une,langue=1	pp_10_a,language∼une,langue=1
pp_is,a∼est,une=1	pp_10_is,a∼est,une=1
pp_flash,of∼flash,de=1	pp_10_flash,of∼flash,de=1
...	...

HTMMs compute $P(z_n, \psi_n | d, w_{i=1}, .., w_N)$ for each sentence, where z_n is the topic of sentence n, d is the document and w_i are words in sentence n. ψ_n determines the topic transition between words and can be non-zero only at sentence boundaries. When $\psi_n = 0$, the topic is identical to the previous topic, when $\psi_n = 1$, a new topic is drawn from a distribution θ_d. Once the sentence topic has been selected, all w_i are generated according to a multinomial distribution with topic-specific parameters. In order to assign topics to sentences in our training data, we derive a sentence topic distribution

$$\begin{aligned} P(topic|sentence) &= P(z_n|d, w_{i=1}, .., w_N) \\ &= P(z_n, \psi_n = 0|d, w_{i=1}, .., w_N) \\ &+ P(z_n, \psi_n = 1|d, w_{i=1}, .., w_N) \end{aligned} \quad (1)$$

We noticed that the distributions $P(topic|sentence)$ were quite peaked in most cases and therefore we tried to use a more compact representation. First, we selected the most likely topic according to the topic distribution and treated this as ground truth, ignoring all other possible topics. Alternatively, we selected the two most likely topics along with their probabilities, ignoring the second most likely topics with a probability lower than 30%. The topic probabilies were then used instead of the binary feature values in order to integrate the confidence of the topic model in its assigments. Experimental results were slightly better for the first representation without probabilities and therefore we chose this simpler presentation in all reported experiments.

In order to improve the quality of the topic models, we used stop word lists and lists of salient TED talk terms to clean the in-domain data before training the topic models.

Table 2: *Sample English and German HTMM topics and their interpretation in quotes.*

"cancer"	"ocean"	"body"	"universe"
cancer	water	brain	universe
cells	ice	human	space
body	surface	neurons	Earth
heart	Earth	system	light
blood	Mars	mind	stars
Krebs	Wasser	DNA	Erde
Patienten	Meer	Leben	Universum
Gehirn	Menschen	Licht	Planeten
Zellen	Ozean	Bakterien	Leben
Körper	Tiere	Menschen	Sonne

Table 3: *Topic assignment to training sentences with topic probabilities in brackets.*

"universe" (0.41)	"And physicists came and started using it sometime in the 1980s."
"universe" (0.47)	"And the miners in the early part of the last century worked, literally, in candle-light."
"ocean" (0.71)	"And today, you would see this inside the mine, half a mile underground."
"ocean"/"universe" (0.51/0.49)	"This is one of the largest underground labs in the world."
"universe" (0.99)	"And, among other things, they're looking for dark matter."
"universe" (1.00)	"There is another way to search for dark matter, which is indirectly."
"universe" (1.00)	"If dark matter exists in our universe, in our galaxy, then these particles should be smashing together..."

All TED talks come with a small set of keywords (∼300 in total) describing the content of the talk. The idea was to use the information contained in these keywords to select salient terms that frequently cooccur with the keywords. We first computed tf-idf for all words in each talk, normalised by the number of words in the talk. We then summed up the normalised tf-idf counts for each keyword, i.e. the counts of words in all documents associated with a particular keyword, and selected the top 100 terms for each keyword. This yielded ∼10500 terms for English and ∼11700 terms for German.

In cases where this filtering yielded empty sentences in the in-domain data (sentences with no salient terms), the topic information was replaced by "unk". We ran the topic training for 100 iterations and trained 30 topics over training, development and test sets. We modified the Moses decoder to accept topic information as XML mark-up and annotated all data with sentence-wise topics (and optionally the respective probabilities). Table 2 gives some examples of topics and their 5 most frequent terms for English and German as a source language, as we use topic triggers associated with the source sentence for our sparse features. The topic models represent topics as integers but here we have added labels to indicate the nature of the topics and we selected topics that map across the two languages. In general, the topics do not neccessarily map to equivalent topics in another language.

Table 3 shows a sequence of training sentences and their most likely topic (as well as the second most likely topic if applicable). We can see that for some of the sentences, the model assigns what we have labelled the "universe" topic with high probability while for others it is less certain or makes a transition to the "ocean" topic.

2.4. Jackknife setup

Training sparse features always involves a risk of overfitting on the tuning set, especially with highly lexicalized features that might occur only once in the tuning set. Therefore, training sparse features on the entire training set used to estimate the phrase table is expected to be more reliable. For discriminative training methods this means that the training set needs to be translated in order to infer feature values and compute BLEU scores. However, translating the same data that was used to train the translation system would obviously cause overfitting as well, thus the system needs to be adjusted to prevent this. In order to translate the whole training data without bias, we apply the jackknife method to split up the training data into n=10 folds. We create n subsets of the training data containing n-1 folds and leaving out one fold at a time. These subsets serve as training data for n systems that can be used to translate the respective left-out fold.

To use the jackknife systems for MIRA training, we modified the algorithm to accept n sets of decoder configuration files, input files and reference files. Instead of running n instances of the same translation system in parallel, we run n jackknife systems in parallel and average their weight vectors several times per epoch.

When applying the jackknife method to the TED in-domain data, we noticed a problem with this approach. Usually it would be good practice to create folds in a way that the resulting subsets of training data are as uniform as possible in terms of vocabulary to minimize the performance hit caused by the missing fold. However, the vocabulary of the TED data turned out to be quite repetitive within sentences belonging to the same talk. Thus, splitting up the data uniformly had the effect that each of the n systems had a certain amount of phrasal overlap with its left-out fold. This resulted in a preference for longer phrases, overly long translations on the test set and decreasing performance during MIRA training.

We were able to overcome the overfitting effect of line-wise data splits by splitting the data in a roughly talk-wise fashion instead. That is, the first x =corpus size/n lines were assigned to fold 1, the following x lines to fold 2 and so on. This way the folds were still the same size, but the training

data was much less likely to overlap with the left-out fold. The results on a held-out set during MIRA training (in particular the length penalty and overall length ratio) showed that this helped to prevent overfitting on the left-out fold.

3. Integrating features into mixed-domain models (retuning)

Tuning sparse features on top of large translation models can be time and memory-consuming. Especially the jackknife approach would cause immense overhead to tune with the mixed-domain data because we would need to train *n* different phrase tables that all include most of the in-domain data and all of the out-of-domain data[1]. Therefore, we wanted to investigate whether there is an alternative way of tuning our features on all of the in-domain data while also making use of the out-of-domain data. Tuning with the in-domain models allows for more flexibility in the training setup because the data set is relatively small. Since our goal is to translate documents of the TED talks domain, we assume that tuning sparse features only on the TED domain should provide the model with enough information to select the appropriate vocabulary. Hence we propose to port the tuned features from the in-domain models to the mixed-domain models. The advantage of this method is that features can be tuned on all the in-domain training data (jackknife) or in other ways that are feasible on a smaller in-domain model but might not scale well on a large mixed-domain model.

However, porting tuned feature weights from one model to another is not straightforward because the scaling of the core features is likely to be different. Therefore, to bring the sparse feature weights on the right scale to integrate them into the mixed-domain model, we perform a retuning step with MIRA. We take the sparse features tuned with the jackknife method and combine them into one aggregated meta-feature with a single weight. During decoding, the weight of the meta-feature is applied to all sparse features belonging to the same class (word pair or phrase pair features). In the retuning step, the core weights of the mixed-domain model are tuned together with the meta-feature weight.

An overview of our tuning schemes is given in figure 1. The training step denotes the entire training pipeline yielding the baseline models. Direct tuning refers to tuning with MIRA on a small development set and applies to both kinds of baseline models, while jackknife tuning only applies to in-domain models and retuning only to mixed-domain models.

4. Experiments

We evaluate our training schemes on English-French (en-fr) and German-English (de-en) translation systems trained on the data sets as advised for the IWSLT2012 TED task. As in-domain data we used the TED talks from the WIT[3] website[2] [2]. As out-of-domain data we used the Europarl, News Commentary and MultiUN [8] corpora and for en-fr also the 10^9 corpus taken from the WMT2012 release. An overview of all training data as well as development and test data is given in table 4 (sentence counts).

With this data we trained in-domain and mixed-domain baselines for both language pairs. For the mixed-domain baselines (trained on data from all domains), we used simple concatenations of all parallel training data, but trained separate language models for each domain and linearly interpolated them on the development set. All systems are phrase-based systems trained with the Moses toolkit [13]. Compound splitting and syntactic pre-reordering was applied to all German data. As optimizers we used MERT as implemented in the current version of Moses and a modified version of the MIRA implementation in Moses as described in section 2.1. We provide baseline results for tuning with both MERT and MIRA for comparison, though our model extensions are evaluated with respect to the MIRA baselines. Reported BLEU scores were computed using the mteval-v11b.pl script.

Figure 1: *In-domain (IN) and mixed-domain (IN+OUT) models with three tuning schemes for tuning sparse feature weights: direct tuning, jackknife tuning and retuning.*

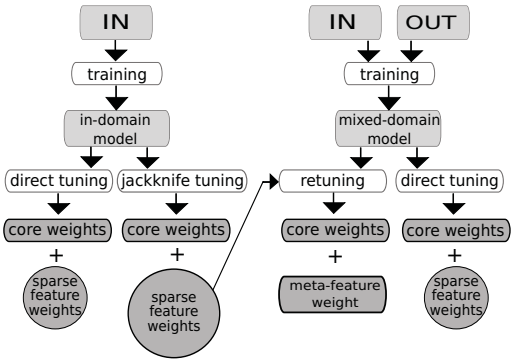


Table 4: *Sentence counts of in-domain (TED talks) and out-of-domain training data used in our systems.*

	en-fr	de-en
TED talks	140K (1029 talks)	130K (976 talks)
Europarl v7	2M	1.9M
News Commentary v7	137K	159K
MultiUN	12.9M	161K
10^9 corpus	22.5M	n/a
total	35.9M	2.3M
TED talks (monoling.)	143K	142K
dev2010	934 (8 talks)	900 (8 talks)
test2010.part1	898 (5 talks)	665 (5 talks)
test2010.part2	766 (6 talks)	900 (6 talks)

[1]Training the mixed-domain system for the en-fr language pair took more than a week.

[2]https://wit3.fbk.eu/mt.php?release=2012-03

All experiments except the jackknife experiments used the TED dev2010 set as development set (dev). The TED test2010 set was split into two parts, test2010.part1 and test2010.part2. For the in-domain experiments, one part was used to select the best weights found during MIRA training and the other part was used for evaluation, respectively. We refer to these sets as test1 and test2 to indicate which of the two parts was used as the test set. We note that test1 and test2 yield quite different BLEU scores for the baseline models. However, table 5 shows that the relative improvements achieved with MIRA are roughly proportional and thus we will report results on just one of the two sets for experiments on the mixed-domain baselines.

All MIRA experiments were initialized with the tuned weights of the MERT baselines. MIRA experiments on the dev set were run for 20 epochs, retuning experiments for 10 epochs and jackknife experiments on the entire training set for 2 epochs.

4.1. Results

We are evaluating the impact of our sparse features on the in-domain and mixed-domain systems. Tables 5 and 6 show the results on the in-domain system with BLEU scores reported on both parts of the test2010 set, using the respective other part as devtest set. Improvements over the MIRA baseline are marked in bold print and the relative changes are indicated in brackets. First we note that MIRA training improves the MERT baseline performance for the en-fr system by 0.8 BLEU on both test sets, but decreases performance for the de-en system by 0.3 BLEU. We believe that this divergence has to do with the changes in length ratio after MIRA training, as shown in table 7. For en-fr, translations get longer during MIRA training while for de-en they get shorter, incurring an increased brevity penalty according to the BLEU score.

Since MIRA has quite a different impact on the translation performance with the core features (translation model, reordering model, language model, word penalty, phrase penalty), we focus on the impact of sparse features with respect to the MIRA baselines. For en-fr, we observe that all sparse feature setups beat the MERT baseline and most of them beat the MIRA baseline. For the MIRA experiments on the dev set we notice that phrase pair features seem to perform better than word pair features on both test sets and sparse features with topic triggers seem to do better than sparse features without topic information. The results of the MIRA experiments using the jackknife method are in almost all cases better than the results trained on the small dev set. We get an increase of up to 1.3/0.2 BLEU (en-fr/de-en) over the MERT baseline and up to 0.5/0.7 BLEU (en-fr/de-en) over the MIRA baselines. This shows that the jackknife method is better suited to train sparse features than training on a small dev set. We still observe slightly better results for phrase pair features than for word pair features with the en-fr models, even though this observation is less conclusive than on the dev data.

Table 5: *In-domain baselines (IN) and results for sparse feature training on en-fr in-domain model, training on a development set (dev) and on all training data (jackknife).*

en-fr	BLEU(test1)	BLEU(test2)
MERT(dev) IN	*28.6*	*30.9*
MIRA(dev) IN	29.4	31.7
MIRA(dev)		
+ wp	29.2 (-0.2)	31.6 (-0.1)
+ wp + topics	**29.5** (+0.1)	**31.8** (+0.1)
+ pp	**29.6** (+0.2)	31.7 (+0.0)
+ pp + topics	**29.6** (+0.2)	**31.9** (+0.2)
MIRA(jackknife)		
+ wp	**29.7** (+0.3)	**32.2** (+0.5)
+ wp + topics	**29.5** (+0.1)	**32.1** (+0.4)
+ pp	**29.9** (+0.5)	**32.2** (+0.5)
+ pp + topics	**29.6** (+0.2)	**32.0** (+0.4)

Table 6: *In-domain baselines (IN) and results for sparse feature training on de-en in-domain model, training on a development set (dev) and on all training data (jackknife).*

de-en	BLEU(test1)	BLEU(test2)
MERT(dev) IN	*26.6*	*29.9*
MIRA(dev) IN	26.3	29.6
MIRA(dev)		
+ wp	**26.7** (+0.4)	**29.8** (+0.2)
+ wp + topics	**26.6** (+0.3)	**29.7** (+0.1)
+ pp	**26.5** (+0.2)	**29.7** (+0.1)
+ pp + topics	**26.4** (+0.1)	**29.8** (+0.2)
MIRA(jackknife)		
+ wp	**27.0** (+0.7)	**30.1** (+0.5)
+ wp + topics	**26.4** (+0.1)	**29.7** (+0.1)
+ pp	**26.8** (+0.5)	**30.0** (+0.4)
+ pp + topics	**26.4** (+0.1)	**29.8** (+0.2)

Tables 8 and 9 show results on the mixed-domain models, where we observe a similar divergence in performance between the MERT and MIRA baselines as on the in-domain models: a plus of 1.1 BLEU for en-fr and a minus of 0.4 BLEU for de-en. The first block of results refers to MIRA training on the dev2010 set as for the in-domain models (direct tuning), while the second block results from the retuning setup described in section 3 (retuning). The direct approach gains up to 0.5 BLEU for en-fr and up to 0.1 BLEU for de-en over the MIRA baselines, retuning with MIRA and jackknife features gains up to 0.5 BLEU for en-fr and up to 0.4 BLEU for de-en over the MIRA baselines. This is another indication that sparse features trained with the jackknife method can leverage information from the in-domain training data to help the model select appropriate words and phrases for the target domain. In some cases we can observe that topic

Table 7: *Changes to the length ratio (hypotheses/reference, in brackets) between MERT and MIRA tuning, indicated by (+) and (-).*

		BLEU(test1)	BLEU(test2)
en-fr	MERT(dev) IN	28.6 (0.969)	30.9 (0.963)
	MIRA(dev) IN	29.4 (0.987) (+)	31.7 (0.982) (+)
de-en	MERT(dev) IN	26.6 (0.987)	29.9 (1.001)
	MIRA(dev) IN	26.3 (0.955) (-)	29.6 (0.969) (-)

Table 8: *Mixed-domain baselines (IN+OUT) and results for sparse feature training on en-fr mixed-domain model: direct sparse feature tuning and retuning with MIRA using jackknife-trained features.*

en-fr	BLEU(test1)
MERT(dev) IN+OUT	*30.0*
MIRA(dev) IN+OUT	31.1
MIRA(dev), direct tuning	
+ wp	**31.6** (+0.5)
+ wp + topics	**31.4** (+0.3)
+ pp	**31.4** (+0.3)
+ pp + topics	**31.5** (+0.4)
MIRA(dev), retuning	
+ wp	**31.6** (+0.5)
+ wp + topics	31.1 (+0.0)
+ pp	**31.5** (+0.4)
+ pp + topics	**31.3** (+0.2)

features improve over simple features, even though they perform weaker in more of the cases. We suspect that sparsity issues need to be addressed to benefit more from these features. In general, the results show that features trained only on in-domain models can help to improve performance of much larger mixed-domain models. While for the in-domain models the results on both language pairs are similar w.r.t. the MIRA baselines, the results on mixed-domain models are clearly better for en-fr which can be considered an easier language pair for translation than de-en.

