IWSLT 2017
International Workshop on Spoken Language Translation

PROCEEDINGS

iwslt.org

14th-15th December, 2017
Tokyo, Japan

Proceedings of the

International Workshop on Spoken Language Translation

14th-15th December, 2017
Tokyo, Japan

Edited by
Sakriani Sakti
Masao Utiyama

Contents

FOREWORD

The International Workshop on Spoken Language Translation (IWSLT) is an annual scientific workshop, associated with an open evaluation campaign on spoken language translation, where both scientific papers and system descriptions are presented. The 14th International Workshop on Spoken Language Translation takes place in Tokyo, Japan on Dec. 14 and 15, 2017. Since 2004, the annual workshop has been held in Kyoto, Pittsburgh, Kyoto, Trento, Honolulu, Tokyo, Paris, San Francisco, Hong Kong, and Heidelberg, Lake Tahoe, Da Nang, Seattle, and this year in Tokyo.

One of the prominent research activities in spoken language translation is the work conducted by the Consortium for Speech Translation Advanced Research (C-STAR), which was an international partnership of research laboratories engaged in automatic translation of spoken language started in early 90s. The C-STAR members had initiated the first shared task-type Spoken Language Translation Workshop in 2004 and the IWSLT has been growing up with more participants and steering committee members.

The IWSLT includes scientific papers in dedicated technical sessions, either in oral or poster form. The contributions cover theoretical and practical issues in the field of Machine Translation (MT) in general and Spoken Language Translation (SLT), including Automatic Speech Recognition (ASR), Text-to-Speech Synthesis (TTS), and MT, 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 three members of the program committee and papers were selected based on their technical merit and relevance to the conference. In addition to core statistical machine translation papers, the technical program covers a wide spectrum of topics related to spoken language translation, ranging from issues related to real-time interpretation or to the translation of dialogs to more practical issues related to the integration of speech and translation technologies. Several important new annotated corpora will also be presented during the workshop. In summary, 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 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 organized in the manner of competition. 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. This year, the IWSLT evaluation offered a very challenging and appealing task on the spoken language translation of public speeches (TALK) in a variety of topics and dialogue, including a dedicated task to automatic speech recognition in order to cover the full pipeline of speech translation.

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 the system and the utilized resources. A survey of the evaluation campaigns is presented by the organizers.

Apart from the technical content of the conference, I hope all participants enjoy staying in Tokyo, one of the world's biggest metropolitan with cultural diversity.

Welcome to Tokyo!
Satoshi Nakamura,
Workshop Chair IWSLT 2017

ORGANIZERS

Workshop Chairs

- Satoshi Nakamura (NAIST, JP)
- Tetsunori Kobayashi (Uni-Waseda, JP)

Evaluation Chairs

- **Multilingual Task**
  - Mauro Cettolo (FBK, IT)
  - Marcello Federico (FBK, IT)
- **Dialog Task:**
  - Katsuhito Sudoh (NAIST, JP)
  - Koichiro Yoshino (NAIST, JP)
- **Lecture Task:**
  - Jan Niehues (KIT, DE)
  - Sebastian Stüker (KIT, DE)
- **Human Evaluation:**
  - Luisa Bentivogli (FBK, IT)

Program Chairs

- Sakriani Sakti (NAIST, JP)
- Masao Utiyama (NICT, JP)

Publicity Chairs

- Marcello Federico (FBK, IT)
- Koichiro Yoshino (NAIST, JP)

Local and Financial Chairs

- Masakiyo Fujimoto (NICT, JP)

Steering Committee

- Marcello Federico (FBK, IT)
- Masakiyo Fujimoto (NICT, JP)
- Will Lewis (MSR, USA)
- Chi Mai Luong (IOIT, VI)
- Joseph Mariani (CNRS-LIMSI, FR)
- Satoshi Nakamura (NAIST, JP)
- Hermann Ney (RWTH, DE)
- Sebastian Stüker (KIT, DE)
- Alex Waibel (KIT, DE & CMU, USA)
- Francois Yvon (CNRS-LIMSI, FR)

Program Committee

Allexandre Allauzen (LIMSI, FR)
Boxing Chen (NRC-CNRC, CA)
ChenChen Ding (NICT, JP)
Eunah Cho (Amazon, USA)
Eva Hasler (Uni-Edinburgh, UK)
Francisco Casacuberta (UPV, SP)
Francisco J. Guzman (Facebook, USA)
François Yvon (LIMSI, FR)
Gilles Adda (LIMSI, FR)
Graham Neubig (CMU, USA)
Guillaume Wisniewski (LIMSI, FR)
Hajime Tsukada (TUT, JP)
Hany Hassan (Microsoft, USA)
Isabel Transcoso (INESC-ID, PT)
Joachim Bogaert (CrossLang, BE)
Katsuhito Sudoh (NAIST, JP)
Kenji Imamura (NICT, JP)
Kevin Duh (JHU, USA)
Laurent Besacier (LIG, FR)
Luisa Bentivogli (FBK, IT)
Mamoru Komachi (TMU, JP)
Marion Weller (Uni-Stuttgart, DE)
Markus Freitag (IBM, USA)
Markus Müller (KIT, DE)
Markus Nussbaum-Thom (IBM, USA)
Masaaki Nagata (NTT, JP)
Masao Utiyama (NICT, JP)
Matthias Huck (LMU, DE)
Matthias Sperber (KIT, DE)
Mauro Cettolo (FBK, IT)
Michael Heck (NAIST, JP)
Michael Paul (ATR-Trek, JP)
Mohammed Mediani (KIT, DE)
Pavel Golik (RWTH Aachen, DE)
Qun Liu (DCU, IE)
Rico Sennrich (Uni-Edinburgh, UK)
Rui Wang (NICT, JP)
Sakriani Sakti (NAIST, JP)
Sara Stymne (Uni-Uppsala, SE)
Sebastian Stüker (KIT, DE)
Stephan Peitz (Apple, USA)
Teresa Herrmann (EU Commission, LU)
Thai-Son Nguyen (KIT, DE)
Thanh-Le Ha (KIT, DE)
Toshiaki Nakazawa (JST, JP)
Yuki Arase (Uni-Osaka, JP)
Yves Lepage (Uni-Waseda, JP)
Zoltan Tuske (RWTH Aachen, DE)

ACKNOWLEDGEMENTS

Sponsorships & Endorsements

We are grateful to our silver sponsor:

ATR-Trek

Our gratitude also goes to our bronze sponsor:

kites
Karlsruhe Information Technology Solutions

Endorsed by

- Asia-Pacific Association for Machine Translation (AAMT)
- The Association for Natural Language Processing (ANLP)
- The Acoustical Society of Japan (ASJ)
- The Institute of Electronics, Information and Communication Engineers (IEICE)

In cooperation with

- The Information Processing Society of Japan (IPSJ), SIG-SLP

PROGRAM

Thursday, December 14th, 2017

08:15-09:00	**WORKSHOP REGISTRATION**
09:00-09:30	**WELCOME REMARKS** (1F 102 Presentation Room) Satoshi Nakamura (Workshop Chair) Alex Waibel (IWSLT Steering Committee) Fumihiko Tomita (NICT Vice President)
09:30-10:30	**INVITED TALK 1** Chair: Alex Waibel (1F 102 Presentation Room) "The Move to Neural Machine Translation at Google", Mike Schuster (Google, USA)
10:30-11:00	*Coffee Break*
11:00-12:00	**REPORT** Chair: Katsuhito Sudoh (1F 102 Presentation Room) "Overview of the IWSLT 2017 Evaluation Campaign", Mauro Cettolo (FBK, Italy), Marcello Federico (FBK, Italy), Luisa Bentivogli (FBK, Italy), Jan Niehues (KIT, Germany), Sebastian Stüker (KIT, Germany), Katsuhito Sudoh (NAIST, Japan), Koichiro Yoshino (NAIST, Japan), Christian Federmann (Microsoft, USA)
12:00-13:30	*Lunch*
13:30-15:30	**POSTER and EXHIBITION SESSION** Chair: Sakriani Sakti (2F 205 Presentation Room) ■ **Scientific Papers:** P01: "Neural Machine Translation Training in a Multi-Domain Scenario", Hassan Sajjad (QCRI-HBKU, Qatar), Nadir Durrani (QCRI-HBKU, Qatar), Fahim Dalvi (QCRI-HBKU, Qatar), Yonatan Belinkov (MIT CSAIL, USA) and Stephan Vogel (QCRI-HBKU, Qatar) P02: "Domain-independent Punctuation and Segmentation Insertion", Eunah Cho (KIT, Germany), Jan Niehues (KIT, Germany), and Alex Waibel (KIT, Germany) P03: "Synthetic Data for Neural Machine Translation of Spoken-Dialects", Hany Hassan (MSR AI, USA), Mostafa Elaraby (MSR AI, USA) and Ahmed Tawfik (MSR AI, USA) P04: "Toward Robust Neural Machine Translation for Noisy Input Sequences", Matthias Sperber (KIT, Germany), Jan Niehues (KIT, Germany), and Alex Waibel (KIT, Germany) P05: "Monolingual Embeddings for Low Resourced Neural Machine Translation", Mattia Antonino Di Gangi (FBK & Uni. Trento, Italy) and Marcello Federico (FBK, Italy) P06: "Effective Strategies in Zero-Shot Neural Machine Translation", Thanh-Le Ha (KIT, Germany), Jan Niehues (KIT, Germany) and Alex Waibel (KIT, Germany)