The feature sets ranged in size between around 5K-15K when training on a dev set and 60K-600K when training on all training data, depending on the particular feature type.

4.2. Topic features

For the en-fr in-domain systems trained on dev data, we see an improvement of topic features over simple sparse features. That these effects are not stronger might be due to the quite diverging distributions of topics across dev, devtest and test sets (see figure 2[3]). For example, the "universe" topic (topic 29) appears quite frequently in the training and dev data, but only twice in test2 and never in test1. For future experiments with sentence-level topic features it should be ensured that

[3]Training data counts were between 2252 and 7170 sentences per topic.

Table 9: *Mixed-domain baselines (IN+OUT) and results for sparse feature training on de-en mixed-domain model: direct sparse feature tuning and retuning with MIRA using jackknife-trained features.*

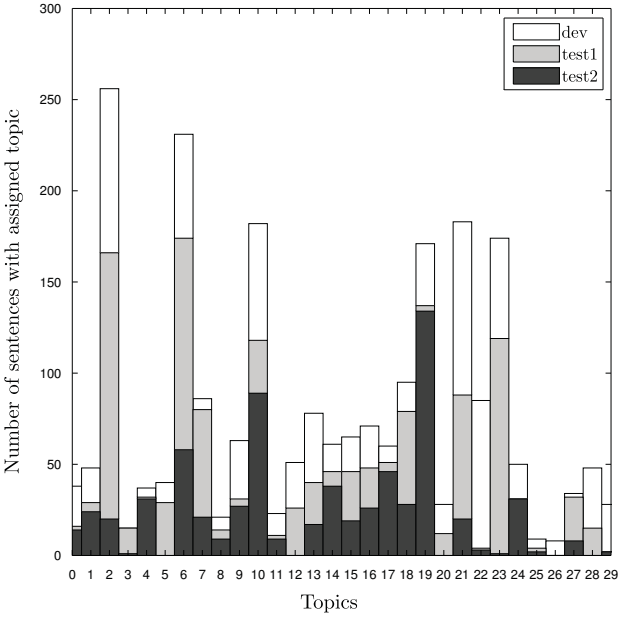
de-en	BLEU(test1)
MERT(dev) IN+OUT	*27.2*
MIRA(dev) IN+OUT	26.8
MIRA(dev), direct tuning	
+ wp	**26.9** (+0.1)
+ wp + topics	**26.9** (+0.1)
+ pp	**26.9** (+0.1)
+ pp + topics	26.7 (-0.1)
MIRA(dev), retuning	
+ wp	**27.1** (+0.3)
+ wp + topics	**27.2** (+0.4)
+ pp	**27.0** (+0.2)
+ pp + topics	**27.0** (+0.2)

topics are distributed more evenly across development sets.

Lexicalised features with topic triggers are even sparser than simple lexicalised features and therefore we would expect that they benefit particularly from jackknife training. However, our current results show the opposite tendency in that topic features seem to do worse than simple features under the jackknife setup. Table 10 gives an example of word pair features trained with the jackknife method, with and without topic information. It shows the features with the largest positive/negative weights (those with the highest discriminative power learned by the model) for translating the English source word "matter". Both models have learned that "matière" is the most appropriate French translation for the English word "matter". Both models penalize some translations of the other word sense like the French word "important". However, the model without topic information considers "importe" an almost equally likely translation, while the model with topic information penalizes all translations that do not preserve the physical word sense (as in "dark matter"). As mentioned above, the "universe" topic did not appear at all in test1, so the impact of features related to this topic has not been measured in the evaluation.

Table 11 shows jackknife-trained features for the source word "language". While with simple word pair features the most likely translation is "langage" (mode of speaking), the topic features express translation preferences according to the source topic. For example, given the "science" topic, the most likely translation is "langage", but given the "school" topic, the most likely translation is "langue". However, in table 1 we see that the input sentence is labelled with topic 10 ("science") but "language" is translated to "langue" in the reference translation. Thus, given the topic labelling the exptected translation with topic features would not match the reference translation, which is something that should be taken into account.

Figure 2: *Distribution of topics in dev, test1, test2.*



5. Related work

The domain adaptation literature can be broadly grouped into approaches adapting the language model and approaches adapting the translation model. Among the latter there has been work on mixture modeling of domain-specific phrase tables [9] and discriminative instance weighting [14] [10]. In similar spirit, [1] introduced a corpus-filtering technique that computes a bilingual cross-entropy difference to determine how similar a sentence pair is to an in-domain corpus and how dissimilar from a general-domain corpus. There has also been previous work on translation model adaptation using topics models. [19] employ HTMMs to train source-side topic models from monolingual in-domain data and the source side of parallel out-of-domain data. Phrase pairs are conditioned on in-domain topics via a mapping from in-domain to out-of-domain topics. Our approach is different in that we use parallel in-domain data and therefore do not need a mapping step. [7] extend previous work by [4] on lexical weighting conditioned on data provenance. They enhance lexical weighting features with topic model information to train separate word translation tables for every domain which can then be used to bias phrase selection based on source topics.

MIRA has been proposed for tuning machine translation systems with large features sets, for example by [20] and [3]. Recent work that compares tuning on a small development set versus tuning on the entire training data has been presented in [18]. The idea of using source triggers to condition word translation is somewhat related to the trigger-based lexicon models of [15], though they use context words as additional triggers and train their features with the EM algorithm.

Table 10: *Examples of en-fr jackknife-trained word pair features, with and without topic information (topic 29: "universe").*

sparse feature	feature weight
wp_matter∼matière	0.00170
wp_matter∼importe	0.00107
wp_matter∼important	-0.00037
wp_matter∼comptent	-0.00188
wp_29_matter∼matière	0.00431
wp_29_matter∼importent	-1.42913e-05
wp_29_matter∼importe	-0.00134
wp_29_matter∼important	-0.00172

Table 11: *Examples of en-fr jackknife-trained word pair features, with and without topic information (topic 10: "science", topic 27: "school").*

sparse feature	feature weight
wt_language∼langage	0.00444
wt_language∼langue	-0.00434
wt_10_language∼langage	0.01088
wt_10_language∼langue	-0.01071
wt_27_language∼langue	0.00792
wt_27_language∼langage	-0.00742

6. Conclusion

We presented a novel way of training lexicalised features for a domain adaptation setting by adding sparse word pair and phrase pair features to in-domain and mixed-domain models. In addition, we suggested a method of using topic information derived from HTMMs trained on the source language to condition the translation of words or phrases on the sentence topic. This was shown to yield improvements over simple sparse features on English-French in-domain models. We experimented with the jackknife method to use the entire in-domain data for feature training and showed BLEU score improvements for both language pairs. Finally, we introduced a retuning method for mixed-domain models that allows us to adapt features trained on the entire in-domain data to the mixed-domain models.

In the future, we would like to test our methods on hierarchical phrase-based or syntactic models. Other work in this field suggests that discriminative training yields larger gains with those types of models than with purely phrase-based models, so this would be an interesting comparison. We would also like to address the evaluation of topic features, which we believe requires a more controlled setting. Induced topics should be distributed more evenly across data sets and the quality of sentence topic labels should be taken into account.

7. References

[1] Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo In-Domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

[2] Cettolo, M., Girardi, C., and Federico, M. (2012). Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of EAMT*, pages 261–268, Trento, Italy.

[3] Chiang, D., Knight, K., and Wang, W. (2009). 11,001 new features for statistical machine translation. In *Proceedings of HLT: The 2009 Annual Conference of the NACL*, Stroudsburg, PA, USA. Association for Computational Linguistics.

[4] Chiang, D., DeNeefe, S., and Pust, M. (2011). Two easy improvements to lexical weighting. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies*, pages 455–460, Stroudsburg, PA, USA. Association for Computational Linguistics.

[5] Chiang, D. (2012). Hope and fear for discriminative training of statistical translation models. In *J. Machine Learning Research 13*, pages 1159–1187.

[6] Crammer, K. and Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3(4-5):951–991.

[7] Eidelman, V., Boyd-Graber, J., and Resnik, P. (2012). Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju Island, Korea. Association for Computational Linguistics.

[8] Eisele, A. and Chen, Y. (2010). Multiun: A multilingual corpus from united nation documents. In *LREC'10*.

[9] Foster, G. and Kuhn, R. (2007). Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Stroudsburg, PA, USA. Association for Computational Linguistics.

[10] Foster, G., Goutte, C., and Kuhn, R. (2010). Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Stroudsburg, PA, USA. Association for Computational Linguistics.

[11] Gruber, A., Rosen-Zvi, M., and Weiss, Y. (2007). Hidden Topic Markov Models. In *Journal of Machine Learnin Research, pp. 163-170*.

[12] Hasler, E., Haddow, B., and Koehn, P. (2011). Margin Infused Relaxed Algorithm for Moses. In *The Prague Bulletin of Mathematical Linguistics No. 96, 2011, pp. 69-78*, Prague.

[13] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *ACL 2007: proceedings of demo and poster sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

[14] Matsoukas, S., Rosti, A.-V. I., and Zhang, B. (2009). Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, pages 708–717, Stroudsburg, PA, USA. Association for Computational Linguistics.

[15] Mauser, A., Hasan, S., and Ney, H. (2009). Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Conference on Empirical Methods in Natural Language Processing*, pages 210–217, Singapore.

[16] Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *ACL-2003: 41st Annual meeting of the ACL*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

[17] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *ACL 02: Proceedings of the 40th Annual Meeting on ACL*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

[18] Simianer, P., Riezler, S., and Dyer, C. (2012). Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of the 50th Annual Meeting of the ACL*. Association for Computational Linguistics.

[19] Su, J., Wu, H., Wang, H., Chen, Y., Shi, X., Dong, H., and Liu, Q. (2012). Translation model adaptation for statistical machine translation with monolingual topic information. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju Island, Korea. Association for Computational Linguistics.

[20] Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H. (2007). Online large-margin training for statistical machine translation. In *Proceedings of EMNLP-CoNLL*, pages 764–773, Prague. Association for Computational Linguistics.

Spoken Language Translation Using Automatically Transcribed Text in Training

Stephan Peitz, Simon Wiesler, Markus Nußbaum-Thom, Hermann Ney

Human Language Technology and Pattern Recognition
Computer Science Department
RWTH Aachen University
Aachen, Germany
surname@cs.rwth-aachen.de

Abstract

In spoken language translation a machine translation system takes speech as input and translates it into another language. A standard machine translation system is trained on written language data and expects written language as input. In this paper we propose an approach to close the gap between the output of automatic speech recognition and the input of machine translation by training the translation system on automatically transcribed speech. In our experiments we show improvements of up to 0.9 BLEU points on the IWSLT 2012 English-to-French speech translation task.

1. Introduction

Spoken language translation (SLT) connects automatic speech recognition (ASR) and machine translation (MT) by translating recognized spoken language into a target language. In general, the speech translation process is divided into two separate parts. First, an ASR system provides an automatic transcription of spoken words. Then, the recognized words are translated by a machine translation system.

However, a difficult part of SLT is the interface between the ASR system and the MT system, due to the mismatch between the output of the ASR system and the expected input of the MT system. A standard MT system expects grammatically correct written language as input, because it is usually trained on written bilingual text with punctuation marks and case information. In contrast, the output of an ASR system is automatically transcribed natural speech containing recognition errors. Thus, the expected input of the MT system does not match the actual ASR output. Furthermore, ASR systems recognize sequences of words and do not provide punctuation marks or case information.

In this paper, we describe how the inconsistency between the ASR output and the SMT input is solved by replacing the source language data of a bilingual training corpus with automatically transcribed text. In a first approach, we keep the target language including case information and punctuation, because our goal is to improve the translation quality directly in an SLT task. On this new corpus, we train a statistical machine translation (SMT) system and use the system to translate the recognized speech into another language. Furthermore, case information and punctuation are restored during the translation process.

As a second approach, we built a bilingual training corpus with ASR output as source language data and the corresponding manual transcription with case information and punctuation marks as target language data. In the next step, an SMT system is trained on this corpus. Before translating the recognized speech into the target language, the ASR output is translated into manual transcription. Thus, the postprocessing of the ASR output is modelled as machine translation and we are able to translate the postprocessed ASR output with a standard translation system which is trained on written bilingual text.

On the English-French SLT task from IWSLT 2012, we show that our presented approaches improve the translation quality by up to 0.9 BLEU and 0.9 TER.

The paper is organized as follows. In the next section, we give a short overview of related work. In Section 3, we describe the usage of automatically transcribed text in the training process of an SMT system. Finally, we discuss the experimental results in Section 5, followed by a conclusion.

2. Related Work

In [1], an approach is presented to improve automatic call classification by training an SMT system on a bilingual corpus with ASR output as source language data and the corresponding manual transcribed text as target language data. The SMT system cleans the automatically transcribed text before the call classification. For further improvement of their framework, n-Best lists of the recognition were used. They performed experiments using IBM model 2 on live data collected from an enterprise call center and showed improvements in class classification accuracy.

A similar approach is presented in [2]. The authors describe a statistical transformation model which transforms spoken language into written language. Further, they compare the approach with a rule-based transformations model

in terms of precision and recall.

Another approach to transform spoken language into written language is described in [3]. A transduction model based on weighted finite-state transducers is trained on a parallel corpus of automatic transcription and manual transcription. In the experiments, Cantonese speech was transformed to standard written Chinese. The authors report improvements in Word Error Rate.

In [4], the use of automatically transcribed text as training data was described. The authors recognized audio recordings of parallel speech with an ASR system to create additional monolingual as well as bilingual corpora. They showed improvements by training a language, an acoustic and a translation model including the additional data.

In [5] different methods for punctuation prediction were analyzed. By using a translation system to translate from unpuncated to punctuated text the translation quality was improved on the IWSLT 2011 English-to-French Speech Translation of Talks task.

In our work, we revisit the idea of building a new corpus using automatically transcribed text as source language data. However, instead of cleaning the ASR output, we translate from ASR output into a target language directly, i.e. we replace the source language data of the bilingual corpus only. Furthermore, we do not want to collect additional monolingual or bilingual data, but the goal is to improve the quality of spoken language translation by using automatically transcripted text in the training process of a translation system. By training a phrase-based machine translation system on the new corpus, we want to close the gap between the output of an ASR system and the expected input of an SMT system. Moreover, we combine the original and the new corpus in various ways and extract *n*-Best lists from lattices to create a larger corpus. In addition, based on the idea of modeling punctuation prediction as machine translation, we train a translation system on a bilingual corpus with ASR output as source language data and corresponding manual transcription as target language data. This system translates from ASR output to manual transcription, i.e. the postprocessing of the ASR output is performed with a machine translation system. The main advantage of this method is that a standard text translation system can be used to translate the postprocessed ASR output.

3. Automatically Transcribed Text in Training

The starting point of this work is a data source which provides audio recordings, the corresponding manual transcriptions and the translation of these transcriptions. The online-available TED talks are such a kind of source [1]. This website provides manually transcribed and translated lecture-type talks presented at TED conferences. Furthermore, WIT^3 (Web Inventory of Transcribed and Translated Talks) redistributes the original content published by the TED website for the machine translation community [6]. The transcriptions and the translations are processed as parallel bilingual corpus to be able to train an SMT system. Further, development and test sets are provided.

In an SLT application, the development and test sets are automatically transcribed speech, which have to be translated into a target language. We assume in this work that the recognitions of the development and test sets do not contain punctuation and casing and the segmentation is given and corresponds to sentence-like units. With an SMT system, the automatically recognized speech is translated. Furthermore, the punctuation and the case information are restored during the translation process as described in [7]. In order to train such an SMT system, the punctuation and the case information of source language data in the bilingual training corpus are deleted to create a *pseudo ASR output*. In our work, we train an SMT system on a bilingual corpus with real ASR output instead of pseudo ASR output as source language data.

Due to the fact that WIT^3 also specifies the talks which were used to create the provided bilingual corpora, we are able to recognize the relevant audio recordings with our ASR system. About 1028 relevant talks are available on the web. In sum, roughly 250 hours of speech have to be recognized. Using the automatically transcribed recordings as source language data, we build a new bilingual corpus to train an SMT system for an SLT task.

3.1. Sentence Alignment

In general, an ASR system does not provide sentence-wise segmentation. However, a bilingual corpus, which is used to train an SMT system, consists of parallel sentences. In order to align automatic transcriptions sentence-wise to a given segmented manual transcription, we employ an automatic re-segmentation algorithm as described in [8].

The re-segmentation algorithm calculates the Levenshtein alignment between the recognition and its manual transcription. By backtracing the decisions of the edit distance algorithm, an alignment between a given sequence of words and an already sentence-wise segmented manual transcription as reference can be found. Thus, the sentence segmentation of the reference is transferred to the recognition. The re-segmentation algorithm is solved by dynamic programming.

As mentioned, WIT^3 provides manually transcribed text as well as the corresponding translation. First, we align our recognized training data to the manual transcription, which is already segmented on sentence level. In a second step, we replace the manual transcription with its translation. This results in a parallel bilingual corpus with ASR output as source language data and its translation with punctuation and case information as target language data.

Table 1 shows an example of an aligned bilingual sentence pair with various source language sentences. Starting with the given *manual transcription*, the *pseudo ASR output* is created by removing the full stop at the end of the sen-

[1] http://www.ted.com/

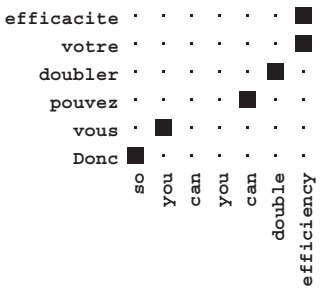


Figure 1: Partial alignment between *automatic transcription* and *manual translation* (Table 1).

tence and lowercasing the very first word. This transformed sentence is grammatically correct. In contrast, the *automatic transcription* of the sentence contains the repetition of the phrase "*you can*". Furthermore, "*60*" is transcribed as written number "*sixty*".

In Figure 1 a part of the corresponding alignment between the automatic transcription and its translation is shown. During the training procedure of the SMT system, phrase pairs such as

- ⟨you can you can, vous pouvez⟩
- ⟨sixty percent, 60 %⟩

are learned. With these phrase pairs, the SMT system is able to correct ASR output and to rewrite written numbers as digits during the translation process. Instead of translating the phrase "*you can*" twice, the SMT system has got the option to translate the phrase into "*vous pouvez*" directly, if such an error occurs in a given ASR output.

3.2. ASR Output Postprocessing

Another approach to make use of automatically transcribed text is to set up an SMT system which translates from ASR output into manually transcribed text. Therefore, we do not replace the manual translation with its translation as described before, but an SMT system is trained on a corpus with automatically transcribed text as source language data and manual transcriptions as target language data. Before the actual translation of the recognized speech, the SMT system performs a postprocessing of the ASR output. The ASR output is translated and during the translation process punctuation marks and case information are restored. Considering the bilingual sentence pair in Table 1 and the corresponding alignment in Figure 2, during the training of the SMT system phrase pairs such as

- ⟨you can you can, you can⟩
- ⟨sixty percent, 60 %⟩

are extracted. The main advantage is that the postprocessed ASR output can be used as input for an existing standard text translation system. Thus, we do not have to modify the training data of the translation system to translate ASR output.

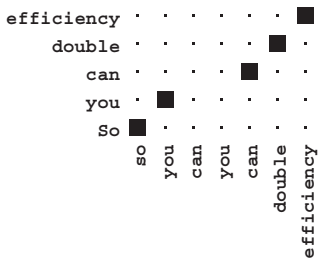


Figure 2: Partial alignment between *automatic transcription* and *manual transcription* (Table 1).

4. System Description

In this section, we describe our ASR and MT system, which are employed in this work. With the ASR system, we recognize the source language data of the new bilingual corpus as well as the development and test sets in a given SLT task. We train a MT system on the different corpora and combination to verify the impact of automatically transcribed text in the training. All setups are tuned on a development set and are compared on a test set.

4.1. ASR System

The ASR system is based on our English speech recognition system that we successfully applied in Quaero evaluations [9].

The recognizer is a generative statistical classifier that maps a sequence of acoustic observations x_1^T to a word sequence w_1^N via Bayes decision rule:

$$\hat{w}_1^N = \underset{w_1^N}{\operatorname{argmax}}\, p(w_1^N)^{\gamma} p(x_1^T | w_1^N)\,. \tag{1}$$

The prior probability $p(w_1^N)$ is the *language model*, $p(x_1^T | w_1^N)$ is the *acoustic model*, and γ is the *language model scale*.

In the acoustic feature extraction, the system computes Mel-frequency cepstral coefficients (MFCC) from the audio signal, which are transformed with a vocal tract length normalization (VTLN). In addition, a voicedness feature is computed. Acoustic context is incorporated by concatenating nine feature vectors in a sliding window. The resulting feature vector is reduced to 45 dimensions by means of a linear discriminant analysis (LDA). Furthermore, bottleneck features derived from a multilayer perceptron (MLP) are concatenated with the feature vector.

The acoustic model is based on hidden Markov models (HMMs) with Gaussian mixture models (GMMs) as emission probabilities. The GMM has a pooled, diagonal covariance matrix. It models 4500 generalized triphones which are derived by a hierarchical clustering procedure (CART). The parameters of the GMM are estimated with the expectation-maximization (EM) algorithm with a splitting procedure according to the maximum likelihood criterion.

Table 1: Example of a bilingual sentence pair. *pseudo ASR output* is created by removing punctuation and case information of the *manual transcription*. The *automatic transcription* was recognized with our ASR system and *manual translation* is the corresponding given translation.

Corpus	
manual transcription	So you can double efficiency with a 60 percent internal rate of return .
pseudo ASR output	so you can double efficiency with a 60 percent internal rate of return
automatic transcription	so you can you can double efficiency with a sixty percent internal rate of return
manual translation	Donc vous pouvez doubler votre efficacite nergtique avec un Taux de Rendement Interne de 60 % .

The language model is a Kneser-Ney smoothed 4-gram. Several language models are trained on different datasets. The final language model is obtained by linear interpolation. The vocabulary of the recognition lexicon is obtained by applying a count-cut-off on the language model data. Each word in the lexicon can have multiple pronunciations. Missing pronunciations are derived with a grapheme-to-phoneme tool.

The recognition is structured in three passes, In the first pass, a speaker independent model is used. The recognition result of the first pass is used for estimating feature transformations for speaker adaptation (CMLLR). The second pass uses the CMLLR transformed features. Finally, a confusion network decoding is performed on the word lattices obtained from the second pass.

Table 2: Acoustic training data of ASR system

Corpus	Amount of data [hours]
quaero-2011	268h
hub4+tdt4	393h
epps	102h

Table 3: Language model training data of ASR system

Corpus	Amount of data [running words]
Gigaword 4	2.6B
Ted	2.7M
Acoustic transcriptions	5M

The acoustic model of the ASR system is trained on 793 hours of transcribed acoustic data in total, see Table 2. The acoustic training data consists of American broadcast news data (hub4+tdt4), European parliament speeches (epps), and British broadcast conversations (quaero). The MLP is trained on the 268 hours of the quaero corpus only. We use 4500 triphone states and perform eight EM splits, resulting in a GMM with roughly 1.1 million mixture components.

The language model is trained on a large amount of news data (Gigaword), the transcriptions of the audio training data, and a small amount of in-domain data (ted), see Table 3. The recognition lexicon consists of 150k words.

4.2. MT System

The decoder of the phrase-based translation system which is used in this work is described in [10] and is part of RWTH's open-source SMT toolkit Jane 2.1 [2]. We use the standard set of models with phrase translation probabilities and lexical smoothing in both directions, word and phrase penalty, distance-based distortion model, a 4-gram target language model and three binary count features. The features $h_m(f_1^J, e_1^J)$ are combined in a weighted log-linear model to find the best translation $\hat{e}_1^{\hat{I}}$

$$\hat{e}_1^{\hat{I}} = \arg\max_{e_1^I} \sum_{m=1}^{M} \lambda_m h_m(f_1^J, e_1^J). \quad (2)$$

The weights are optimized using standard MERT [11] on 200-best lists with BLEU as optimization criterion.

5. Experimental Evaluation

The proposed approach was evaluated on the IWSLT 2012 English-to-French spoken language translation task based on the already mentioned TED talks. For the evaluation, WIT[3] provides in-domain bilingual training data based on manually transcribed text and its translation. The 1028 talks (around 250 hours of speech), which corresponds to the bilingual training data, were recognized with the described ASR system.

For the baseline model, we removed punctuation and case information of the source language to create pseudo ASR output (Table 7) as we assume that the source language as produced by the speech recognition system does not contain any punctuation marks or case information. Punctuation and case information are restored during the translation process. To indicate that an SMT system was trained on this corpus, we mark the setup with MANUAL-TRANSCRIPTION.

In Table 8, the data statistics for the bilingual corpus with ASR output as source language data are shown. The number of sentences and running words differs from the

[2] http://www-i6.informatik.rwth-aachen.de/jane/

original bilingual corpus in Table 7, because a small number of recordings were not accessible. In the following, setups based on this data are tagged with AUTOMATIC-TRANSCRIPTION.

As a first approach, we only consider the output of the ASR system based on a confusion network decoding on the word lattices obtained from the second pass. Setups trained on the corpus are marked with AUTOMATIC-TRANSCRIPTION (cn-decoding).

To extend the training corpus, we further extracted n-Best lists from the resulting lattices of the second pass. We hope that the MT system could gain by using more ASR output in training. For the extraction of the n-Best lists, we used the LATTICE-TOOL from the SRI toolkit [12]. The n-Best lists were sentence-aligned to the corresponding manual translation as described before. In our experiments, we chose $n = \{1, 10, 20\}$. Thus, the size of the corpus was multiplied by n. Setups using corpora based on n-Best lists are labelled with AUTOMATIC-TRANSCRIPTION (n-Best). Note that AUTOMATIC-TRANSCRIPTION (1-Best) differs from AUTOMATIC-TRANSCRIPTION (cn-decoding). In contrast to 1-Best decoding which extracts the maximum probability sentence from the search space, cn-decoding approximates the minimization of the expected WER and is closer to the theoretical WER optimal decision rule for ASR. Therefore cn-decoding in practice always performs better than 1-Best output.

For the spoken language translation task in the IWSLT 2012 evaluation campaign, ASR output is provided as development set and test set (Table 5). However, to be consistent with the recognized training data, we used our own recognitions of the development and test sets in all experiments (except for one of the baseline experiments). In Table 4, we compare the word error rate (WER) of the provided sets (IWSLT 2012) with our recognitions (RWTH). A lower WER indicates a better recognition quality. The data statistics for RWTH (cn-decoding) are shown in Table 6.

Table 4: Comparison of the development and test sets in terms of WER

	dev	**test**
IWSLT 2012	18.0	16.7
RWTH (pass 1)	20.0	18.4
RWTH (pass 2)	17.5	15.9
RWTH (cn-decoding)	17.3	15.7

For all experiments, we used a 4-gram language model with modified Kneser-Ney smoothing which was trained with the SRILM toolkit on the monolingual version of the in-domain bilingual training data and on the Europarl and News Commentary data. Further, GIZA++ [13] was employed to train word alignments for each setup.

Table 5: Data Statistics for the provided development and test set (IWSLT 2012)

	dev	**test**
Sentences	934	1 664
Running Words	17 755	27 754
Vocabulary	3 133	3 698

Table 6: Data Statistics for development and test set recognized by our ASR system (RWTH (cn-decoding))

	dev	**test**
Sentences	934	1 664
Running Words	17 804	27 514
Vocabulary	3 149	3 689

5.1. Phrase Table and Data Combination

In this work, we analyze three different approaches to combine both corpora AUTOMATIC-TRANSCRIPTION and MANUAL-TRANSCRIPTION. We hope to further improve the translation quality by augmenting our baseline system with the original data. Due to the fact, that a small amount of the recordings were not accessible or were recognized with a low quality, the system could gain from adding the manually transcribed data.

5.1.1. Union

As first approach, we built the union of the phrase tables of AUTOMATIC-TRANSCRIPTION and MANUAL-TRANSCRIPTION. If a phrase pair occurs in both phrase tables, the phrase probabilities and lexical probabilities of both phrase pairs are interpolated linearly. In all other cases, we just keep the phrase pair. This method is denoted by AUTOMATIC-TRANSCRIPTION $\cup$ MANUAL-TRANSCRIPTION.

5.1.2. Two Phrase Tables

We augmented the phrase table of our baseline system, which was trained on AUTOMATIC-TRANSCRIPTION, with an additional phrase table based on MANUAL-TRANSCRIPTION. The phrase tables were connected by a bi-

Table 7: Data Statistics for pseudo ASR output as source language data (MANUAL-TRANSCRIPTION)

	English	French
Sentences	140 537	
Running Words	2 361 366	2 894 364
Vocabulary	47 159	64 627
Singletons	18 722	27 696

Table 8: Data Statistics for ASR output as source language data (AUTOMATIC-TRANSCRIPTION (cn-decoding))

	English	French
Sentences	135 603	
Running Words	2 311 602	2 803 745
Vocabulary	37 886	63 558
Singletons	12 715	27 211

nary feature, i.e phrases from AUTOMATIC-TRANSCRIPTION got the feature value 1 and phrases from MANUAL-TRANSCRIPTION the value 0. Setups using two phrase tables are marked as AUTOMATIC-TRANSCRIPTION ∘ MANUAL-TRANSCRIPTION.

5.1.3. Training Data Concatenation

In contrast to the other two methods, the training corpora MANUAL-TRANSCRIPTION and AUTOMATIC-TRANSCRIPTION were combined before the phrase extraction. In particular, MANUAL-TRANSCRIPTION and AUTOMATIC-TRANSCRIPTION were concatenated and the translation model was re-trained. This setup is named AUTOMATIC-TRANSCRIPTION + MANUAL-TRANSCRIPTION.