Thursday, December 14th, 2017

13:30-15:30	■ **System Description Papers:** P07: "Going beyond zero-shot MT: combining phonological, morphological and semantic factors. The UdS-DFKI System at IWSLT 2017", Cristina España-Bonet (UdS & DFKI GmbH, Germany) and Josef van Genabith (UdS & DFKI GmbH, Germany) P08: "The Samsung and University of Edinburgh's submission to IWSLT17", Pawel Przybysz (Samsung, Poland), Marcin Chochowski (Samsung R&D Center, Poland), Rico Sennrich (Uni. Edinburg, UK), Barry Haddow (Uni. Edinburg, UK) and Alexandra Birch (Uni. Edinburg, UK) P09: "The RWTH Aachen Machine Translation Systems for IWSLT 2017", Parnia Bahar (RWTH Aachen, Germany), Jan Rosendahl (RWTH Aachen, Germany), Nick Rosenbach (RWTH Aachen, Germany), and Hermann Ney (RWTH Aachen, Germany) P10: "FBK's Multilingual Neural Machine Translation System for IWSLT 2017", Surafel M. Lakew (FBK & Uni. Trento, Italy), Quintino F. Lotito (Uni. Trento, Italy), Marco Turchi (FBK, Italy), Matteo Negri (FBK, Italy), and Marcello Federico (FBK, Italy) P11: "KIT´s Multilingual Neural Machine Translation systems for IWSLT 2017", Ngoc-Quan Pham (KIT, Germany), Matthias Sperber (KIT, Germany), Elizabeth Salesky (KIT, Germany), Thanh-Le Ha (KIT, Germany), Jan Niehues (KIT, Germany), and Alex Waibel (KIT, Germany & CMU, USA) ■ **Exhibition:** Ex1: "Multilingual Translation System," Panasonic Corporation, Japan Ex2: "Superfast Online Speech Recognition with Offline Translation for CH/EN/JA/KO," Kodensha Co., Ltd., Japan Ex3: "NAIST Japanese-to-English Simultaneous Interpretation System," NAIST, Japan Ex4: "VoiceTra: Multilingual Speech Translation Application," NICT, Japan
15:30-16:00	*Coffee Break*
16:00-17:30	**ORAL SESSION 1** Chair: Marcello Federico (1F 102 Presentation Room) (each 30min: 25min presentation + 5min Q&A) O1-1: "Towards Better Translation Performance on Spoken Language", Chao Bei (GTCOM China) and Hao Zong (GTCOM, China) O1-2: "Kyoto University MT System Description for IWSLT 2017", Raj Dabre (Kyoto Uni., Japan), Fabien Cromieres (JST, Japan) and Sadao Kurohashi (Kyoto Uni., Japan) O1-3: "The 2017 KIT IWSLT Speech-to-Text Systems for English and German", Thai Son Nguyen (KIT, Germany), Markus Müller (KIT, Germany), Matthias Sperber (KIT, Germany), Thomas Zenkel (KIT, Germany), Sebastian Stüker (KIT, Germany), and Alex Waibel (KIT, Germany)
18:00-	**SOCIAL EVENT DINNER** Banquet at RIHGA Royal Hotel Tokyo 3F Garden Terrace

Friday, December 15th, 2017

08:30-09:30	**WORKSHOP REGISTRATION**
09:30-10:30	**INVITED TALK 2** Chair: Satoshi Nakamura (1F 102 Presentation Room) Simultaneous Interpreting, Cognitive Constraints, and Information Structure Akira Mizuno (Aoyama Gakuin University & JAITS, Japan)
10:30-11:00	*Coffee Break*
11:00-12:00	**ORAL SESSION 2** Chair: Luisa Bentivogli (1F 102 Presentation Room) (each 30min: 25min presentation + 5min Q&A) O2-1: "CharCut: Human-Targeted Character-Based MT Evaluation with Loose Differences", Adrien Lardilleux (Fujitsu & DGT, Luxembourg) and Yves Lepage (Waseda Uni., Japan) O2-2: "Data Selection with Cluster-Based Language Difference Models and Cynical Selection", Lucía Santamaría (Amazon, Germany) and Amittai Axelrod (Amazon, USA)
12:00-13:30	*Lunch*
13:30-15:00	**ORAL SESSION 3** Chair: Sebastian Stüker (1F 102 Presentation Room) (each 30min: 25min presentation + 5min Q&A) O3-1: "Continuous Space Reordering Models for Phrase-based MT", Nadir Durrani (QCRI-HBKU, Qatar) and Fahim Dalvi (QCRI-HBKU, Qatar) O3-2: "Evolution Strategy based Automatic Tuning of Neural Machine Translation Systems", Hao Qin (Tokyo Inst. Tech, Japan), Takahiro Shinozaki (Tokyo Inst. Tech, Japan), and Kevin Duh (JHU, USA) O3-3: "Improving Zero-Shot Translation of Low-Resource Languages", Surafel M. Lakew (FBK & Uni. Trento, Italy), Quintino F. Lotito (Uni. Trento, Italy), Matteo Negri (FBK, Italy), Marco Turchi (FBK, Italy), and Marcello Federico (FBK, Italy)
15:00-15:30	*Coffee Break*
15:30-17:00	**PANEL DISCUSSION** Chair: Jan Niehues (1F 102 Presentation Room) Panelists: Marcello Federico (FBK, Italy), Satoshi Nakamura (NAIST, Japan), Hermann Ney (RWTH Aachen, Germany), Mike Schuster (Google, USA), Alex Waibel (KIT, Germany & CMU, USA) Topic: 1. New Trends in Spoken Language Translation 2. The Future of the IWSLT Evaluation Campaign
17:00-17:30	**CLOSING REMARKS + ANNOUNCEMENTS** IWSLT 2017 Best Paper Award and Next IWSLT

KEYNOTES

The move to Neural Machine Translation at Google

Mike Schuster (Google, USA)

Abstract:

Machine learning and in particular neural networks have made great advances in the last few years for products that are used by millions of people, most notably in speech recognition, image recognition and most recently in neural machine translation. Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation, with the potential to overcome many of the weaknesses of conventional phrase-based translation systems. Unfortunately, NMT systems are known to be computationally expensive both in training and in translation inference. Also, most NMT systems have difficulty with rare words. These issues have hindered NMT's use in practical deployments and services, where both accuracy and speed are essential. In this work, we present GNMT, Google's Neural Machine Translation system, which addresses many of these issues. The model consists of a deep LSTM network with 8 encoder and 8 decoder layers using attention and residual connections. To accelerate final translation speed, we employ low-precision arithmetic during inference computations. To improve handling of rare words, we divide words into a limited set of common sub-word units for both input and output. On the WMT'14 English-to-French and English-to-German benchmarks, GNMT achieves competitive results to state-of-the-art. Using human side-by-side evaluations it reduces translation errors by more than 60% compared to Google's phrase-based production system. The new Google Translate was launched in late 2016 and has improved translation quality significantly for all Google users.

Biography:

Dr. Mike Schuster graduated in Electric Engineering from the Gerhard-Mercator University in Duisburg, Germany in 1993. After receiving a scholarship he spent a year in Japan to study Japanese in Kyoto and Fiber Optics in the Kikuchi laboratory at Tokyo University. His professional career in machine learning and speech brought him to Advanced Telecommunications Research Laboratories in Kyoto, Nuance in the US and NTT in Japan where he worked on general machine learning and speech recognition research and development after getting his PhD at the Nara Institute of Science and Technology. Dr. Schuster joined the Google speech group in the beginning of 2006, seeing speech products being developed from scratch to toy demos to serving millions of users in many languages over the next eight years, and he was the main developer of the original Japanese and Korean speech recognition models. He is now part of the Google Brain group which focuses on building large-scale neural network and machine learning infrastructure for Google and has been working on infrastructure with the TensorFlow toolkit as well as on research, mostly in the field of speech and translation with various types of recurrent neural networks. In 2016 he led the development of the new Google Neural Machine Translation system, which reduced translation errors by more than 60% compared to the previous system.