5.2. Results

Table 9 shows the comparison between different setups. We measured the translation quality of all systems in BLEU [14] and TER [15] on the development set as well as on the test set. First, we ran two baseline experiments. Both systems were trained on MANUAL-TRANSCRIPTION. The first setup was tuned and tested on the provided development and test sets (IWSLT 2012) and the second one on our own recognitions. It seems that a better WER results in a higher translation quality.

Using AUTOMATIC-TRANSCRIPTION (cn-decoding) performs only slightly better then the baseline. The biggest improvement was achieved by AUTOMATIC-TRANSCRIPTION (cn-decoding) ∘ MANUAL-TRANSCRIPTION in comparison to MANUAL-TRANSCRIPTION (baseline, RWTH (cn-decoding)). The translation quality was improved by 0.5 points in BLEU and 0.4 points in TER on the test set. With AUTOMATIC-TRANSCRIPTION + MANUAL-TRANSCRIPTION, we get an improvement of 0.4 points in BLEU and 0.7 points in TER. AUTOMATIC-TRANSCRIPTION (cn-decoding) ∪ MANUAL-TRANSCRIPTION performs worst of all combination methods.

The idea to improve the SLT system by using a larger corpus based on *n*-Best lists does not help. At least the system trained on 20-Best lists performs similar to the baseline. It seems that there is a mismatch between the development and test sets, which are based on confusion network decoding, and the *n*-Best lists extracted with the LATTICE-TOOL.

Finally, we employed the idea of ASR output postprocessing with an MT system. For a robust baseline, we used an existing text translation system trained on TED data, Europarl and News Commentary data, Multi-UN data and Gigaword data. This system was chosen to show the impact of this method even in a large setup. In Table 10, we compare the IMPLICIT method as described in [7] with our approach (POSTPROCESSING).

The training data for IMPLICIT setup was preprocessed by removing all punctuation marks and case information from the source language data, while the target language is kept untouched. The removal was done after the word alignment. The punctuation marks in the target sentence which were aligned with punctuation marks in the source sentences become non-aligned.

For POSTPROCESSING, we set up a standard phrase-based system trained on a bilingual corpus with ASR output as source language data and manual transcription as target language data. As development and sets we used again our recognitions. The system was tuned on the development using standard MERT on 200-best lists with BLEU as optimization criterion. The output of this system was the input of the existing text translation system.

With our proposed method, we achieve an improvement of 0.9 points in BLEU and 0.9 points in TER.

Table 11 shows an example of different input (English) and their translations (French). During the postprocessing of the ASR output repetition such as "*i i*" and "*i 'm i 'm*" are transformed to "*I*" and "*I 'm*". With the IMPLICIT approach, "*i 'm i 'm*" is translated twice. In the translation of post-processed ASR output, the phrase "*je suis*" is obtained only once. It seems that the postprocessing of the ASR output helps the text translation system to translate automatically transcripted input.

6. Conclusion

In this paper, we have introduced an approach to close the gap between automatic speech recognition and machine translation in the application of spoken language translation. In a speech translation setting, we showed that using automatically transcripted text in the training process of a machine translation system can improve the translation quality.

Further, we modelled the ASR output postprocessing as machine translation. The main advantage is that the translation system used in speech translation does not require any preprocessing. On the IWSLT 2012, we got an improvement of up to 0.9 points in BLEU and TER.

In future work, we would like to improve the WER of an ASR system directly by applying a machine translation system as postprocessing step.

7. Acknowledgements

This work was partly achieved as part of the Quaero Programme, funded by OSEO, French State agency for innova-

Table 9: Comparison of results for the SLT task English-French (IWSLT 2012), including data used to train the translation model.

setup	dev		test	
	BLEU[%]	TER[%]	BLEU[%]	TER[%]
MANUAL-TRANSCRIPTION (baseline, IWSLT 2012)	18.0	69.1	20.8	62.7
MANUAL-TRANSCRIPTION (baseline, RWTH (cn-decoding))	18.5	68.4	21.1	62.5
AUTOMATIC-TRANSCRIPTION (cn-decoding)	18.4	68.8	21.3	62.3
AUTOMATIC-TRANSCRIPTION (cn-decoding) $\cup$ MANUAL-TRANSCRIPTION	18.6	68.1	21.2	62.2
AUTOMATIC-TRANSCRIPTION (cn-decoding) $\circ$ MANUAL-TRANSCRIPTION	18.7	68.0	21.6	62.1
AUTOMATIC-TRANSCRIPTION (cn-decoding) + MANUAL-TRANSCRIPTION	18.6	67.9	21.5	61.8
AUTOMATIC-TRANSCRIPTION (1-Best)	18.4	68.7	21.1	62.4
AUTOMATIC-TRANSCRIPTION (10-Best)	18.4	68.8	21.0	62.3
AUTOMATIC-TRANSCRIPTION (20-Best)	18.5	68.6	21.2	62.4

Table 10: Comparison between the methods IMPLICIT and POSTPROCESSING on the SLT task English-French (IWSLT 2012).

method	dev		test	
	BLEU[%]	TER[%]	BLEU[%]	TER[%]
IMPLICIT	19.2	67.8	22.5	61.6
POSTPROCESSING	20.1	67.2	23.4	60.7

Table 11: Comparison of different input sentences and the corresponding reference and translation. POSTPROCESSING is the output of the SMT which postprocesses the automatic transcription.

Input/Translations	
automatic transcription	and you know i i thought well i 'm i 'm like living in a science fiction movie
manual transcription	and I thought like , " Wow . I am like living in a science fiction movie .
POSTPROCESSING	and , you know , I thought , " Well , I 'm like living in a science fiction movie .
IMPLICIT translation	et , vous savez , je me suis dit : " Eh bien , je suis comme je suis vivant dans un film de science-fiction .
POSTPROCESSING translation	et , vous savez , j' ai pens : " Eh bien , je suis vivant dans un film de science-fiction .
reference translation	et l j' ai pens : " Wow . c' est comme si je vivais dans un film de science-fiction .

tion. The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

8. References

[1] T. Faruquie, N. Rajput, and V. Raj, "Improving automatic call classification using machine translation," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, april 2007, pp. IV–129 –IV–132.

[2] T. Kawahara, K. Shitaoka, and H. Nanjo, "Automatic transformation of lecture transcription into document style using statistical framework," in *INTERSPEECH*. ISCA, 2004.

[3] P. Xu, P. Fung, and R. Chan, "Phrase-level transduction model with reordering for spoken to written language transformation," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, march 2012, pp. 4965 –4968.

[4] M. Paulik and A. Waibel, "Spoken language translation from parallel speech audio: Simultaneous interpretation as slt training data," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, march 2010, pp. 5210 –5213.

[5] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling punctuation prediction as machine translation," in *International Workshop on Spoken Language Translation*, San Francisco, California, USA, Dec. 2011, pp. 238–245.

[6] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, May 2012, pp. 261–268.

[7] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *International Workshop on Spoken Language Translation*, Kyoto, Japan, Nov. 2006, pp. 158–165.

[8] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating machine translation output with automatic sentence segmentation," in *International Workshop on Spoken Language Translation*, Pittsburgh, PA, USA, Oct. 2005, pp. 148–154.

[9] M. Sundermeyer, M. Nußbaum-Thom, S. Wiesler, C. Plahl, A. El-Desoky Mousa, S. Hahn, D. Nolden, R. Schlüter, and H. Ney, "The RWTH 2010 Quaero ASR evaluation system for English, French, and German," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP*, 2011, pp. 2212–2215.

[10] J. Wuebker, H. Ney, and R. Zens, "Fast and scalable decoding with language model look-ahead for phrase-based statistical machine translation," in *Annual Meeting of the Assoc. for Computational Linguistics*, Jeju, Republic of Korea, July 2012, pp. 28–32.

[11] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003, pp. 160–167.

[12] A. Stolcke, "Srilman extensible language modeling toolkit," in *In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, 2002, pp. 901–904.

[13] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[14] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, IBM Research Report RC22176 (W0109-022), Sept. 2001.

[15] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA, August 2006, pp. 223–231.

Towards a Better Understanding of Statistical Post-Edition Usefulness

Marion Potet[1], *Laurent Besacier*[1], *Hervé Blanchon*[2], *Marwen Azouzi*[1]

[1] UJF-Grenoble1, [2] UPMF-Grenoble2
LIG UMR 5217, Grenoble, F-38041, France
FirstName.LastName@imag.fr

Abstract

We describe several experiments to better understand the usefulness of statistical post-edition (SPE) to improve phrase-based statistical MT (PBMT) systems raw outputs. Whatever the size of the training corpus, we show that SPE systems trained on general domain data offers no breakthrough to our baseline general domain PBMT system. However, using manually post-edited system outputs to train the SPE led to a slight improvement in the translations quality compared with the use of professional reference translations. We also show that SPE is far more effective for domain adaptation, mainly because it recovers a lot of specific terms unknown to our general PBMT system. Finally, we compare two domain adaptation techniques, post-editing a general domain PBMT system *vs* building a new domain-adapted PBMT system with two different techniques, and show that the latter outperforms the first one. Yet, when the PBMT is a "black box", SPE trained with post-edited system outputs remains an interesting option for domain adaptation.

1. Introduction and Related Work

The post-edition task consists of editing the textual output produced by an error-prone process (Machine Translation, Optical Character Recognition, Speech Recognition, etc.) in order to improve it. In documents diffusion workflows where Machine Translation (MT) is one of the components, manual post-edition has been used for years. The MT system produces raw translations (or translation hypotheses) which are manually post-edited by professional translators or trained post-editors who correct the translation errors.

Many studies have shown the benefits of using MT combined with manual post-edition in a diffusion workflow. The work presented in [1] showed that even if post-editing raw MT output does not lead to any improvement in terms of productivity, it helps to produce significantly better translations compared to direct manual translations from the source text, regardless of the language direction, the text difficulty or the translator's experience. Autodesk recently draw opposite conclusion of an experiment to test whether using MT would improve translators' productivity or not. Indeed, the results[1] showed that post-editing MT output leads to a significant increase in productivity when compared with translations done from scratch, whatever the language pair, the experience and preference (post-editing or translating from scratch) of the translator, or the sentence length.

Improving the quality of the output in terms of fluency and adequacy has always been a major goal of MT developers, and in the manual post-edition setting, *"the better the MT output, the easier and faster post-edition will be"*. In the early 90's, K. Knight and I. Chander [2] proposed automated post-edition (APE) in order to help with article selection when translating from Japanese to English. Later, J. Allen and C. Hogan [3] proposed the development of an automated rule-based post-edition module able to capture and correct *"the frequent and repeated errors produced by Rule-Based Machine Translation (RBMT) systems*. Then, J. Elming [4] was the first to propose and evaluate an APE module. In his settings, J. Elming carried out domain-specialized translations of chemistry patents, cascading a RBMT system called *Patrans*, used to produce raw translations, with a "transformation-based" APE trained on 12 000 manually post-edited translations, to correct the raw output. There was a significant improvement in translation quality with the use of a "transformation-based" APE. The increasing amount of raw MT translation (hypotheses) aligned with their manually post-edited good translations gave rise to the idea of automatic statistical post-edition. A statistical post-edition (SPE) system is developed as a monolingual statistical MT system using the original hypotheses as the source language and the human post-editions as the target language.

In 2007, M. Simard & al. [5] were the first to propose the use of a phrase-based statistical machine translation (PBMT) system for SPE purpose. In this framework, the PBMT aims to learn "correction rules" between initial MT hypotheses (PBMT source language) and their corrected version (PBMT target language). Such an approach makes SPE easy to learn and tune with new training data. In their work, they successfully showed the efficiency of using an SPE system (built with the PBMT *Portage*) to improve the output of a commercial RBMT system. The experiments were done in a specific domain (a job offer Web site[2]) and the SPE system was trained using 35,000 manually post-edited sentences. Encouraged by these results, post-editing the outputs

[1] http://translate.autodesk.com/productivity.html

[2] http://www.jobbank.gc.ca

of the PBMT system *Portage* was also tried but in this setting no improvements were observed. In the same way, the following studies described in [6], [7] and [8] have shown that a RBMT system that was automatically post-edited by a PBMT system performed significantly better than each of the individual systems on their own.

Quite a lot of studies have focused on pipeline architectures where SPE systems are successfully applied to RBMT systems outputs to improve translation quality. However, only few studies ([9, 8, 10]), have investigated the efficiency of SPE systems applied after PBMT systems.

The goal of our study is to provide a better understanding of SPE usefulness when pipelined to PBMT systems. We first describe our baseline experimental settings (Section 2) and then we try to answer the following questions: is there a difference between a real and a simulated corpus for SPE training (Section 3)? Is SPE useful in improving a generic PBMT system and what explains the effectiveness of SPE on specialized domain (Section 4)? And, finally, is SPE really the simplest and most efficient and effective way for domain-adaptation purposes (Section 5)?

2. Experimental setting

2.1. Baseline PBMT

Our baseline MT system (described in more detail in [11]) translates news stories (general domain) from French into English. It is a state-of-the-art phrase-based machine translation (PBMT) system presented at the international Workshop of Machine Translation (WMT[3]) evaluation campaign in july 2010.

The system was built using free open source toolkits: we used standard Moses [12] system set-up, a 3-gram language model trained with SriLM [13] and Kneser-Ney smoothing, the GIZA++ implementation of IBM word alignment model 4 [14] and the phrase extraction heuristics described in [12]. The system has been trained on two parallel corpora, containing in total 1,638,440 aligned sentences: the fourth version of the Europarl corpus (data derived from transcriptions of European parliament proceedings) and news corpora (data extracted from various Websites). Both corpora were provided in the framework of WMT 2010.

The PBMT decoding model is a log-linear combination of fourteen weighted feature functions extracted from the monolingual and bilingual training data: six distortion models; lexicon word-based and phrase-based translation models for both directions; a target language model; and word, phrase and distortion penalty models.

2.2. Post-edited corpus

Our parallel post-edited corpus is a set of 10,881 French/English sentences taken from several news corpora (WMT evaluation campaigns from 2006 to 2010). Each sentence has been translated with our baseline PBMT system and the translation hypotheses have been manually post-edited by human annotators who were given the French source sentence and its English translation hypothesis and had to verify the translation quality and correct it if needed.

Post-editions were collected using a crowdsourcing Web platform (Amazon Mechanical Turk - MTurk). The ethical, social and economic aspects implied by such tools are subject to intense debates [15], so we defined and applied the following "good conduct" guidelines: data collected for the contributors should be used for non-profit organization and available for free to the community; contributors should be informed about the context of the task (Who are we? What are we doing? And why?); contributor should be paid a decent amount (with a reasonable hourly rate); and contributors should be filtered by country of residence according to the task, to avoid those who consider MTurk as their major source of income (we only authorized American, Canadian, and French residents to participate in our study).

Contributors were required to have an understanding of the French language and be fluent in English. Clear instructions and controlled review allowed us to deal with untrained human post-editors (native of the target language or not). A complete analysis of the collected data indicated high quality corrections with more than 94 % of the crowdsourced post-editions which are at least of professional quality. Some examples of translation hypothesis corrections collected during the post-edition campaign are given in Table 1. The post-editions corpus collection and data analysis are more detailed in [16].

The collected corpus was divided into three subsets: 8,681 sentences for the SPE training set, 1,000 sentences for the SPE development set, and 1,200 sentences for the SPE test set. Thus, all the following SPE experiment results are evaluated on the 1,200 sentences long test corpus.

For each French source sentence, we have our English baseline PBMT translation hypothesis and two different reference translations: the baseline post-edited output and an independent professional translation provided with the parallel corpus.

2.3. Baseline SPE system

As in many of the previous experiments reported here, we have considered automatic post-edition as a translation task performed by a PBMT system where the source corpus consists of the raw MT outputs and the target corpus consists of the post-edited version of these raw translations.