Simultaneous Interpreting, Cognitive Constraints, and Information Structure

Akira Mizuno
(Aoyama Gakuin University, Japan)

Abstract:

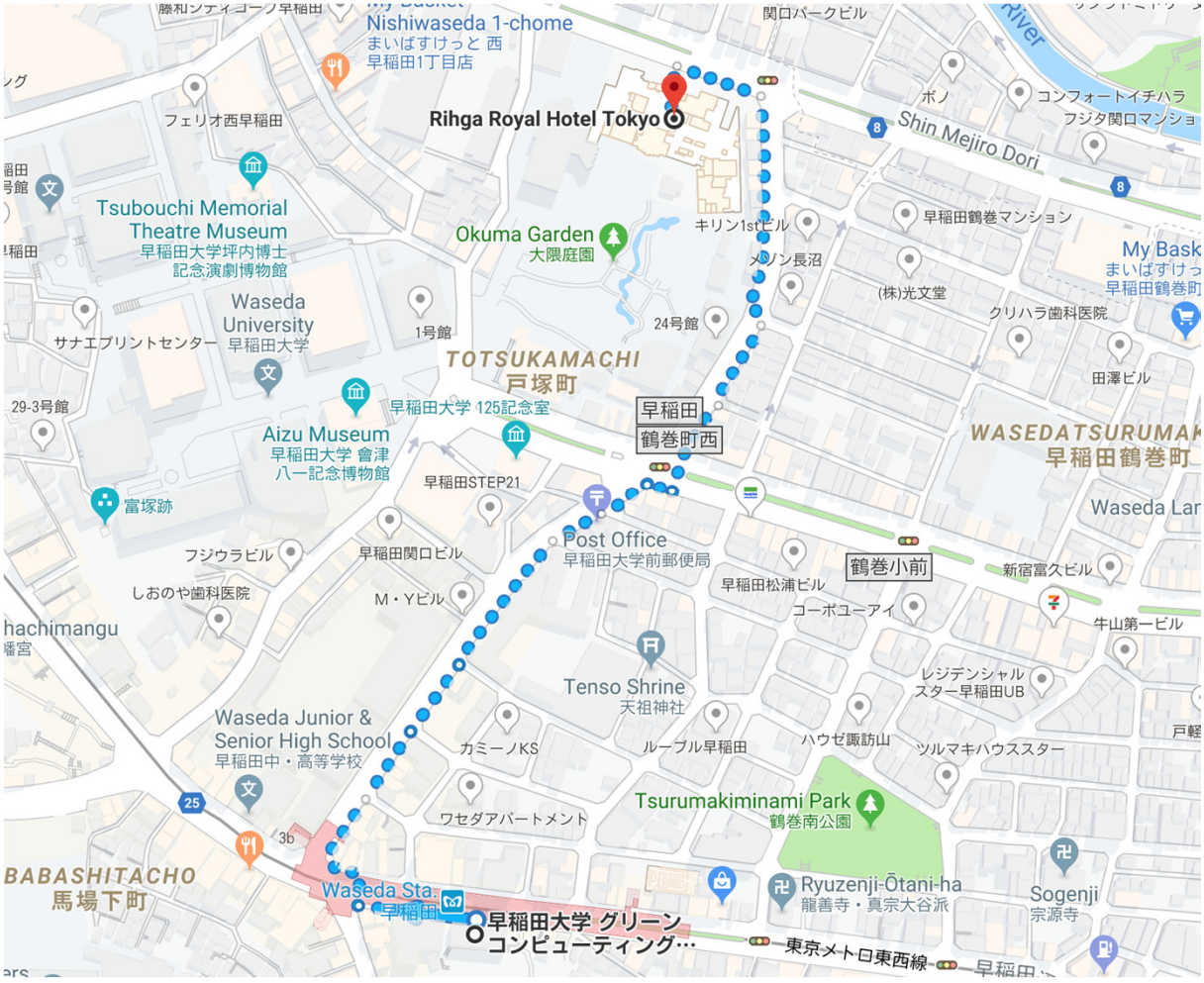
Simultaneous interpreting involves heavy cognitive load, which becomes heavier when interpreters interpret simultaneously between structurally different languages such as Japanese and English. The cognitive load can be measured by the number of chunks held in the focus of attention of the Cowan's model of working memory. An analysis of a small corpus of simultaneous interpreting between English and Japanese indicated that simultaneous interpreters frequently made use of translation strategies in order not to surpass the capacity of working memory. These strategies, different from traditional translation method which frequently involves word order reversal, seem to have intended to perform "a minimum reverse integration". In this talk, I will indicate that these are not ad-hoc strategies but more appropriate translation method than the traditional method, which can be supported by the theories of information structure and contribute to the research of machine translation.

Biography:

Akira Mizuno is a former professor of Aoyama Gakuin University and the President of the Japan Association for Interpreting and Translation Studies (JAITS). He has been involved in conference interpreting and broadcast interpreting since 1988. His main interest is Interpreting and Translation Studies, Functional Linguistics, and Cognitive Science. In 2010, he co-edited and co-authored Translation Theories in Japan and in 2015, published Theories of Simultaneous Interpreting Cognitive Constraints and Translation Strategies.

MAPS

Map and direction to the banquet place
(10-min walk)



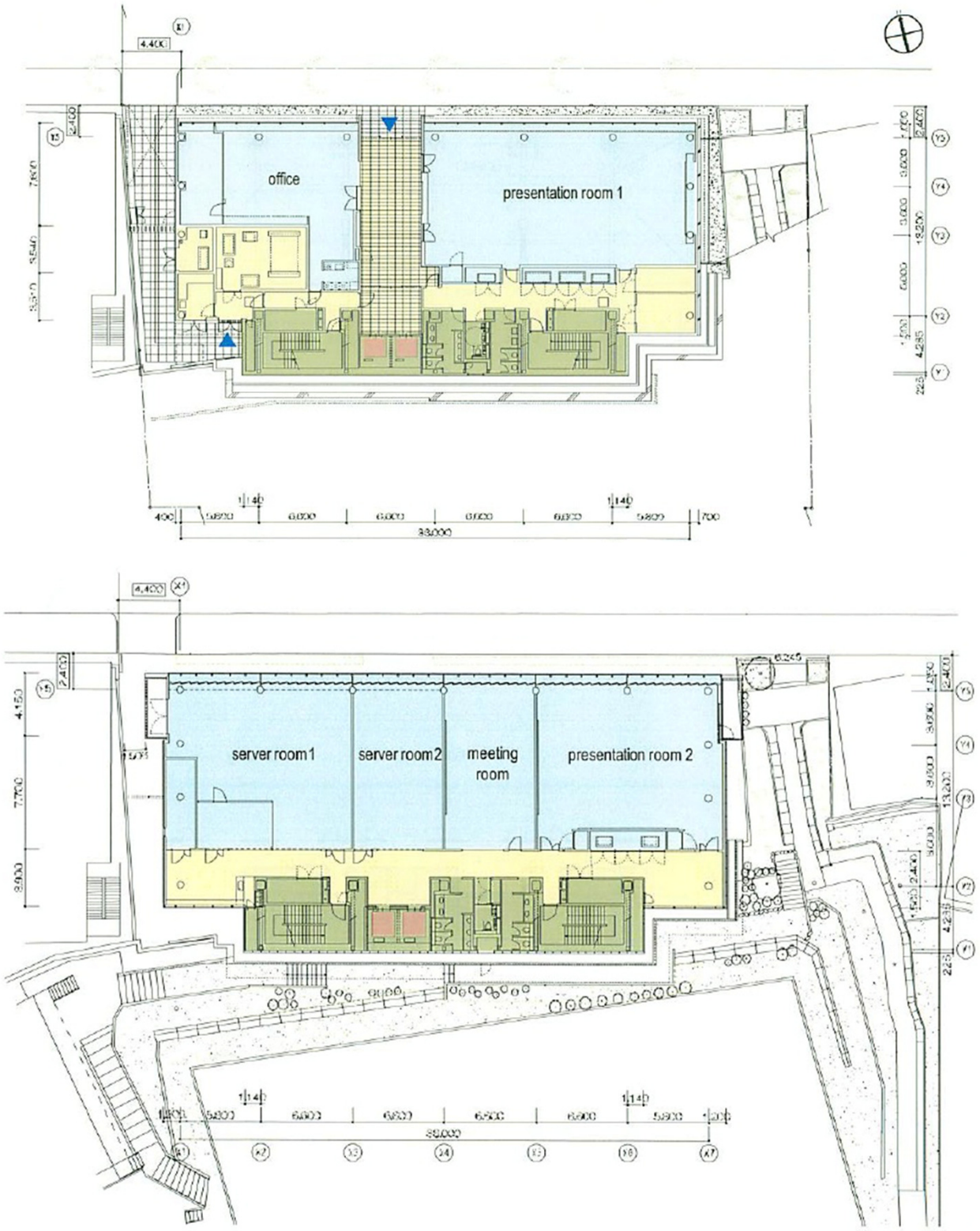
From RIHGA Royal Hotel Tokyo:

To Tokyo Metro Waseda Station: 10-minute walk

To JR Takadanobaba Station

- Free shuttle bus from hotel bus stop: 09:00 to 21:00, every 00/30 minutes
- 8 minutes by taxi (900 - 1,000 JPY)
- 30-minute walk

Floor Maps



An ID card is required to access to the 2F 205 presentation room.
If you want to access to the room, please ask the registration desk.

EVALUATION CAMPAIGN

Overview of the IWSLT 2017 Evaluation Campaign

M. Cettolo[(1)] *M. Federico*[(1)] *L. Bentivogli*[(1)] *J. Niehues*[(2)]
S. Stüker[(2)] *K. Sudoh*[(3)] *K. Yoshino*[(3)] *C. Federmann*[(4)]

[(1)] FBK - Trento, Italy
[(2)] KIT - Karlsruhe, Germany
[(3)] NAIST - Nara, Japan
[(4)] Microsoft AI+Research - Redmond, WA, USA

Abstract

The IWSLT 2017 evaluation campaign has organised three tasks. The Multilingual task, which is about training machine translation systems handling many-to-many language directions, including so-called zero-shot directions. The Dialogue task, which calls for the integration of context information in machine translation, in order to resolve anaphoric references that typically occur in human-human dialogue turns. And, finally, the Lecture task, which offers the challenge of automatically transcribing and translating real-life university lectures. Following the tradition of these reports, we will described all tasks in detail and present the results of all runs submitted by their participants.

1. Introduction

Spoken language translation (SLT) is the subfield of machine translation (MT) that deals with the translation of spoken language. Spoken language, besides differing from written language from a linguistic point of view [1], also implies that it is processed under form of a transcript, either manually created and cleaned or generated via automatic speech recognition (ASR) and thus possibly noisy.

Since 2004, the International Workshop on Spoken Language Translation has been organizing a yearly evaluation campaign in conjunction with a scientific workshop. The main purpose of the evaluation campaigns is to offer to researchers working in the fields of MT and ASR challenging tasks to work on, as well as providing for them a venue where to present, compare and discuss their results. Moreover, in order to offer a friendly environment for scientific exchange, the spirit of our evaluation has never been competitive, but rather collaborative.

The tasks offered during the last 13 years have followed the trend and progress in the field of MT and ASR. In the first years, SLT tasks focused on restricted domains, with low language complexity. Then, following the steady rise of statistical methods and computing power, less restricted and more data intensive tasks were progressively introduced, up to the translation of TED Talks and university lectures. However, in order to keep the participation barrier low, IWSLT has also always offered at the same time tasks that were affordable to small teams or even students with limited access to computing resources. Another distinctive feature of IWSLT is the variety of translation directions covered over the years, which include many American, European and Asian languages.

We believe that scientific communication is greatly facilitated when all experimental conditions are set in advance and shared by everyone. This is the reason why, since the begin, IWSLT has organized shared tasks in which all the training data, experimental conditions and evaluation metrics were set and provided in advance.

This year, the IWSLT evaluation campaign has focused on three tasks, which address rather different and orthogonal open issues in MT, in general, and spoken language translation, in particular. The Multilingual task investigates the possibility of machines to simultaneously learn to translate across multiple languages, given parallel data (TED Talks) that only partially covers the tested translation directions. The Dialogues task targets instead the challenge for MT to consider the context of the input (utterance transcript) that has to be translated, in order to resolve the translation of pronouns and other empty categories. Finally, the Lecture task addresses the challenge of automatically transcribing and translating real-life university lectures, in contrast of staged and well-rehearsed talks, such as the TED Talks.

The following sections describe in great details each task, including the benchmark that has been developed around it and the outcome of the evaluation. One specific section will be devoted to report on the manual evaluation that was car-

ried out for the Multilingual task. An appendix concludes this report, which contains all the tables with the results of all the submitted runs.
Finally, this year we have witnessed, unfortunately, a significant drop in the number of participants to the evaluation campaign (see Table 1). For this reason, part of the open discussion that will take place at the workshop will regard this issue. Our aim will be to understand if the lack of participation has a contingent nature or expresses a shift of interest in the community. In either cases, as organizers, we will see if and how we can find better ways to serve the community.

2. Multilingual Task

2.1. Definition

The introduction of translation of TED talks in IWSLT evaluation campaigns dates back to 2010. The task continues to receive attention by the research community because it is challenging but at the same time manageable. In fact, besides being a realistic exercise, the variety of topics dealt with in TED talks can be considered unlimited, which is an interesting research issue in itself. On the other hand, the truly "in-domain" training data, that is the set of transcriptions and translations of TED talks only, amount to just few million words per side, making the training/adaptation of even neural engines reasonably fast.

With the aim of keeping the task interesting and to follow current trends in research and industry, this year we proposed the multilingual translation between any pair of languages from {Dutch, English, German, Italian, Romanian} by means of an engine trained with either only in-domain data (*small data condition*) or a long list of permissible resources (*large data condition*). In addition, within the small condition, we proposed the *zero-shot* translation for the pairs Dutch-German and Italian-Romanian, in both directions. Zero-shot means to translate with a multilingual engine between language pairs that have never seen in this combination during training. In the specific, the zero-shot engine could be trained on the in-domain training data of all the other 16 pairs, but not of those four pairs. Training data synthesis from the 16 pairs and pivoting were explicitly forbidden, in order to force the adoption of methods that deal with the problem instead of getting around it. The zero-shot paired languages are from the same family (West-Germanic and Romance, respectively) in the hope that they can somehow leverage from their common origin.

A set of unofficial standard bilingual tasks between English from one side and {Arabic, Chinese, French, German, Japanese, Korean} on the other were proposed as well to keep continuity with past editions.

2.2. Data

In-domain training, development and evaluation sets were supplied through the website of the WIT[3] project [9], while out-of-domain training data were linked in the workshop's website. With respect to edition 2016 of the evaluation campaign, some of the talks added to the TED repository during the last year have been used to define the evaluation sets (tst2017), while the remaining new talks have been included in the training sets.

Two development sets (dev2010 and tst2010) are either the same of past editions - when available - or have been built upon the same talks - for pairs never proposed in the past.

Table 2 provides statistics on in-domain texts supplied for training, development and evaluation purposes, averaged on the 20 language pairs.

Concerning the unofficial bilingual task, besides the tst2017 evaluation set, we asked to translate the progressive tst2016 test set as well.

2.3. Evaluation

Participants had to provide MT outputs of the test sets in NIST XML format. Outputs had to be case-sensitive, detokenized and punctuated. The quality of translations was measured both automatically, against human translations created by the TED open translation project, and via human evaluation (Section 5). Case sensitive automatic scores were calculated with the three automatic standard metrics BLEU, NIST, and TER, as implemented in mteval-v13a.pl[1] and tercom-0.7.25[2], by calling:

- `mteval-v13a.pl -c`
- `java -Dfile.encoding=UTF8 -jar tercom.7.25.jar -N -s`

Detokenized texts were used, since the two scoring scripts apply their own internal tokenizers.

In order to allow participants to evaluate their progresses automatically and under identical conditions, an evaluation server was set up. Participants could submit the translation of any development set to either a REST Webservice or through a GUI on the web, receiving as output BLEU, NIST and TER scores computed as described above.