Our SPE system was developed using the same architecture and the same tools we used for our baseline system (Moses, SriLM and GIZA++). We trained the SPE models on the training set of the post-edited corpus (8,681 sentences) and adjusted the model's features weights with the Minimum Error Rate Training (MERT) process [17] on the development set of the post-edited corpus (1,000 sentences).

The language model was trained on a general domain cor-

[3] http://www.statmt.org/wmt10/

Source Sentence	PBMT translation	PBMT + human corrections
• La police anti-émeutes les ont aussitôt encerclés et sont intervenus sans ménagement, jetant plusieurs d'entre eux à terre.	• The anti-riot police were immediately surrounded and spoke blunty, several of them on land.	• ~~The~~ **A**nti-riot police**men** ~~were~~ immediately surrounded **them** and ~~spoke blunty~~ **stepped in ruthlessly**, **throwing** several of them ~~on land~~ **to the ground**.
• Forte mobilisation à Copenhague et à travers le mode, pour le climat.	• Strong involvement in Copenhague and in the world climate.	• Strong ~~involvement~~ **mobilization** in Copenhague and ~~in~~ **across** the world **for the** climate.
•Il y a des rivières qui s'assèchent en Afrique, des cours d'eau où l'on peut marcher comme on ne l'avait jamais fait avant.	• There are rivers are drying up in Africa, rivers where you can walk as it had never done before.	• There are rivers ~~are~~ drying up in Africa, ~~rivers~~ **watercourses** where ~~you~~ **one** can walk as it had never done before.

Table 1: Examples of PBMT hypothesis post-editions

pus of 48,653,884 english sentences (about 2 billion words).

The result is a phrase table where English baseline SMT output segments are aligned with their corresponding human post-edition. As a statistical translation model, the SPE system takes as input a raw MT output and produces a new translation hypothesis using its models.

2.4. Evaluation metrics

Translation output quality has been evaluated using the Translation Error Rate (TER) [18] and the BLEU score [19]. The TER score reflects the number of edit operations (insertions, deletions, words substitutions and blocks shifts) needed to transform a hypothesis translation into the reference translation, while the BLEU score is the geometric mean of n-gram precision. Lower TER and higher BLEU scores suggest better translation quality. To ensure that differences between scores are real, we estimated the statistical significance of test results in terms of BLEU score, according to the bootstrap resampling method described in [20].

3. Real *vs* Simulated post-edited corpus for SPE training

3.1. Previous work

In order to build SPE systems, manually post-edited MT hypotheses are usually used as target translations instead of translations produced by professional translators. When pre-existing human translations are used, we will speak of "simulated PE" in contrast to "real PE" when target translations are manually post-edited MT hypotheses. It is important to notice that the "real PE" setting corresponds to the workflows implemented in real-life situations (when users feedback is re-used to improve a given system) and "simulated PE" setup will allow access to much more training data (use of pre-translated parallel corpus).

Several works [21, 10, 22, 9] have attempted to show that SPE can be successfully trained on pre-existing human translations rather than on system-specific post-edited translations. Both simulated (MT system hypotheses aligned with their human translations version) and real post-edited (MT system hypotheses aligned with their manually post-edited versions) training corpora are used in [23]. Each setting ("real" SPE and "simulated" SPE) shows good results, but performances are not really comparable because neither the RBMT system baseline nor the SPE training corpus (in terms of size and domain) are the same in the two cases.

To our knowledge, there is no work that compares both approaches (real *vs* simulated PE) on the same source language data (post-edited MT hypotheses *vs* professional translations) to train an SPE. Considering the same source language data, we tried to find out if a simulated PE corpus is as effective as a real PE corpus to train an SPE system. This is what we will try to find out in the following experiment.

3.2. Experiment

In order to build two comparable SPE using real *vs* simulated target corpus, we used in both cases the same training corpus on the source side (the one described in 2.2) and, for one system we used the PBMT post-edited hypotheses ("real" setting) on the target side and for the other system, we used the translations provided with the parallel corpus ("simulated" setting) as the target side. Both SPE were applied on the same PBMT system outputs and we estimated the translation quality of each SPE on the test corpus (1,200 sentences) using the same distinction as we did for the training corpus: we used the test set post-edited MT outputs, for the "real" setting, and the professional translations for the "simulated" setting.

System	Simulated PE corpus	Real PE corpus
PBMT	55.3 (*26.5*)	22.8 (*62.1*)
PBMT + SPE	57.5 (*25.0*)	23.4 (*61.3*)

Table 2: Performance — TER (*BLEU*) scores — according to the use of the simulated *vs* the real post-edited corpus to train the SPE

3.3. Results

As presented in Table 2, raw PBMT output obtains a TER score of 22.8 when compared with human post-editions and 55.3 when compared with independent reference translations. A TER score of 22.8 means that slightly over 22.8% of the words needed to be changed to produce the "correct" (or reference) translation.

We expected that real post-edited corpus would lead to better results than the simulated one because of the closeness between MT raw translation hypotheses and translation post-editions. Applying the "real" SPE on PBMT outputs led to a slight increase of the TER (from 22.8 for PBMT outputs to 23.4 after statistical post-editing) and decrease of BLEU score (from 62.1 for PBMT outputs to 61.3 after statistical post-editing). However, these differences in scores do not reach a significant level (according to [20]).

So, the SPE system trained on real post-edited corpus does not significantly degrade translation results, whereas there is a significant deterioration when post-editing with the SPE trained on simulated post-edited corpus (after statistical post-editing, translation quality loses relatively 4.0% of TER score and 6.0% of BLEU score).

According to our experiment settings (i.e. a medium size corpus and general domain data), we noticed that statistical post-edition of our PBMT system brings no improvement whatever the data (real *vs* simulated) used for SPE training.

3.4. Is more data always better?

To complete our previous result, we studied the impact of training corpus size on the SPE performance. Given the moderate size of our available human post-edited corpora (10,881 sentences), we considered simulated SPE to carry out larger-scale experiments.

We used the French/English United Nation parallel corpus which consists of the texts of resolutions made by the UN General Assembly, translated by professionals. In the SMT translation community, this corpus is widely used as a general and large training corpus[4].

We considered the 8,681 sentence-sized (10k) news corpora (see part 2.2) and split the UN corpus to set up a 50,000 sentence-sized (50k), 100,000 sentence-sized (100k), 500,000 sentence-sized (500k), 1,000,000 sentence-sized (1M) and 2,000,000 sentence-sized (2M) corpora (each included the 10k news corpus). We then trained SPE systems on those 6 corpora. Note that the only thing that differentiates the systems is the training corpus size. The LM used in the different sized experiments is the same as the one used by the baseline SPE system in Section 2.3.

We evaluated the different SPE systems on the test set and report the performances, in terms of TER and BLEU scores, on Figure 1 (systems are ranked according to their training corpus size). The results show no significant gains,

[4]The corpus is available at http://www.statmt.org/wmt12/translation-task.html

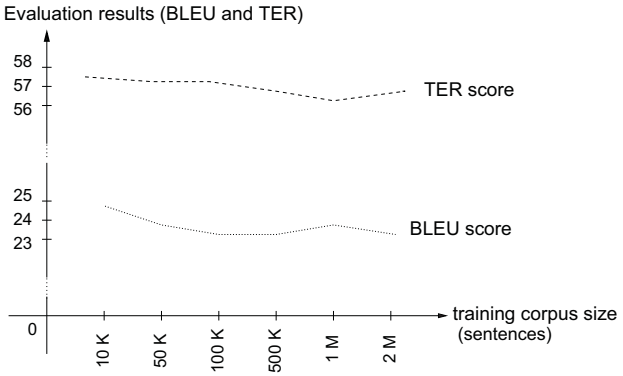


Figure 1: Performance — TER (*BLEU*) scores — of simulated SPE systems according to training corpora size (in sentences)

neither for the TER score nor for the BLEU score, while the corpus size increase. In other words, in a general French/English context translation, additional training data do not improve the SPE result of our PBMT system.

4. General domain *vs* Domain-specific application for SPE

4.1. Previous work

As SPE has shown its effectiveness in significantly improving RBMT results, further works have focused on its application in domain adaptation. Thus, P. Isabelle & al. [6] and M. Simard & al. [7] showed that an SPE trained on domain-specific data could be used to adapt a general RBMT system to a new specialized domain.

D. De Ilarraza & al. [8] noticed that if applying an SPE system after a RBMT system is efficient enough to adapt the RBMT system to a new domain, applying an SPE system after a PBMT system, for the same task, does not lead to any improvement. In their works, A. Lagarda & al. [10] and H. Becahara & al. [9] reached the same conclusion when they applied a baseline domain-specific SPE on generic PBMT system outputs. The work presented in [9], meanwhile, proposed some SPE customizations, by adding the source context into the post-edition to improve PBMT domain-adaptation.

Even if these studies confirm SPE efficiency when applied after RBMT for domain adaptation purpose, they do not show positive results when an SPE system is applied after a PBMT system. As shown before in our study, general-domain SPE brings no improvement when applied after a generic PBMT system. If the SPE system could not correct the PBMT system, can an SPE system be used to adapt the same baseline system to a new domain? To answer this question, we set up an experiment to test the potential of a generic SPE approach compared to a domain-specific one.

4.2. Experiments

Given the nature of our available corpora, the following experiments use only a simulated post-edited corpus for SPE training. We used the post-edited corpus described in 2.2 with the independent professional reference translations and a domain-specific corpus on water sciences.

The domain-specific and general corpora used for our experimentations are described in Table 3. They are very comparable in terms of size and only differ from each other by their domain specificity. As the general domain corpus, the domain-specific corpus has been split into a training set ($\approx$ 9,000 sentences), a development set (1,000 sentences) and a test set (1,200 sentences). A new SPE system has been built using the domain-specific data (the previous one presented used general domain data).

4.3. Results

As seen in Table 4, the general domain baseline PBMT achieves a TER score of 55.3 on the general domain and a score of 46.7 on the specific domain, meaning that these latter data are easier to translate than those of the general domain. Although the general domain SPE brings no gain on general data, the specific-domain SPE significantly improves the baseline PBMT outputs on the specialized data: the TER score subsequently drops from 46.7 to 39.2 (-19.2%) and the BLEU score follows the same trend, increasing from 33.3 to 40.1 (+20.6%).

The first line of Table 5 indicates that the domain-specific SPE is not only better (as seen in Table 4) but it modifies more sentences (91%) as compared to the general domain SPE (which modifies 75% of sentences). The second line shows the proportion of baseline PBMT translations improved through statistical post-edition: the specific-domain SPE improves 58% of the PBMT outputs while only 11% for the general domain SPE. Some examples of domain-specific translations before and after post-editions are presented in Table 6.

System	Specific domain	General domain
PBMT	46.7 (*33.3*)	55.3 (*26.5*)
PBMT+SPE	39.2 (*40.1*)	57.5 (*25.0*)

Table 4: Systems' performances — TER (*BLEU*) scores — according to the domain

4.4. Real domain adaptation or vocabulary correction?

The main follow up questions raised by these new experiments are: Why does SPE work on the domain-specific inputs and fail on general ones? Is SPE doomed to domain-adaptation? In [21], SPE modifications in the raw MT output have been manually categorized and results conclude that SPE makes significant improvements in terms of lexical choice, but no improvement in word reordering or grammaticality.

[5]http://www.statmt.org/wmt10

Post-edit rate	Specific domain	General domain
Post-edited sentences	91 %	75 %
SPE-improved PBMT outputs	58 %	11 %

Table 5: Rate of post-edited sentences according to the domain

Is SPE successful in domain-adaptation task only thanks to lexical correction? We decided to analyze how SPE handles out-of-vocabulary (OOV) words. So, we compared OOV words before and after general and domain-specific SPE. We did this experiment on two sets of 2,200 sentences (concatenated development and test sets for both domain-specific and general domain settings).

The results, shown in Table 7, point out an equivalent proportion of OOV words in both sets (2.8% for the domain-specific corpus and 2.7% for the general one) but with a type-token OOV word ratio[6] of 61 %, the domain-specific data contain less lexical variation than the general one. The application of SPE corrected 56% of the PBMT outputs OOV words for the domain-specific data and 7% for the general data.

OOV words statistics	Specific domain	General domain
Outputs with OOV words	40 %	43 %
Rate of OOV words	2.8 %	2.7 %
Type-token OOV words ratio	61 %	72 %
OOV words corrected by SPE	56 %	7 %
OOV common nouns corrected by SPE	**42%**	**1%**

Table 7: OOVs statistics according to the domain

Nature of corrected OOV words	Specific domain	General domain
Proper nouns	16.8 %	**46.8 %**
Foreign language words	2.3 %	34.7 %
Source mistake	1.5 %	2.4 %
Numbers	3.3 %	5.6 %
Common nouns	**75.6 %**	9.7 %

Table 8: Nature of corrected OOVs according to the domain

In order to better understand these results, we analyzed the nature of OOV words for both data sets. The results

[6]The type-token ratio is a measure of text vocabulary variability. The higher is the ratio, the larger is the lexical variability.

Corpus	Specific	General
Domain	Water Sciences	News
Nature	EOLSS encyclopaedia	Various websites
Vocabulary size	14 015 words	21 982 words
Sentence length	$\approx$ 22 words	$\approx$ 28 words
Source	Corpus translated by SECTra_w project [24]	Corpus provided by WMT international workshop[5]

Table 3: General *vs* specific corpus comparison

Source sentence	PBMT translation	PBMT + SPE result
• Unité africaine de recherche sur les questions de l'eau	• African **unit of research on issues of water**	• African **water issues research unit**
• Réduction de la salinité des eaux souterraines dans les zones agricoles	• Reducing **the salt content of groundwater** in agricultural areas	• Reducing **groundwater salinity** in agricultural areas
• L'offre est en grande partie déterminée par la productivité dans les zones irriguées et pluviales[...]	• **The offer** is largely determined by productivity in the irrigated **areas and *pluviales***[...]	• **Supply** is largely determined by productivity in the irrigated **and rain-fed areas**[...]

Table 6: Examples of specific-domain translations

are presented in Table 8. We noticed that the baseline PBMT OOV words are mostly common nouns (75.6%) for the domain-specific data, whereas they are mostly proper nouns and foreign language words (81.5%) for the general data. In a translation task, the latter just have to be copied out (this is what the baseline PBMT usually does with OOV words) whereas common nouns have to be correctly translated. The figure to retain is that SPE corrects 42% of OOV common nouns on the domain-specific data and only 1% on the general data.

OOV correction analysis also showed that the SPE learned to correct very domain-specific words that frequently appear in the data (for example: ions, évaporite, électrolytes, etc.). Our experiment results indicate that, when applied to domain specific data, SPE corrects a lot of OOV common nouns. This can explain the overall translation quality improvement. To sum up: SPE does not safely and effectively correct a general PBMT system output but it does some good work for domain adaptation thanks to its ability to restore domain-specific vocabulary. The follow up question remains: Is another simple domain adaptation method capable of outperforming SPE?

5. Domain-specific SPE *vs* other domain-adaptation methods

As SPE seems to be an efficient domain-adaptation method, we propose to compare this approach to other usual domain-adaptation methods. For these experiments, we used the general domain data and the PBMT system described in Section 2 and the domain-specific data described in Section 4.

5.1. Corpus-based domain-adaptation experiments

Our corpus-based domain-adaptation method consists simply of appending the domain-specific corpus to the general domain training corpus and then build the PBMT system as usual. The success of this straightforward method depends on the homogeneity of both corpora, i.e. the way they complete one another (in terms of OOV coverage, for example) and basically on the relative size of both corpora. As seen in Table 9 line (2), we get a significant improvement in terms of BLEU and TER (+37.0% and -25%) despite the fact that the general domain data greatly outnumbers the domain-specific one (which represents only 0.5% of the total training corpus). However, we reached better improvement by giving greater weight to the domain-specific training data by appending it several times to the corpus used for training (results line (3), (4) and (5)). The system achieved its best performance in terms of BLEU and TER (+48.2% and -45.0%) with domain-specific data weighing 35.5% of the total corpus size (line (4)).