The evaluation server was utilized by the organizers for the automatic evaluation of the official submissions. After the evaluation period, the

[1] http://www.itl.nist.gov/iad/mig/tests/mt/2009/
[2] 5http://www.cs.umd.edu/ snover/tercom/

Table 1: List of Participants

FBK	Fondazione Bruno Kessler, Italy [2]
GTCT	Global Tone Communication Technology Co. Ltd, China[3]
KIT	Karlsruhe Institute of Technology, Germany [4]
KYOTO	Kyoto University, Japan [5]
RWTH	Rheinisch-Westfälische Technische Hochschule, Germany [6]
UEDIN	University of Edinburgh, United Kingdom [7]
UDSDFKI	Universität des Saarlandes and Deutsche Forschungszentrum für Künstliche Intelligenz, Germany [8]

Table 2: Average size of bilingual resources made available for the 20 language pairs of the multilingual task.

data set	sent	tokens source	tokens target	talks
train	160k	3.99M	3.99M	1749
dev2010	940	18,8k	18,8k	8
tst2010	1,660	30,0k	30,0k	11
tst2017	1,146	19,8k	19,8k	10

evaluation of test sets was allowed to all participants as well.

2.4. Submissions

We received 9 primary multilingual submissions from 5 different sites, distributed according to training conditions as follows: 4 on small-data, 4 on zero-shot and 1 on large-data; in addition, 3 small-data, 2 zero-shot and 1 large-data contrastive runs were submitted. One out of those five participants also sent a bilingual run on Chinese-English, while two other participants provided their runs on German-English bilingual task.

The total number of test sets evaluated for the multilingual task was then 300 (180 primary, 120 contrastive), while as far as the bilingual tasks are concerned, 12 translations were scored.

2.5. Automatic results

The automatic scores computed on the 2017 official test set for each participant are shown in Appendix A. The two uppermost tables concern the four zero-shot language pairs, where scores of all multilingual submissions are provided.

Table 3 reports the automatic scores of the 9 primary multilingual submissions averaged on the four directions involving the zero-shot condition. Despite being questionable, the average operation allows to synthesize some general outcomes in a easier way than looking at the many tables of the appendix:

- as proved by KYOTO, zero-shot systems ("ML ZS") can well compete with those trained including data of the language pairs they are tested on ("ML SD")
- also other labs were able to develop zero-shot systems reasonably good with respect to their best systems, endorsing the general feasibility of zero-shot translation
- KIT, the only lab that submitted runs for both small- and large-data conditions, was able to reach the highest MT quality by using more data for training, but not by far. Such performance proximity could be due to multilinguality, which allows the weaker condition (SD) to handle sparsity, problem that does not affect too much the LD engine. In other words, multilinguality seems to represent an effective solution to data sparsity, alternative to the use of large out-of-domain data sets.

Table 3: Automatic scores of the primary multilingual submissions averaged on the four zero-shot language pairs.

system	**cond.**	BLEU	NIST	TER
FBK	ML SD	19.54	5.432	62.81
	ML ZS	17.26	5.077	65.29
GTCT	ML ZS	19.40	5.343	63.27
KIT	ML SD	20.97	5.716	60.38
	ML LD	21.13	5.765	59.77
KYOTO	ML SD	20.60	5.621	61.54
	ML ZS	20.55	5.573	61.84
UDSDFKI	ML SD	19.06	5.342	64.26
	ML ZS	17.10	5.088	65.81

Table 4 reports the automatic scores of the 9 primary multilingual submissions averaged on the 16 directions other than the zero-shot. For these directions, the ML ZS systems are not at all "zero-shot" systems, but simply multilingual systems trained on parallel data for 16 pairs, including that which they are tested on. Therefore, the table compares multilingual systems trained

on either 20 or 16 pairs. In one case (FBK) the ML SD system is better than the ML ZS, in another (KYOTO) it is the opposite, while in the third case (UDSDFKI) they perform equally; no general conclusion can be drawn for now but the issue deserves further investigation.

Table 4: Automatic scores of the primary multilingual submissions averaged on the 16 non zero-shot language pairs.

system	cond.	BLEU	NIST	TER
FBK	ML SD	22.31	5.818	59.89
	ML ZS	21.89	5.760	60.36
GTCT	ML ZS	24.46	6.112	57.61
KIT	ML SD	24.07	6.139	57.12
	ML LD	24.42	6.191	56.56
KYOTO	ML SD	23.73	6.059	58.00
	ML ZS	24.10	6.083	57.78
UDSDFKI	ML SD	21.69	5.764	60.75
	ML ZS	21.63	5.749	60.89

3. Dialogue Task

3.1. Definition

Despite the recent advances of machine translation technologies, their effectiveness has not been investigated well by highly context-dependent situations such as dialogues. One typical problem in the translation of dialogues is the existence of empty categories [10], especially in pro-drop source languages such as Chinese, Japanese, and Korean. Translating such empty categories is also problematic other than dialogues [11], but it becomes very severe in natural conversations. A past shared task in IWSLT [12] included translator-assisted dialogues in a travel domain. A Chinese-English-Japanese corpus related to Olympic games, a.k.a. HIT corpus [13], which were also used for IWSLT shared task [14], also included some dialogues in a travel domain. These travel domain corpora have been widely used for spoken language translation studies, but these dialogues are in very limited situations and not necessarily natural conversations.

We focus on different types of dialogues called attentive listening, where a listener listens to people attentively about what they think. Conversations in attentive listening are not task-oriented so it is not easy to assume pre-defined information that can help to understand and translate them.

Table 5: Corpus statistics in the numbers of utterances (excluding backchannel and filler ones) and words. #words is based on tokenization using KyTea (ja) and Moses tokenizer (en).

	#utt.	#words (ja)	#words (en)
dev. (#1-#5)	1,476	25,780	16,235
test (#6-11)	1,510	31,857	20,099

3.2. Data

In-domain development and test data are based on the attentive listening corpus developed in NAIST [15], whose recorded and transcribed dialogues were originally in Japanese and then translated into English. We chose eleven dialogues for this task including 2,986 utterances, excluding 2,904 utterances just with backchannel and fillers. The translators were asked to translate literally with least supplement of empty categories by pronouns that were required grammatically. They could also refer to the original dialogue transcriptions with backchannel and fillers for taking the dialogue context into account.

In the recorded dialogues, many participants spoke Kansai dialect of Japanese. This caused some difficulties on Japanese morphological analyses and translation. We conducted rewriting of such expressions into standard Japanese by four annotators.

Table 5 shows the statistics of the development and test data. Since there are no other in-domain resources for this task, we did not provide any training data; participants can use any external Japanese-English resources.

3.3. Evaluation

Unfortunately we received no submissions for this task while some task registrations were made. The development and evaluation data can be obtained from the evaluation campaign website[3] for future studies.

4. Lecture Task

4.1. Definition

The lecture task covered two tracks: ASR and SLT. In the ASR track, the participants should transcribe the English and German audio. In the SLT track, these transcriptions should be translated into the other language.

[3] https://sites.google.com/site/iwsltevaluation2017/Dialogues-task

4.1.1. Data

The evaluation data for the lecture task (*tst2017*) consists of German and English recordings of talks and lectures.

The English data that participants were asked to recognize and translate consists in part of TED talks as in the years before, and in part of real-life lectures and talks that have been mainly recorded in lecture halls at KIT and Carnegie Mellon University. TED talks are challenging due to their variety in topics, but are very benign as they are very thoroughly rehearsed and planned, leading to easy to recognize and translate language. The real-life lectures that we included in the test set are more difficult to process as reflected by the scores on them in comparison to the scores on the TED talks. As this is the first edition in which we offer real-life lectures, and the amount of available test data is limited, we included both, TED talks and real lectures in the English evaluation data.

The German data consisted solely of German real-live lectures given at KIT.

4.1.2. ASR

In the ASR track participants were asked to recognize the unsegmented audio of the lectures and transcribe them automatically into the spoken word sequence. The training data for the acoustic model was limited to publicly available data, while the training data for the language model was restricted to a known list of corpora. But participants could suggest corpora to include in the list.

4.1.3. SLT

The SLT track covered the translations of university lectures and TED talks from English to German and the translation of university lectures from German to English. The participants should translate from the English and German audio signal. The challenge of this translation task is the necessity to deal with automatic, and in general error prone, transcriptions of the audio signal, instead of correct human transcriptions. Furthermore, for the lecture tasks no manual segmentation into sentences was provided. Therefore, participants needed to develop methods to automatically segment the automatic transcript and insert punctuation marks.

4.2. Evaluation

Participants to the ASR evaluation had to submit the results of the recognition of the *tst2017* sets in CTM format. The word error rate was measured case-insensitive. After the end of the evaluation, scoring was performed with the references derived from the subtitles of the TED talks and human transcripts of the real lectures.

For the SLT evaluation, participants could choose to either use their own ASR technology, or to use ASR output provided by the conference organizers.

For both input languages, the ASR output provided by the organizers was a single system output from one of the submissions to the ASR track.

Since the participants needed to segment the input into sentences, the segmentation of the reference and the automatic translation was different. In order to calculate the automatic evaluation metric, we needed to realign the sentences of the reference and the automatic translation. This was done by minimizing the WER between the automatic translation and reference as described in [16].

4.3. Submissions

We received two primary submissions for every SLT task and one primary submission for the ASR task.

4.4. Results

The detailed results of the automatic evaluation in terms of BLEU and WER can be found in Appendix B.

5. Human Evaluation

This year human evaluation focused on Multilingual translation (see Section 2) and was specifically carried out on the four language directions for which also the Zero-Shot translation task was proposed, *i.e.* $NlDe$, $DeNl$, $RoIt$ and $ItRo$.

For these four tasks, we received multilingual submissions for all the training data conditions offered, namely *large data* (ML LD), *small data* (ML SD), and *zero-shot* (ML ZS). Since multilingual translation was offered for the first time as an IWSLT task, we were interested in comparing the results with the traditional bilingual (BL) approach, where a different system is created for each language direction. For this reason, for the $NlDe$ and $RoIt$ tasks we asked those teams who participated with both ML SD and ML ZS runs to provide additional BL SD runs, to be manually evaluated as well.

A major novelty with respect to previous campaigns is that human evaluation was extended to include two different assessment methodologies, namely *direct assessment* (DA) of absolute translation quality as well as the traditional IWSLT evaluation based on *post-editing*

(PE), where the MT outupts are post-edited (*i.e.* manually corrected) by professional translators and then evaluated according to TER-based metrics [17].

We believe that carrying out a double evaluation on the same data adds great value to IWSLT 2017, since it allows to compare complementary methodologies which address different human perspectives. Indeed, while DA focuses on the generic assessment of overall translation quality, PE-based evaluation reflects a real application scenario – the integration of MT in Computer-Assisted Translation (CAT) tools – and directly measures the utility of a given MT output to translators. Also, this evaluation is particularly suitable for performing fine-grained analyses, since it produces a set of edits pointing to specific translation errors.

In this year's campaign, all systems submitted to the $NlDe$, $DeNl$, $RoIt$ and $ItRo$ tasks were officially evaluated and ranked according to DA, while PE-based evaluation was carried out on a subset of systems submitted to the $NlDe$ and $RoIt$ tasks, with the aim of analysing in detail the feasibility of the novel multilingual - and zero-shot - approach.

The human evaluation (HE) dataset created for each language direction was a subset of the corresponding 2017 test set *(tst2017)*. All the four *tst2017* sets ($NlDe$, $DeNl$, $RoIt$ and $ItRo$) are composed of the same 10 TED Talks, and around the first half of each talk was included in the HE set. The resulting HE sets are identical and include 603 segments, corresponding to around 10,000 words words for each source text.

In the following subsections we present the two evaluation methodologies and their outcomes on the HE datasets.

5.1. Direct Assessment

Recently, there has been increased interest in human evaluation of machine translation output using *direct assessment* (DA). Here, the annotator sees a simple annotation interface which shows 1) the reference translation, 2) a single candidate translation, and 3) a slider to score the translation quality from 1 to 100, focusing on the adequacy of the given translation output, compared to the given reference translation. For this year's IWSLT, we follow the setup of WMT17 [18] and run a human evaluation campaign based on DA.

Considering that any reference-based approach to evaluation will inevitably have problems when the reference translation has quality issues or a given candidate translation has an extremely different syntactic structure compared to the given reference (and might thus be judged as poor quality), we also focused on *source-based direct assessment*. This is more difficult to use as it requires a pool of bilingual annotators but (if those annotators are available) it allows to collect annotations on the actual semantic transfer between source and target languages.

Given that source-based DA eliminates reference bias and quality issues by design, we decided to run two separate DA campaigns for IWSLT, one based on the reference-based implementation of DA (identical to what has been used for WMT17) and one based on source-based DA. We used the Appraise framework [19] for both campaigns.

5.1.1. Data Preparation

Data was prepared based on the full set of 603 candidate translations used for the post-editing evaluation. However, as we wanted to ensure that each task is annotated by two annotators, we opted to randomly sample half of the candidate translations for the DA campaigns. Both source-based and reference-based direct assessment data has been prepared using the same random seed so that the only difference between the resulting tasks is in the type of "visual reference" shown to the annotator. Display order of segments and systems is identical across the campaign types.

5.1.2. Annotation Campaign

We collected annotations from a=22 annotators for $NlDe$ and $RoIt$. These language pairs contained a total of n=12 different systems and we conducted the evaluation on t=55 tasks with r=2 redundancy, so that each annotator ended up completing a total of five tasks. For $DeNl$ and $ItRo$ there were a total of a=16 annotators for

Table 6: $NlDe$ Source-based DA Human evaluation results showing average raw DA scores (Ave %) and average standardized scores (Ave z), lines between systems indicate clusters according to Wilcoxon rank-sum test at p-level $p \leq 0.05$.

#	Ave %	Ave z	**System**	**Cond.**
1	70.2	0.173	KIT	ML LD
2	70.2	0.145	KYOTO	BL SD
	69.4	0.139	KYOTO	ML SD
3	68.1	0.110	KIT	ML SD
4	68.4	0.103	KYOTO	ML ZS
	66.5	0.040	GTCT	ML ZS
	67.0	0.029	UDSDFKI	ML SD
5	64.5	-0.045	FBK	BL SD
	63.5	-0.078	UDSDFKI	ML ZS
	63.3	-0.079	FBK	ML SD
6	60.0	-0.212	FBK	ML ZS
7	57.2	-0.338	UDSDFKI	BL SD

Table 7: $NlDe$ Reference-based DA Human evaluation results showing average raw DA scores (Ave %) and average standardized scores (Ave z), lines between systems indicate clusters according to Wilcoxon rank-sum test at p-level $p \leq 0.05$.

#	Ave %	Ave z	**System**	**Cond.**
1	64.2	0.121	KIT	ML LD
2	63.5	0.100	KYOTO	ML SD
3	64.6	0.102	KYOTO	BL SD
4	63.0	0.069	KYOTO	ML ZS
	62.1	0.061	KIT	ML SD
	62.7	0.045	UDSDFKI	ML SD
	61.2	0.014	GTCT	ML ZS
5	61.1	0.017	FBK	BL SD
6	59.2	-0.076	UDSDFKI	ML ZS
	58.0	-0.092	FBK	ML SD
7	56.2	-0.178	FBK	ML ZS
	54.9	-0.241	UDSDFKI	BL SD

Table 8: $RoIt$ Source-based DA Human evaluation results showing average raw DA scores (Ave %) and average standardized scores (Ave z), lines between systems indicate clusters according to Wilcoxon rank-sum test at p-level $p \leq 0.05$.

#	Ave %	Ave z	**System**	**Cond.**
1	74.8	0.222	KYOTO	BL SD
2	74.4	0.200	KIT	ML SD
	72.1	0.131	KYOTO	ML SD
3	72.1	0.136	KYOTO	ML ZS
	71.8	0.115	KIT	ML LD
4	71.1	0.081	UDSDFKI	ML SD
	70.3	0.049	FBK	ML SD
	69.1	0.017	GTCT	ML ZS
	68.5	0.000	FBK	BL SD
5	66.9	-0.090	UDSDFKI	ML ZS
6	61.6	-0.268	FBK	ML ZS
7	55.3	-0.546	UDSDFKI	BL SD

n=9 individual systems. We annotated a set of t=40 tasks, again using r=2 redundancy, for the same annotator work load of five tasks. Our annotators were experienced linguistic consultants.

5.1.3. Results

Table 6 includes source-based DA results for $NlDe$ and Table 7 shows corresponding results from the reference-based DA campaign. Clusters are identified by grouping systems together according to which systems significantly outperform all others in lower ranking clusters, according to Wilcoxon rank-sum test. Tables 8 and 9 show results for source-based and reference-based DA for $RoIt$, respectively. Results for $DeNl$ and $ItRo$ are given in Tables 10, 11, 12, and 13.

Table 9: $RoIt$ Reference-based DA Human evaluation results showing average raw DA scores (Ave %) and average standardized scores (Ave z), lines between systems indicate clusters according to Wilcoxon rank-sum test at p-level $p \leq 0.05$.