5.2. Model-based domain-adaptation experiments

Corpus-based domain-adaptation methods led to a huge increase in the training time. Instead of simply concatenating all of the available training data, we have experimented with two methods using multiple phrase tables (PT) and language models (LM).

On one hand, we built separate phrase tables and language models for each data sets (domain-specific LM and PT, general domain LM and PT) and then we used all of them in the log-linear model. This model-based adaptation method is referred to "*domain-specific PT-LM*$_1$", line (6) in Table 9.

On the other hand, we tried to interpolate specific and general language models before using it in the log-linear

Baseline PBMT	...with domain-specific SPE	...with domain-specific PT-LM$_2$
• There is some maximum quantity of water vapor for each of the value of the air temperatures. • This is in connection with the effects of noise. • A reduction in consumption of animal products will very probably a positive effect on consumption of water to agriculture	• There is **some maximum** amount of water vapor for **each of the** value of the air temperature. • This is in **connection with the effects of acoustic**. • A **shift** in consumption of **animal products** will **most likely** positive effect on water consumption **to** agriculture	• There is **a certain** amount of water vapor **maximum possible** for **every** value of the air temperature. • This is in **relation to the acoustic effects**. • A **reduction** in **the** consumption of **products of animal origin** will **very probably** a positive effect on water consumption **of** agriculture

Table 10: Examples of translations according to the domain-adaptation method

Systems	TER(*BLEU*)
Generic PBMT	*46.7 (33.3)*
(1) domain-specific SPE	39.2 (*40.1*)
—— Corpus-based adaptation ——	
(2) 1×domain-specific corpus (=0.5%)	35.2 (*45.5*)
(3) 10×domain-specific corpus (=5.2%)	33.1 (*48.5*)
(4) 10^2×domain-specific corpus (=35.5%)	32.3 (*49.2*)
(5) 10^3×domain-specific corpus (=84.5%)	32.6 (*48.9*)
—— Model-based adaptation ——	
(6) domain-specific PT-LM$_1$	33.0 (*47.9*)
(7) domain-specific PT-LM$_2$	32.2 (*49.2*)

Table 9: Performance — TER (*BLEU*) scores — on a specialized domain corpus according to domain adaptation method

model. The LMs interpolation weights were estimated using an EM algorithm[7] and then, the two LMs were merged (using SriLM tool [13]) into a single model. We observed a slight improvement in terms of BLEU and TER (referred as "*domain-specific PT-LM$_2$*", line (7) in Table 9).

According to the experiment results, the systems produced with the corpus-based and the model-based domain-adaptation methods (TER from 32.2 to 35.2) significantly outperform the SPE method (TER of 39.2). Figure 10 shows some examples of specific-domain translation hypotheses using the domain-specific SPE system and the *domain-specific PT-LM$_2$* system.

6. Conclusion

The aim of this study was to better understand the usefulness of statistical post-edition to improve PBMT systems outputs. In order to do so, we tried to answer the following questions: Is simulated SPE really comparable to real SPE? Can an SPE system be applied to PBMT system outputs in order to improve them? Can an SPE system be used to adapt a general domain "black-box" MT system towards a particular domain? For domain-adaptation, is SPE more efficient than building a new domain-adapted PBMT sytem?

First, we noticed that an SPE system trained on moderate-size and general domain data ($\approx$ 9,000 sentences) brings no gain to a baseline general domain PBMT system in terms of TER or BLEU. In such a setting, using manually post-edited outputs ("real setting") instead of independent professional reference translations ("simulated setting") leads to a slight improvement of the translation quality. We also observed that increasing the amount of the training data is not sufficient to significantly improve the SPE system performances. So, whatever the available corpora, it seems difficult to improve/correct, general domain PBMT outputs with statistical post-editing.

However, according to our experiments, an SPE system seems more effective when trained on domain-specific data and can be successfully used to adapt a general PBMT system to a new specialized domain. Comparing our general domain and domain-specific SPE systems, we pointed out that better results are achieved with the latter one. This is mainly due to the fact that in-domain unknown common nouns of the general-domain PBMT system are recovered by the domain-specific SPE system.

In our last experiment we decided to compare SPE-based domain-adaptation with another adaptation approach which consist of training specialized phrase-tables and language models and interpolate them with the baseline general models. For this latter experiment, each methods shared the same baseline PBMT system and the same data sets. Results show that the PT-LM domain-adaptation method significantly outperforms the domain-specific SPE.

It is however important to note that in the case of model-based adaptation, a brand new PBMT system is built. There might be practical situations where it is impossible to build a new PBMT system (the one used is a "black box"), or it may be useful to keep a general PBMT system and a record of several SPE systems each adapted to a different domain.

7. References

[1] I. Garcia, "Translating by post-editing: is it the way forward?" *Journal of Machine Translation*, vol. 25, no. 3,

[7] http://sourceforge.net/apps/mediawiki/irstlm/index.php?title=LM_interpolation

pp. 217–237, 2011.

[2] K. Knight and I. Chander, "Automated Postediting of documents," in *Artificial Intelligence Conf.*, Seattle, USA, 1994, pp. 779–784.

[3] J. Allen and C. Hogan, "Toward the development of a post-editing module for Machine Translation raw output: a controlled language perspective," in *International Controlled Language Applications workshop*, Washington DC, USA, 2000, pp. 62–71.

[4] J. Elming, "Transformation-based corrections of rule-based MT," in *European Association on Machine Translation Conf.*, Oslo, Norway, 2006, pp. 219–226.

[5] M. Simard, C. Goutte, and P. Isabelle, "Statistical phrase-based post-editing," in *North American Chapter of the Association for Computational Linguistics and Human Language Technologies conf.*, Los Angeles, USA, 2007, pp. 507–515.

[6] P. Isabelle, C. Goutte, and M. Simard, "Domain adaptation of MT systems through automatic post-editing," in *North American Chapter of the Association for Computational Linguistics*, Stroudsburg, USA, 2007, pp. 217–220.

[7] M. Simard, N. Ueffing, P. Isabelle, and R. Kuhn, "Rule-based translation with statistical phrase-based post-editing," in *Statistical Machine Translation workshop*, Prague, Czech Republic, 2007, pp. 203–206.

[8] A. Diaz de Ilarraza, G. Labaka, and K. Sarasol, "Statistical post-editing: a valuable method in domain adaptation of RBMT systems for less-resourced languages," in *Mixing Approaches to Machine Translation*, Donostia-San Sebastian, Spain, 2008, pp. 35–40.

[9] H. Béchara, Y. Ma, and J. Van Genabith, "Statistical post-editing for a statistical MT system," in *MT SUMMIT XIII*, Xiamen, China, 2011, pp. 308–315.

[10] A. Lagarda, S. Casacuberta, and E. Diaz-de Liano, "Statistical post-editing of a Rule-based machine Translation System," in *North American Chapter of the Association for Computational Linguistics conf.*, Boulder, Colorado, USA, 2009, pp. 217–220.

[11] M. Potet, l. Besacier, and H. Blanchon, "The LIG machine translation system for WMT 2010," in *Statistical Machine Translation workshop*, Uppsala, Sweden, 2010, pp. 11–17.

[12] H. Hieu, A. Birch, C. Callison-burch, R. Zens, R. Aachen, A. Constantin, M. Federico, N. Bertoldi, C. Dyer, B. Cowan, W. Shen, C. Moran, and O. Bojar, "Moses: Open source toolkit for statistical machine translation," in *Association for Computational Linguistics*, Prague, Czech Republic, 2007, pp. 177–180.

[13] A. Stolcke, "SRILM: An Extensible Language Modeling Toolkit," in *Spoken Language Processing conf.*, Denver, USA, 2002, pp. 901–904.

[14] F. J. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models," *Journal of Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[15] K. Fort, G. Adda, and K. B. Cohen, "Amazon Mechanical Turk: Gold Mine or Coal Mine?" *Journal of Computational Linguistics*, vol. 37, pp. 413–420, June 2011.

[16] M. Potet, E. Esperan¸ca Rodier, L. Besacier, and H. Blanchon, "Collection of a Large Database of French-English SMT Output Corrections," in *Language Resources and Evaluation Conf.*, Istanbul, Turkey, 2012, pp. 23–25.

[17] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in *Association for Computational Linguistics*, Sapporo, Japan, 2003, pp. 71–79.

[18] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Association for Machine Translation in the Americas conf.*, Cambridge, USA, 2006, pp. 223–231.

[19] K. Papineni, S. Roukos, T. Ward, and Z. Wei-Jing, "BLEU : A Method for Automatic Evaluation of Machine Translation," in *Association for Computational Linguistics*, Philadelphia, PA, USA, 2002, pp. 311–318.

[20] P. Koehn, "Statistical Significance Tests for Machine Translation Evaluation," in *Empirical Methods in Natural Language Processing conf.*, Barcelona, Spain, 2004.

[21] L. Dugast, J. Senellard, and P. Koehn, "Statistical post-editing on Systran's rule-based translation system," in *Statistical Machine Translation workshop*, Prague, Czech Republic, 2007, pp. 220–223.

[22] L. Dugast, J. Senellart, and P. Koehn, "Statistical post-editing and dictionary extraction: systran/edinburg submissions for WMT2009," in *Statistical Machine Translation workshop*, Athens, Greece, 2009, pp. 110–114.

[23] R. Kuhn, P. Isabelle, C. Goutte, J. Senellart, M. Simard, and N. Ueffing, "Recent advances in automatic post-editing," *Journal of Multilingual computing and technology*, vol. 21, no. 1, pp. 43–46, 2010.

[24] C.-P. Huynh, C. Boitet, and H. Blanchon, "SECTra_w.1: an Online Collaborative System for Evaluating, Post-editing and Presenting MT Translation Corpora," in *Language Resources and Evaluation Conf.*, Marrakech, Morocco, 2008, pp. 28–30.

Towards Contextual Adaptation for Any-text Translation

Li Gong, Aurélien Max, François Yvon

LIMSI-CNRS & Univ. Paris Sud
Orsay, France
{firstname.lastname}@limsi.fr

Abstract

Adaptation for Machine Translation has been studied in a variety of ways, using an ideal scenario where the training data can be split into "out-of-domain" and "in-domain" corpora, on which the adaptation is based. In this paper, we consider a more realistic setting which does not assume the availability of any kind of "in-domain" data, hence the name "any-text translation". In this context, we present a new approach to contextually adapt a translation model *on-the-fly*, and present several experimental results where this approach outperforms conventionaly trained baselines. We also present a document-level contrastive evaluation whose results can be easily interpreted, even by non-specialists.

1. Introduction

It is now a well-established fact in Statistical Machine Translation that systems must be adapted to each particular input text. Adaptation has been tackled in a variety of ways (see e.g. [1, 2, 3]), most notably by adapting the translation model, by adapting the target language model, and by adaptating the tuning set. In most of these works, it is assumed that the bilingual training corpus can be partitioned into "in-domain" and "out-of-domain" subsets relative to the input text, and that there exists some smaller "in-domain" held-out corpus to tune the system. In typical settings, large bilingual corpora are collected opportunistically; as a result, the amount of data that do not resemble closely the input text largely outweights the data that appear to be the most relevant.

Using as much data as is available for a given language pair is necessary to alleviate the data sparseness issue through better coverage: in particular, it seems to improve the alignment of some rare translation units, which would otherwise be misaligned, and yield inappropriate phrase pairs. On the other hand, adding more bilingual data increases the possibility of encountering new translations, and makes the translation of phrases more ambiguous, sometimes in a detrimental way, since not all corresponding translations (or senses) are appropriate for the input text. The data sparseness and the ambiguity problem thus entertain a repulsion relationship that is at the core of the adaptation problem (see e.g. [4]), even though the recent work of Haddow and Koehn [5] concludes that good coverage is more important than appropriate scoring: adding out-of-domain corpora containing examples of rare units benefits more to translation than the inclusion of inappropriate examples of frequent units harms it.

A practical solution is to use all the available training data, but to consider differently translation examples depending on their relevance to the input text, possibly at the corpus [1], sentence [6] or phrase [3] level. As noted e.g. by Haddow and Koehn [5], although the in-domain vs. out-of-domain distinction is frequently used, precise definitions are still lacking; in their words, "it is normally understood that data from the same domain is in some sense similar (for example in the words and grammatical constructions used)" and, in their experiments, they characterize domain differences in terms of word distributions and out-of-vocabulary rates. While some domain distinctions are clearly undebatable, such as when opposing e.g. News commentaries and parliamentary speeches, other distinctions may in fact be more difficult to draw when one considers arbitrary text inputs, as may be submitted to online translation services.

In this work, we consider a case that has been so far comparatively less studied, where the characteristics of the input text are completely unknown before translation. We thus make the following assumptions:

- The input text is short and corresponds to a coherent discourse (i.e. is not made by concatenating unrelated documents).
- The text can be from any arbitrary domain, which precludes any realistic off-line adaptation using any predefined specific bilingual corpora; therefore, the only "in-domain" corpus available is the input text itself.
- No adapted development corpus is available, which precludes the use of tuning techniques relying on a development bitext from the same data source or domain.
- Training data was collected opportunistically and no specific document metadata (e.g. genre, document boundaries) are available for the full data set.

Note that the issues of adapting alignments and target language models will not be considered in this work. As to the former, it has previously been shown that using all the available corpora during word alignment tends to improve

translation performance [7, 5], so our word alignment models will be built offline using all available parallel data. As to the latter, there is a large body of works addressing language model adaptation which all report improvements over non-adapted language models (e.g. [1]). We leave it to our future work to evaluate whether the effects of all types of adaptations can be compounded.

This paper is to our knowledge the first attempt at studying the scenario of what we call here "any-text translation", with the notable absence of some predefined identifiable in-domain training and tuning corpora. An important aspect of our scenario is that there is no guarantee that appropriate data will be available for the input text as regards e.g. genre, phraseology, theme vocabulary, or even effects of original language. Thus, adaptation will be performed with the objective of modeling some *a priori* confidence into the system's ability to translate short translation units.

Another consequence of our setting is that online adaptation is necessary and is in fact the only solution. We therefore propose an *on-the-fly* pipeline consisting of the following stages : sampling at the level of translation units is performed (similarly to [8, 9]) for selecting sentences from the training data, and instance weighting is applied for scoring phrase pairs (e.g. [6]). Based on these computations, two additional scores are then produced: the first estimates the *goodness* of each collected source phrase as a translation unit for the language pair at hand; the second estimates how much confidence should be put in the adaptated translation distribution for each source phrase[1]. An important result of the paper will be the description of a document-level contrastive evaluation scheme that enables a more interpretable analysis of the differences between two systems.

The rest of this article is organized as follows. We first describe our approach to on-the-fly instance weighting for adapting translation models (section 2). We then describe how to model the goodness of source phrases (section 3) and to compute confidence scores for (adapted) translation distributions (section 4). The experimental section (section 5) is decomposed into a description of data sets (section 5.1), systems (section 5.2), and evaluation settings (section 5.3). We next present the main experimental results (section 5.4) and discuss them in relation to previous works (section 6). We finally conclude and describe plans for future work.