#	Ave %	Ave z	**System**	**Cond.**
1	59.9	0.169	KIT	ML SD
2	59.9	0.162	KYOTO	ML SD
3	58.9	0.126	KYOTO	BL SD
	58.6	0.126	KYOTO	ML ZS
	58.3	0.102	KIT	ML LD
4	58.3	0.086	UDSDFKI	ML SD
5	55.2	0.014	GTCT	ML ZS
	55.1	-0.010	FBK	ML SD
	54.0	-0.045	FBK	BL SD
	54.0	-0.047	UDSDFKI	ML ZS
6	49.0	-0.190	FBK	ML ZS
7	42.9	-0.423	UDSDFKI	BL SD

Table 10: $DeNl$ Source-based DA Human evaluation results showing average raw DA scores (Ave %) and average standardized scores (Ave z), lines between systems indicate clusters according to Wilcoxon rank-sum test at p-level $p \leq 0.05$.

#	Ave %	Ave z	**System**	**Cond.**
1	70.3	0.128	KYOTO	ML ZS
2	70.0	0.088	KIT	ML LD
3	69.8	0.094	KYOTO	ML SD
	67.5	0.015	GTCT	ML ZS
	67.5	-0.002	KIT	ML SD
	67.4	-0.006	FBK	ML SD
4	66.5	-0.022	UDSDFKI	ML SD
	66.0	-0.073	UDSDFKI	ML ZS
5	62.4	-0.180	FBK	ML ZS

Note how reference-based DA scores are generally lower than those for source-based DA. It seems that given a reference, annotators are more likely to penalize a candidate translation for missing data. For the source-based case, they seem to be more focused on the actual transfer from source into target language. More detailed investigation is required to draw conclusions here and will be left for future work.

Generally, source-based and reference-based DA produce similar clusters. The decision which direct assessment to use hence comes down to the availability of bilingual annotators. If available, it seems preferable to opt for source-based DA.

For $NlDe$, KIT (ML LD) wins for both source-based and reference-based DA, with KYOTO (BL SD and ML SD) reaching second and third place. KIT is significantly better than all other systems for this language pair. Both DA methods agree on the ranking of the lower scor-

Table 11: $DeNl$ Reference-based DA Human evaluation results showing average raw DA scores (Ave %) and average standardized scores (Ave z), lines between systems indicate clusters according to Wilcoxon rank-sum test at p-level $p \leq 0.05$.

#	Ave %	Ave z	**System**	**Cond.**
1	57.7	0.126	KIT	ML LD
2	57.7	0.119	KYOTO	ML SD
	56.6	0.090	KYOTO	ML ZS
3	54.7	0.004	KIT	ML SD
4	54.4	0.009	GTCT	ML ZS
	53.7	-0.022	UDSDFKI	ML SD
	53.4	-0.068	UDSDFKI	ML ZS
	52.6	-0.073	FBK	ML SD
5	50.2	-0.156	FBK	ML ZS

Table 12: $ItRo$ Source-based DA Human evaluation results showing average raw DA scores (Ave %) and average standardized scores (Ave z), lines between systems indicate clusters according to Wilcoxon rank-sum test at p-level $p \leq 0.05$.

#	Ave %	Ave z	**System**	**Cond.**
1	77.3	0.214	KIT	ML LD
	76.5	0.189	KYOTO	ML SD
	75.9	0.173	KIT	ML SD
	74.7	0.136	KYOTO	ML ZS
2	72.6	0.048	UDSDFKI	ML SD
3	69.6	-0.070	FBK	ML SD
4	68.5	-0.103	UDSDFKI	ML ZS
	68.1	-0.115	GTCT	ML ZS
5	60.4	-0.385	FBK	ML ZS

ing systems.

For $RoIt$, KYOTO (BL SD) wins for source-based DA while KIT (ML SD) performs best for the reference-based DA campaign. For reference-based eval, the KYOTO systems drops to the third cluster. As average scores are really close across the reference-based systems, this should be investigated more. Again, both DA methods agree on the worst clusters.

For $DeNl$, we see the ML ZS system from KYOTO win over an ML LD system from KIT. While this does not happen for the reference-based campaign, the ML ZS system achieves second place there. This indicates that ML ZS can be competitive and outperforms the other approaches.

Finally, for $ItRo$ we observe identical clusters for both DA methods. Of course, average % scores and z scores differ, but the respective pairwise comparisons end up the same. Four systems achieve first rank: KIT (ML SD and ML LD) as well as KYOTO (ML SD and ML ZS).

Table 13: $ItRo$ Reference-based DA Human evaluation results showing average raw DA scores (Ave %) and average standardized scores (Ave z), lines between systems indicate clusters according to Wilcoxon rank-sum test at p-level $p \leq 0.05$.

#	Ave %	Ave z	**System**	**Cond.**
1	66.1	0.165	KIT	ML SD
	65.4	0.145	KYOTO	ML ZS
	65.1	0.142	KIT	ML LD
	64.2	0.112	KYOTO	ML SD
2	61.5	0.021	UDSDFKI	ML SD
3	60.0	-0.050	FBK	ML SD
4	58.1	-0.095	UDSDFKI	ML ZS
	58.3	-0.102	GTCT	ML ZS
5	54.0	-0.229	FBK	ML ZS

5.2. Post-Editing

5.2.1. Evaluation Data

This year, human evaluation based on post-editing was carried out on two language directions, namely $NlDe$ and $RoIt$.

In order to analyze at best the multilingual approach and to properly compare the different data conditions tested in the campaign, we selected for post-editing the six runs of the three teams who submitted both ML SD and ML ZS systems (*i.e.* KYOTO, FBK, UDSDFKI). In addition, we included in the evaluation their three unofficial BL SD runs that they were requested to submit for comparison purposes.

For each language direction, the output of the selected 9 systems on the HE set was assigned to professional translators to be post-edited (for all the details about data preparation and post-editing see [20, 21, 22]).

The resulting evaluation data consists of nine new reference translations for each of the sentences in the HE set. Each one of these references represents the *targeted translation* of the system output from which it was derived, while the post-edits of the other 8 systems are available for evaluation as additional references.

5.2.2. Results

The outcomes for the two language directions are presented in Tables 14 and 15, where systems are grouped by data condition (ML ZS, ML SD, ML LD, and BL SD). Results are analyzed according to multi-reference TER (mTER), where TER is computed against all the 9 available post-edits. Previous IWSLT PE-based evaluations demonstrated that mTER allows a more reliable and consistent evaluation of the real overall MT system performance with respect to HTER – where

TER is calculated against the targeted reference only.

Furthermore, figures are given for HTER as well as TER – both on the HE set and on the full test set – calculated against the official reference translation used for automatic evaluation (see Section 2 and Appendix A).[4] In the tables, BL SD runs are highlighted in light gray to distinguish them from the official IWSLT runs. Also, results for those official IWSLT runs that were not post-edited are given for completeness (*i.e.* KIT, GTCT). Those runs are highlighted in dark gray to signal that they are not directly comparable with the other runs: although they are evaluated with mTER on all nine available references, they do not have their corresponding targeted reference, which could result in a penalizing score.

Finally, the statistical significance of the observed differences between the systems was assessed with the *approximate randomization* method [23], a statistical test well-established in the NLP community [24] and that, especially for the purpose of MT evaluation, has been shown [25] to be less prone to type-I errors than the bootstrap method [26]. In this study, the approximate randomization test was based on 10,000 iterations. Tables 14 and 15 present the results of the test focusing on the systems within the same data condition. Information about the significance of the differences between the systems developed by the same team are given in the following discussion of results.

Table 14: $NlDe$ TED Talk task (HE *tst2017*): human evaluation results. Scores are given in percentage (%). The number next to the mTER score identifies the system(s) within the same setup w.r.t. which the difference is statistically significant at $p < 0.01$.

Cond.	System	mTER HE Set 9 PErefs	HTER HE Set tgt PEref	TER HE Set ref	TER Test Set ref
ML ZS	GTCT	25.36	–	64.40	65.17
	KYOTO[1]	20.33$^{(2,3)}$	25.72	64.33	64.33
	FBK[2]	26.19	33.13	67.01	67.05
	UDSDFKI[3]	27.36	33.60	68.65	68.36
ML SD	KYOTO	20.38$^{(3)}$	25.05	62.99	63.39
	FBK	21.68	27.68	65.48	65.25
	UDSDFKI	23.94	30.75	66.76	66.34
	KIT	21.34	–	62.12	62.56
ML LD	KIT	19.03	–	61.08	61.33
BL SD	KYOTO	20.31$^{(2,3)}$	26.26	63.61	63.81
	FBK	23.71$^{(3)}$	30.18	65.34	66.09
	UDSDFKI	30.27	37.25	70.72	70.30

[4]Note that since TER is an edit-distance measure, lower numbers indicate better performance.

Table 15: $RoIt$ TED Talk task (HE *tst2017*): human evaluation results. Scores are given in percentage (%). The number next to the mTER score identifies the system(s) within the same setup w.r.t. which the difference is statistically significant at $p < 0.01$.