2. Instance-weighting for contextual adaptation

Adaptation can be tackled as a data selection problem: given an in-domain training corpus and out-of-domain corpora, a fixed number of sentences are selected in the out-of-domain corpora on the basis of their similarity to the in-domain corpus. These sentences may be denoted as *pseudo in-domain data* [11], where it is hoped that, given the selected number of sentences to draw, performance will be improved. This approach is in fact flawed in a particular respect, as it does not provide any guarantee that instances of rare units will be selected, specifically if they do not occur in sentences resembling the in-domain data. This has been sometimes solved by ad-hoc strategies to recover infrequent units [12].

We would like instead to make use of all available training corpora. Sampling at the level of phrases is an efficient solution to achieve this goal [8, 9]. Indeed, suffix arrays [13] offer fast access to phrase instances in large corpora, and can be used to select a given number of instances of phrases, rather than sentences, thereby ensuring that all the phrases present in a corpus are appropriately covered.[2]

Previous approaches to sampling have resorted to *random deterministic sampling*, which picks a given number of examples by scanning the suffix array index at fixed intervals (hence the apparently random, and actually deterministic, behavior). This, of course, is sub-optimal as it does not attempt to select the most appropriate data for the input text. We may instead resort to criteria that are often used in data selection approaches: Information Retrieval similarity measures such as `tf.idf` and Information Theory measures such as perplexity.

Once a sample has been collected for every source phrase, (pre-computed) word alignments are retrieved to extract the corresponding translations. Assuming a set of retrieved sentences and their individual similarity score, denoted as w_i, the adapted translation model can be estimated by weighting each example with the corresponding sentence weight [6]:

$$p_{iw}(e|f) = \frac{\sum_{j \in T_f \cap T_e} w_j c_j(e,f)}{\sum_{j \in T_f} w_j c_j(f)}, \qquad (1)$$

where T_f (resp. T_e) is the set of source (resp. target) sentences containing f (resp. e), and $c_j()$ is the count function.

3. Estimating the goodness of translation units

Given that our sampling strategy ensures that all occurrences (up to a maximum sampling size) of each source phrase will be retrieved, all source phrases that are found in the training corpus will also be present in the phrase table. Although no definitive criterion as to what constitutes a good phrase translation unit has emerged[3], the two following criteria have been proposed:

[1] Note that in the present work, the effect of this score will only be to act as a *segmentation* model, so that some segmentation may be preferred over some other. Future work will include searching for more translation examples for those unreliable phrases, as hinted by [5], and having recourse to automatic paraphrasing (e.g. [10]) of those phrases.

[2] Callison-Burch *et al.* [8] found that a sample size of 100 was sufficient for German-to-English phrase-based SMT, while Lopez [9] determined that 300 was an appropriate value for Chinese-English hierarchical SMT. We will use a larger sample size of 1,000 in our experiments in an attempt to let instance weighting find the most appropriate examples from a larger sample.

[3] For instance, limiting phrases to constituents was found to be sub-optimal [14]. The very definition of what a *phrase* is with respect to the SMT problem poses many interesting research questions, see e.g. [15].

- Given some word alignment between a source and target parallel corpora, the absence of an aligned target phrase for a given source phrase may suggest that the corresponding failure of the extraction process should be accounted for in the translation model. Lopez [9] therefore proposes the following *coherent* estimate of the translation conditional probability:

$$p_{coherent}(e|f) = \frac{c(f,e)}{c(f)} \quad (2)$$

  where $c(f)$, the number of occurrences of the source phrase, corresponds to the total number of attempted extractions, in lieu of the traditional summation over all extracted translations for f, $\sum_{e'} c(e', f)$.

- It has been observed that the traditional heuristic approach to phrase pair extraction does not offer a consistent view over the training and the actual use of phrases by decoders. It is thus possible to have recourse to a forced alignment which results in the decoder producing what it believes is the best alignment for a given training sentence. Wuebker *et al.* [16] implement this idea using *leaving-one-out*, so that the phrase examples for each training bi-sentence are not used to decode it, and subsequently estimate their system's models on the resulting alignment. Even though this intuition does not guarantee that the retained phrases are *intrinsically* good translation units, they were selected as pertaining to best derivations allowing to reproduce the reference target sentence.

We exploit the two above ideas as follows. First, we use some pre-trained standard phrase-based system to translate its own training corpus. Instead of sticking strictly to leaving-one-out, we simply remove from the system's phrase table all source phrases occurring only once, corresponding mostly to long phrases. In addition, we consider all phrases coherent with the resulting alignment (i.e. coherent sub- or super-phrases) as candidates for extraction. Then, for all the selected occurrences of a given source phrase f, we count how many times f has both a coherent alignment in the original alignment (using GIZA++ in our case) and in the decoder alignment, and normalize by the number of occurrences of that source phrase[4]. The following calculation was used as a new feature in our experiments:

$$h_{goodness}(f) = \frac{c_{coherent}(f)}{c(f)}, \quad (3)$$

where $c_{coherent}(f)$ denotes the count of instances of phrase f being coherent with respect to both the training and decoding alignment.

[4]This can be done w.r.t. to the full corpus or a to particular sample, depending on the configuration studied.

4. Confidence estimation for adapted translations probabilities

Phrase scoring strategies used in conventional phrase-based SMT systems are based on simple count ratios and can thus be criticized on the following grounds :

1. A source phrase occuring rarely will result in its translations being over-estimated[5].
2. A majority of *inappropriate* examples for a given source phrase will result in incorrect translations being more likely for the translation model[6].

The instance-weighting scheme presented in section 2 allows us to assign an adapted weight to each individual example: in some sense, this weight should reflect the confidence that the associated translation is contextually appropriate. Intuitively, an example matching only loosely the context of the input sentence should not partipate much to the confidence that the final translation distribution is correctly estimated. The worst-case scenario would be if all available examples were poor matches (such as examples for an incorrect translation sense for a polysemous phrase). Conversely, a perfect match (such as finding in the training data the very input sentence or a very close match) would indicate that the translation distribution was derived from appropriate data, at least for this example.

In addition to the appropriateness of the examples used, their number should also participate in estimating the confidence in a translation distribution. Given a particular number of examples for a source phrase, the least informative, or least *committing*, situation would be one in which all translation examples are different, yielding the following conditional entropy:

$$H_{unif}(f) = -\sum_e p(e|f)\log(p(e|f)) = \log(\frac{1}{c(f)}) \quad (4)$$

Intuitively, the better the examples used for contextual estimation of a phrase's translations, and the better the instance-weighting scheme, the more the conditional entropy for that phrase should be reduced, as translation alternatives should be restricted to a few synonymous translations. The information gain measured as a difference of entropy values between the previous situation and the more informative situation of a given model provides some account of how much confidence should be put in the collective contribution of all weighted examples. We thus used the following as a new feature in our experiments involving adapted translation models:

$$\begin{aligned} h_{confidence}(f) &= H_{unif}(f) - H(f) \quad (5) \\ &= -\log(\frac{1}{c(f)}) + \sum_e p_{iw}(e|f)\log(p_{iw}(e|f)) \end{aligned}$$

[5]Inverse translation models and lexical weighting are in a way meant to compensate for this.

[6]Context-dependent phrase tables (e.g. [17]) is a way to address this.

Corpus		#lines	#tok.en	#tok.fr	ppl.en	ppl.fr	oov.en	oov.fr
tuning	newsco (in)	934	22.4K	25.3K	316.19	211.07	629	273
	ted (out)	934	19.6K	20.3K	265.63	164.57	238	273
test	newsco	1,859	44.2K	48.8K	307.14	222.79	1,700	1,558

Table 1: Tuning and test documents statistics

Corpus	domain w.r.t. test	# lines	# tokens.en	# tokens.fr
newsco	in	137K	3,381M	4,017M
epps	out	1,982M	54,170M	59,702M
newsco+epps	mixed	2,119M	57,551M	63,790M

Table 2: Training corpora statistics

This value increases when either the number of examples for f is high or when the entropy of the adapted translation distribution is low.

5. Experiments

We now describe experiments intended to show whether on-the-fly contextual adaptation can improve over standard estimation of translation models, as well as over a standard way of combining translation models estimated from different corpora. For this, we resort to data conditions that simulate short input documents and training corpora for which the in-domain part is either clearly identified or dissolved in a larger corpora, and use three scenarios where an out-of-domain, an in-domain and a perfect tuning set is available[7]. For each system configuration, we compute traditional evaluation metrics over the full document collection (as is typically done with corpus-based metrics such as BLEU). We also propose a new document-based evaluation method that is more appropriate for the problem at hand.

5.1. Data sets

Experiments were performed on the English-French language pair in both directions, using data released for the evaluation track of the Workshop on Statistical Machine Translation[8]. Our test document collection, described in Table 1, also stems from WMT data: it consists of a set of 76 News commentary documents (from newstest2009).

We use the tuning sets described in Table 1: one is "in-domain" (in its traditional sense in SMT) w.r.t. to our test corpus (newsco), and one is out-of-domain and is taken from presentations from TED talks[9] (ted). These conditions allow us to compare situations where tuning corpora of various degrees of appropriateness are available and can be identified as more appropriate; we will also simulate the availability of a "perfect" tuning set by performing self-tuning.

Lastly, our training corpus, described in Table 2, contains two sub-corpora of in-domain News commentaries (newsco) and out-of-domain parliamentary debates (epps). These sub-corpora will be either used separately or jointly.

[7]Performing tuning set adaptation at the document-level as in [18] will be part of our future work.

[8]http://www.statmt.org/wmt12

[9]Available from IWSLT'11: http://iwslt2011.org

5.2. Systems

5.2.1. *Off-line baseline systems*

We build standard phrase-based systems using moses[10], and use MERT for tuning parameters. We compare the following conditions: training on all available data (newsco+epps), as well as using two separate phrase tables built from newsco and epps (i.e. multiple alternative decoding paths) as is standard practice in domain adaptation where corpus boundaries are known [1].

5.2.2. *On-the-fly adapted systems*

We build various adapted systems on-the-fly. All use the word alignments produced by Giza++[11] on the full newsco+epps corpus, as out-of-domain data may improve alignment quality in our situation [7]. We test the three following sampling and instance-weighting strategies for estimating translation model: (a) random sampling and uniform weighting [8, 9] (RND), (b) using tf.idf values of training sentences [19] (IR), and (c) perplexity values of training sentences relative to each test document (PPL).[12]

An important difference with our baseline systems is that we do not estimate a back-translation model ($p(f|e)$) as this proves costly using sampling; [9] reported that this model does not have a significant impact on translation performance for large training corpora. Furthermore, we believe that such a model should in fact not be needed, were the translation model appropriately estimated (i.e. contextually appropriate), as there would be no need to compensate for the "ambiguity" in this model by considering the reverse direction.[13]

We build variants where we consider one translation model in isolation (RND, IR, PPL) as well as our source phrase goodness model (section 3) and our translation distribution confidence estimation (section 4). Parameter tuning is performed once for all with MERT, considering the tuning set as a single document. For testing, we build an adapted translation model for each document and use the previously tuned parameters to decode using the moses decoder. For self-tuning, which simulates the availability of a (smallish)

[10]http://www.statmt.org/moses

[11]http://code.google.com/p/giza-pp/

[12]Note that scoring test examples at the sentence level, as done e.g. by [6], might be sub-optimal: we would rather consider thematically-coherent units from the training corpus. We did not have this information at our disposal here, but plan to perform automatic thematic segmentation of the training corpora as part of our future work. Note also that the target side of our "in-domain" corpus (i.e. test documents) was not available for adaptation.

[13]The argument also holds for lexical weighting models, which are meant to model intra-biphrase cohesion.

perfect tuning set for each input document, each document is tuned independently using the reference corpus and the best optimization point is used for testing; this is obviously an oracle situation, and will be denoted as such in our results for `moses` and our adapted systems.

5.3. Evaluation setting and contrastive document-level evaluation

We will compare our various settings using the well-established BLEU [20] and TER [21] metrics, using initially the full test corpus made up of the full collection of documents. Absolute values being always difficult to interpret, we propose to resort to contrastive evaluation between two systems. Our contrastive document-level evaluation is performed as follows: given two systems we wish to compare, a single configuration, and a target evaluation metrics, we look on *a per document basis* which system outperformed the other for some interval (e.g. "1-2 BLEU increase", "0.5-0.75 TER decrease"). We then compute statistics over the entire document set. Considering a particular significance level for the selected metrics, we can then report the percentage of cases the first system outperformed the second system, the other way round, and when they leveled each other out, corresponding to figures that are easier to interpret.

5.4. Results

The results of all systems under our three tuning conditions are given in Table 3. It immediately stands out that we are looking at two very different situations: on the one hand, French to English translation shows a clear advantage of the adapted systems over both `moses` and the adapted baseline (`2-tables`) under all tuning conditions; on the other hand, English to French translation calls for a closer look at results as no immediate conclusion can be drawn.

Tuning condition Considering first the most likely scenario for any-text translation, we look at results obtained when using an out-of-domain tuning set for all systems. On English to French, we find that the `IR` and `PPL` system can achieve slightly better performance than `moses`, which in turn performs slightly better than `RND`. The `2-tables` adaptation system clearly failed to improve over any other system. On French to English, the situation is comparable with the exception of two differences: `IR` and `PPL` now achieve a significant improvement over `moses` (resp. +1.15 and +1.12 BLEU point), and `2-tables` now performs slightly better than `moses`.

The in-domain condition, where a tuning set from the same domain as the test set is used, exhibits a similar pattern: on English to French, `moses` and our best adapted systems are almost indistinguishable, and the `2-tables` system performs comparatively poorly. On French to English, `2-tables` now performs slightly better than `moses`, while our adapted systems outperform again the latter (+0.77 BLEU point for `IR` and +0.57 BLEU point for `PPL`).

	English → French						French → English					
	tuning condition											
	out		in		oracle		out		in		oracle	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
`moses`	28.15	55.27	28.32	**56.72**	30.07	56.61	28.36	55.66	29.46	52.07	32.11	52.69
`2-tables`	27.80	55.31	26.91	58.71	-	-	28.49	55.16	29.53	51.90	-	-
`RND`	28.01	55.15	28.17	57.12	-	-	28.24	56.44	29.99	51.83	-	-
`IR`	**28.36**	**54.83**	**28.42**	56.86	-	-	28.59	55.74	29.57	52.08	-	-
`+good`	28.07	55.34	28.13	57.13	-	-	29.11	54.69	30.01	**51.68**	-	-
`+conf`	27.74	55.27	28.25	57.13	-	-	**29.51**	**54.12**	29.66	52.05	-	-
`+all`	28.17	55.07	27.92	57.45	**30.12**	56.70	28.76	54.98	**30.23**	51.80	31.74	53.52
`PPL`	**28.32**	55.09	27.99	57.46	-	-	28.76	55.27	30.03	51.81	-	-
`+good`	**28.34**	55.15	28.21	57.39	-	-	28.86	54.75	29.54	52.33	-	-
`+conf`	28.22	55.42	28.12	57.60	-	-	29.36	**54.16**	29.51	52.16	-	-
`+all`	27.89	55.25	27.87	57.74	30.03	**56.33**	**29.48**	54.34	29.76	51.95	**32.78**	**51.70**

Table 3: BLEU and TER results. Highest values in a given column appear in bold.

Comparing results between the out-of-domain and in-domain conditions makes the English to French situation look even more complex: there seems to be no marked regular differences between systems tuned with these two conditions (e.g. only +0.17 BLEU point improvement for `moses`). The situation is much clearer on French to English, where all systems benefit from in-domain tuning (e.g. +1.1 BLEU point improvement for `moses`).

Lastly, oracle tuning conditions yield again two different results: `moses` and the two adapted systems are indistinguishable in English to French, while on French to English we find `PPL` to be superior to `moses` (+0.67 BLEU point), itself superior to `IR` (+0.37 BLEU point). In all conditions, we note a substantial improvement over out-of-domain and in-domain tuning (e.g. for `PPL` up to 2.16 BLEU point over in-domain tuning on English to French and 3.02 on French to English). This last result clearly emphasizes the need for performing document-level adaptation for tuning, something that will be addressed in our future work. It also shows that improvements through better tuning are possible even for the (apparently difficult) English to French language pair, where in-domain tuning did not achieve a superior result than out-of-domain tuning.

Adaptation scenarios No instance-weighting scheme (`IR` or `PPL`) appears to clearly outperform the other: they stand in close range in the out-of-domain tuning condition, while `IR` has a slight advantage in the in-domain condition and the `PPL` oracle performs better in French to English. Our two additional features (`+good` and `+conf`) both proved useful under different situations; we can only observe a small tendancy of `conf` to perform better in the out-of-domain condition in French to English. Furthermore, their combination never leads to improvements on English to French, adding to the previously mentioned complexity of this language pair in our experiments.

Contrastive document-level evaluation Pair-wise contrastive results for a set of selected systems and the full range of tuning conditions are given in Table 4, where we consider differences over 0.5 BLEU point. These results allow us to obtain a more interpretable analysis of the comparison between any two systems. For instance, `IR+all` obtained a small advantage of +0.40 BLEU point over `moses` in the

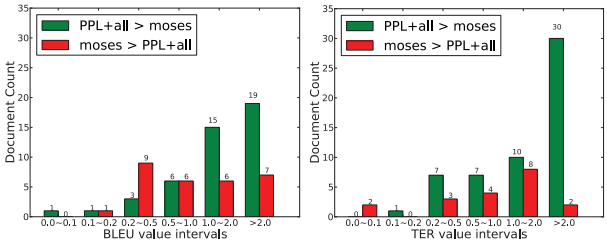


Figure 1: Document-level comparison for any-text translation: green bars (resp. red bars) show number of documents per BLEU (left side) or TER (right side) intervals for which `PPL+all` outperformed `moses` (resp. the other way round) in the out-of-domain tuning condition for French to English.

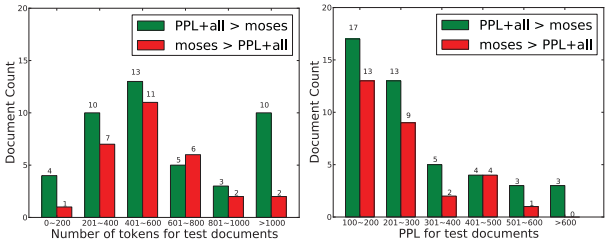


Figure 2: Document-level comparison depending on: (left) test document length (in tokens); (right): test document perplexity. Green bars (resp. red bars) show the number of documents per bins for which `PPL+all` outperformed `moses` (resp. the other way round) in the out-of-domain tuning condition for French to English translation.

French to English out-of-domain condition; however, this translates as 43.42% of documents for which `IR+all` outperforms `moses` (by 0.5 BLEU point or more), and 34.21% for the opposite. Computing those values on a large set of test documents would provide us with some probability that a given system would perform better at translating a new document than some other system, while corpus-based BLEU would give higher importance to longer documents, thus introducing a bias to their respective adaptation situation.

6. Discussion

Our experiments have shown that on-the-fly contextual adaptation could lead to significant improvements over several baselines, including one that exploits translation models derived from different domains. These results shed a new light on the complexity of the adaptation problem and provided concrete examples to illustrate the complexities of conditions under which adaptation can be successful. Furthermore, the oracle self-tuning condition demonstrated the sub-optimality of using large supposedly "in-domain" tuning sets, and our experiments more generally have provided arguments in favor of a document-level adaptation.

Our most salient result in relation to our target scenario of *any-text* translation is that when no well-adapted tuning set is available, i.e. in the out-of-domain tuning condition, the proposed instance-weighting schemes significantly improved over both a `moses` baseline and an adaptation baseline at the corpus-level (`2-tables`) in French to English translation. This condition is illustrated by the histogram on Figure 1, where it is clearly apparent that adaptation at the document-level was very successful in this case, using both BLEU and TER metrics (we note, for instance, that `PPL` improved the translation of 30 documents by a 2 or more TER points decrease compared to `moses`, while the opposite case was found for only 2 documents).

As the synthesis of all test documents shows significant improvement, we may question whether this result would be due to the length of the document, with the intuition that longer documents would allow for better adaptation[14], or to the similarity of documents, with the intuition that documents that have close matches in the training corpus should be translated better. Figure 2 displays results of a document-level contrastive comparison between the same systems of Figure 1 for document length and perplexity values intervals. If we obtain a very clear advantage for our adapted system for documents over 1,000 tokens, this result is also true (though based on a somewhat limited number of documents) for the shortest documents. Likewise, our adapted system clearly performs best for both test documents of low perplexity values, and test documents of high perplexity values.

The question remains of why the English to French language pair resulted in such a different set of observations. We have a number of hypotheses to account for this:

- For this language pair, the advantages of in-domain tuning vs. out-of-domain tuning were non-apparent for all systems, including our `moses` baseline, a fact that seems counter-intuitive.
- The perplexity values of both the in-domain and out-of-domain tuning sets w.r.t. to the training corpus are much higher on English than on French (see Table 1), suggesting that the English texts in our sets use a more "complicated" language. Note also that in the case of our test corpus and in-domain tuning corpus, English texts have significantly more out-of-vocabulary (**oov**) tokens. As the same texts are available in both languages, the differences cannot be attributed to thematic differences w.r.t. the training set.
- It may also be the case that English as an original language, resulting in a more complex language as opposed to when English is the result of translation (i.e. translationese), is less present in our training data. In fact, considering our Europarl data only (`epps`),

[14] Recall that in our settings documents from the training set were limited to single sentences, something we plan to improve on.

	$moses_{out}$	$moses_{out}^{2tables}$	IR_{out}^{+all}	PPL_{out}^{+all}	$moses_{in}$	$moses_{in}^{2tables}$	IR_{in}^{+all}	PPL_{in}^{+all}	$moses_{oracle}$	IR_{oracle}^{+all}	PPL_{oracle}^{+all}
$moses_{out}$	-	39.47	34.21	25.00	28.95	26.32	18.42	25.00	11.84	10.53	13.16
$moses_{out}^{2tables}$	38.16	-	28.95	27.63	25.00	22.37	14.47	23.68	10.53	11.84	10.53
IR_{out}^{+all}	43.42	39.47	-	18.42	30.26	32.89	14.47	26.32	9.21	10.53	10.53
PPL_{out}^{+all}	52.63	57.89	39.47	-	39.47	36.84	19.74	34.21	11.84	10.53	11.84
$moses_{in}$	57.89	55.26	50.00	38.16	-	36.84	26.32	30.26	10.53	9.21	9.21
$moses_{in}^{2tables}$	52.63	56.58	50.00	39.47	32.89	-	27.63	31.58	14.47	11.84	9.21
IR_{in}^{+all}	64.47	63.16	61.84	50.00	50.00	47.37	-	39.47	11.84	13.16	10.53
PPL_{in}^{+all}	57.89	61.84	52.63	44.74	43.42	42.11	21.05	-	10.53	11.84	10.53
$moses_{oracle}$	84.21	84.21	86.84	85.53	85.53	84.21	82.89	82.89	-	31.58	38.16
IR_{oracle}^{+all}	84.21	81.58	82.89	81.58	82.89	80.26	78.95	81.58	48.68	-	38.16
PPL_{oracle}^{+all}	81.58	85.53	82.89	80.26	80.26	81.58	80.26	81.58	48.68	39.47	-

Table 4: Document-level contrastive evaluation for French to English translation experiments. Numbers indicate the percentage of documents for which the system of the row outperformed the system of the column by more than the specified margin (BLEU > 0.5). Green background indicates that the system of the row outperformed the system of the column, while red indicates the opposite, and darker colors indicates larger differences.

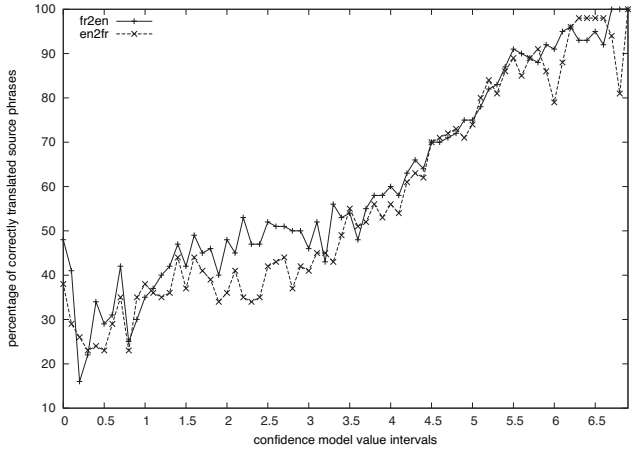


Figure 3: Percentage of correctly translated source phrases in the trace of the decoder for the `PPL+all` systems against score value intervals of the confidence model (`conf`).

> which correspond to the large majority of our training data, previous studies have shown that French as an original language is significantly more represented than English as an original language [22]. Experimenting with other corpora in which original language is known may help us to confirm this hypothesis.

Our adapted systems have recourse to sampling, and consequently do not use a reverse translation model [9], thus resulting in systems that may be built very efficiently, even for large data set conditions. Most previously published domain adaptation techniques cannot be applied directly to our studied scenario, as the availability of an in-domain training corpus is almost always assumed. Note that the `newsco` part of our training corpus was in fact "in-domain" w.r.t. our test documents. However, this corpus part was not identified as such, and our sampling strategies had no means to specifically access these data. The `2-tables` baseline system [1] is the only setting where we perform translation where subparts of the whole training data are known, identifying in particular an in-domain corpus: this situation obtained lower results than our systems under all conditions, indicating that the granularity of training corpus used was not appropriate and should be adapted.

Lastly, we assess whether our confidence model (section 4) is a good predictor of translation quality. Figure 3 plots the percentage of correctly translated source phrases in the trace of the decoder (counted as such when their target phrase matches the reference translation) against score intervals of the model. For our `PPL+all` systems, we observe a clear tendency to provide better translations for test phrases with higher confidence. This result clearly calls for a better handling of low-confidence phrases, e.g. by source-side paraphrasing [10].

7. Conclusion and future work

In this paper, we have studied a new scenario for Machine Translation that we called "any-text" translation, in which no in-domain training or development corpora can be identified in the general case. We have described an adaptation strategy that adapts translation models at the level of each input document by sampling and weighting training examples, and adds information about translation unit goodness and translation confidence. We found that our on-the-fly contextual adaptation significantly improves the results of French to English translation (up to 1.15 BLEU point improvement over `moses` and 1.02 BLEU over a corpus-level adaptation baseline (`2-tables`)). In comparison, results for the English to French pair do not reveal any clear gains. Some of our observations and hypotheses may pave the way to future experiments to determine under what conditions adaptation techniques can improve translation results. In particular, it turned out that our English documents were less similar to our training corpus than our French documents. The precise reasons for this situation should be investigated further.

We have introduced a document-level contrastive evaluation scheme (see Table 4), which offers a straightforward way to interpret and analyze the difference between any two

systems. Each reported value can be understood as the probability that one system would translate a document better (by some pre-defined margin using some evaluation metrics) than the other. The more input documents, the more accurate such probabilities will be. Those figures exhibit interesting conclusions: for instance, using a perfect tuning set at the document level allows to improve translation performance for more than 75% of documents for `moses` or our adapted systems over using a supposedly in-domain tuning set.

Given the large improvements obtained with the oracle tuning condition, we intend to study document-level adaptation schemes [18]. A better method of scoring the examples in the training corpus should be explored, for instance by taking more document context into account. More generally, we would like to recast the issue of instance weighting into one of determining the probability that a given training example is appropriate to translate a given test example in context: in this respect, textual similarity metrics such as `tf.idf` and perplexity values can only be used as features, in conjunction to other relevant features possibly indicating translation equivalence.

8. Acknowledgments

This work was partly funded by the European Union under the FP7 project META-NET (T4ME), Contract No. 249119.

9. References

[1] P. Koehn and J. Schroeder, "Experiments in Domain Adaptation for Statistical Machine Translation," in *WMT*, Prague, Czech Republic, 2007.

[2] G. Foster and R. Kuhn, "Mixture-model adaptation for SMT," in *WMT*, Prague, Czech Republic, 2007.

[3] G. Foster, C. Goutte, and R. Kuhn, "Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation," in *EMNLP*, Cambridge, USA, 2010.

[4] R. Sennrich, "Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation," in *EACL*, Avignon, France, 2012.

[5] B. Haddow and P. Koehn, "Analysing the Effect of Out-of-Domain Data on SMT Systems," in *WMT*, Montréal, Canada, 2012.

[6] S. Matsoukas, A.-V. I. Rosti, and B. Zhang, "Discriminative Corpus Weight Estimation for Machine Translation," in *EMNLP*, Singapore, 2009.

[7] K. Duh, K. Sudoh, and H. Tsukada, "Analysis of Translation Model Adaptation in Statistical Machine Translation," in *IWLST*, Paris, France, 2010.

[8] C. Callison-burch, C. Bannard, and J. Schroeder, "Scaling Phrase-Based Statistical Machine Translation to Larger Corpora and Longer Phrases," in *ACL*, Ann Arbor, USA, 2005.

[9] A. Lopez, "Tera-Scale Translation Models via Pattern Matching," in *COLING*, Manchester, UK, 2008.

[10] T. Onishi, M. Utiyama, and E. Sumita, "Paraphrase Lattice for Statistical Machine Translation," in *ACL, short papers*, Upsala, Sweden, 2010.

[11] A. Axelrod, X. He, and J. Gao, "Domain Adaptation via Pseudo In-Domain Data Selection," in *WMT*, Edinburgh, UK, 2011.

[12] G. Gascó, M.-A. Rocha, G. Sanchis-Trilles, J. Andrés-Ferrer, and F. Casacuberta, "Does more data always yield better translations?" in *EACL*, Avignon, France, 2012.

[13] U. Manber and G. Myers, "Suffix arrays: A new method for on-line string searches," *SIAM Journal of Computing*, vol. 22, no. 5, pp. 935–948, 1993.

[14] P. Koehn, F. J. Och, and D. Marcu, "Statistical Phrase-Based Translation," in *NAACL*, Edmonton, Canada, 2003.

[15] N. Tomeh, M. Turchi, G. Wisniewski, A. Allauzen, and F. Yvon, "How Good Are Your Phrases? Assessing Phrase Quality with Single Class Classification," in *IWSLT*, San Francisco, USA, 2011.

[16] J. Wuebker, A. Mauser, and H. Ney, "Training Phrase Translation Models with Leaving-One-Out," in *ACL*, Upsala, Sweden, 2010.

[17] M. Carpuat and D. Wu, "Context-Dependent Phrasal Translation Lexicons for Statistical Machine Translation," in *MT Summit*, Copenhagen , Denmark, 2007.

[18] L. Liu, H. Cao, T. Watanabe, T. Zhao, M. Yu, and C. Zhu, "Locally Training the Log-Linear Model for SMT," in *EMNLP*, Jeju Island, Korea, 2012.

[19] A. S. Hildebrand, M. Eck, S. Vogel, and A. Waibel, "Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval," in *EAMT*, Budapest, Hungary, 2005.

[20] K. Papineni, S. Roukos, T. Ward, and W.-j. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *ACL*, Philadelphia, USA, 2002.

[21] M. Snover, B. J. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A Study of Translation Edit Rate with Targeted Human Annotation," in *AMTA*, Cambridge, USA, 2006.

[22] B. Cartoni and T. Meyer, "Extracting directional and comparable corpora from a multilingual corpus for translation studies," in *LREC*, Istanbul, Turkey, 2012.

Author Index

hltc.cs.ust.hk/iwslt