Cond.	System	mTER HE Set 9 PErefs	HTER HE Set tgt PEref	TER HE Set ref	TER Test Set ref
ML ZS	GTCT	26.94	–	61.80	61.11
	KYOTO[1]	22.65$^{(2,3)}$	29.33	60.58	60.26
	FBK[2]	29.16	37.38	64.21	63.32
	UDSDFKI[3]	28.74	35.79	64.79	63.97
ML SD	KYOTO	20.27	27.17	60.14	59.75
	FBK	20.74	29.01	60.45	59.65
	UDSDFKI	23.39	31.25	61.95	60.77
	KIT	22.81	–	58.70	58.29
ML LD	KIT	22.48	–	58.46	57.87
BL SD	KYOTO	18.39$^{(2,3)}$	26.09	58.90	58.55
	FBK	22.69$^{(3)}$	30.34	61.25	60.73
	UDSDFKI	26.73	34.85	61.74	63.40

Looking at the tables, some conclusions can be drawn about the feasibility of multilingual MT. It is interesting to note that the same considerations hold across language directions – although to varying degrees. First of all, the impressive results of ML SD runs show that multilingual systems are indeed an effective alternative to traditional bilingual systems. Even more noticeably, ML ZS systems are able to reach a reasonably good quality also when faced with such an extreme translation scenario, clearly showing the feasibility of the zero-shot approach. Finally, by comparing the systems' performance within each condition, some specific characteristics of the ML and BL approach emerge. As we can see in the tables, the three BL SD systems are all significantly different, while ML SD systems (and ML ZS, although to a lesser extent) are mostly similar to each other.

We now compare in detail the systems produced by each team in the different conditions. Considering the $NlDe$ direction (Table 14), KYOTO provides the clearest demonstration of the feasibility of the multilingual zero-shot approach, since it obtains the same outstanding results in all the three translation conditions. FBK and UDSDFKI systems show a very similar behaviour. They further confirm the effectiveness of the multilingual approach, since their ML SD runs improve over their corresponding BL SD runs, and with a statistically significant difference. As for zero-shot translation, FBK and UDSDFKI systems still show a reasonably good quality, although results are significantly lower

than those obtained in the ML SD data condition (+4.51 mTER points for FBK and +3.42 for UDS-DFKI). With respect to the BL SD runs, UDS-DFKI ML ZS performance is higher (though the difference is not statistically significative), while FBK ML ZS results are significantly lower.

Regarding non-comparable runs (in dark grey in the table), we see that the ML ZS system developed by GTCT is in line with the other results. As for KIT, its performance on the ML LD data condition confirms that using more data for training can help improving results. However, the difference with respect to its corresponding ML SD system is not particularly remarkable, although statistically significant.

It is worthwhile to note that the differences between systems highlighted by mTER scores are not so marked when looking at TER scores. As also shown in previous IWSLT evaluations, TER calculated against one independent reference does not allow to discriminate properly between systems; this study supports once more the need for human evaluation to shed light on the peculiarities of the systems.

Considering the $RoIt$ language direction (Table 15), we can draw the same conclusions about the feasibility of the multilingual approach, although results for the zero-shot task are less notable. KYOTO ML SD system is not significantly different from the traditional BL SD system, even though it does not reach its performance. On the contrary, results for ML ZS system are significantly lower than those obtained by the ML SD one, although the difference is only 2.38 mTER points.

As seen for the $NlDe$ direction, FBK and UDSDFKI ML SD runs significantly improve over their corresponding BL SD runs; however, for the $RoIt$ direction the drop in performance of the ML ZS systems with respect to the ML SD ones is more critical (8.42 mTER points for FBK and 5.35 for UDSDFKI). Also, ML ZS runs are worse than BL SD runs, even though for UDS-DFKI the difference is not statistically significant.

5.3. Future Work

We intend to run a deeper analysis on the human evaluation corpus created as part of IWSLT. Not only does it make sense to more closely investigate the differences of source-based and reference-based DA, but it will also be very interesting to compare the results of such "general quality focused" annotation work to more targeted approaches such as post-editing. As we do have such data for two of the language pairs, the resulting three-way dataset will be released for future research.

6. Conclusions

This year the IWSLT Evaluation Campaign featured three tasks: the Multilingual task, evaluating single MT systems translating across multiple languages, the Dialogues task, addressing MT of human-to-human dialogues, and the Lecture task, targeting speech transcription and translation of real-life university lectures. This paper overviews the structure of each task, its experimental conditions, the training and evaluation data made available, and reports on its participation and main outcomes. Besides documenting the evaluation campaign to the perusal of the workshop participants, we hope that this paper will also be useful to researchers and practitioners interested in using our evaluation benchmarks in the future.

7. Acknowledgements

Human evaluation based on post-editing and part of the work by FBK's authors were supported by the CRACKER project, which receives funding from the EU's Horizon 2020 research and innovation programme under grant agreement no. 645357.

8. References

[1] N. Ruiz and M. Federico, "Complexity of spoken versus written language for machine translation," in *Proc. of EAMT*, Dubrovnik, Croatia, 2014, pp. 173–180.

[2] S. M. Lakew, Q. F. Lotito, M. Turchi, M. Negri, and M. Federico, "FBK's multilingual neural machine translation system for IWSLT 2017," in *Proc. of IWSLT*, Tokyo, Japan, 2017.

[3] C. Bei and H. Zong, "Towards better translation performance on spoken language," in *Proc. of IWSLT*, Tokyo, Japan, 2017.

[4] N.-Q. Pham, M. Sperber, E. Salesky, T.-L. Ha, J. Niehues, and A. Waibel, "KIT's multilingual neural machine translation systems for IWSLT 2017," in *Proc. of IWSLT*, Tokyo, Japan, 2017.

[5] R. Dabre, F. Cromieres, and S. Kurohashi, "Kyoto university MT system description for IWSLT 2017," in *Proc. of IWSLT*, Tokyo, Japan, 2017.

[6] P. Bahar, J. Rosendahl, N. Rossenbach, and H. Ney, "The RWTH Aachen machine

translation systems for IWSLT 2017," in *Proc. of IWSLT*, Tokyo, Japan, 2017.

[7] P. Przybysz, M. Chochowski, R. Sennrich, B. Haddow, and A. Birch, "The Samsung and University of Edinburgh's submission to IWSLT17," in *Proc. of IWSLT*, Tokyo, Japan, 2017.

[8] C. España-Bonet and J. van Genabith, "Going beyond zero-shot MT: combining phonological, morphological and semantic factors. The UdS-DFKI system at IWSLT 2017," in *Proc. of IWSLT*, Tokyo, Japan, 2017.

[9] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web inventory of transcribed and translated talks," in *Proc. of EAMT*, Trento, Italy, 2012.

[10] S. Takeno, M. Nagata, and K. Yamamoto, "Integrating empty category detection into preordering machine translation," in *Proc. WAT*, Osaka, Japan, 2016.

[11] T. Chung and D. Gildea, "Effects of empty categories on machine translation," in *Proc. of EMNLP*, Cambridge, US-MA, 2010.

[12] M. Paul, "Overview of the IWSLT 2009 evaluation campaign," in *Proc. of IWSLT*, Tokyo, Japan, 2009.

[13] M. Yang, H. Jiang, T. Zhao, and S. Li, "Construct trilingual parallel corpus on demand," in *Chinese Spoken Language Processing. Lecture Notes in Computer Science*, Q. Huo, B. Ma, E.-S. Chng, and H. Li, Eds. Springer, Berlin, Heidelberg, 2006, vol. 4274, pp. 760–767.

[14] M. Federico, M. Cettolo, L. Bentivogli, M. Paul, and S. Stüker, "Overview of the IWSLT 2012 evaluation campaign," in *Proc. of IWSLT*, Hong Kong, 2012.

[15] H. Tanaka, K. Yoshino, K. Sugiyama, S. Nakamura, and M. Kondo, "Multimodal interaction data between clinical psychologists and students for attentive listening modeling," in *Proc. of O-COCOSDA*, Bali, Indonesia, 2016.

[16] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating machine translation output with automatic sentence segmentation," in *Proc. of IWSLT*, Pittsburgh, US-PA, 2005.

[17] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proc. of AMTA*, Cambridge, US-MA, 2006.

[18] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, C. Monz, M. Negri, M. Post, R. Rubino, L. Specia, and M. Turchi, "Findings of the 2017 conference on machine translation (WMT17)," in *Proc. of WMT: Shared Task Papers*, Copenhagen, Denmark, 2017.

[19] C. Federmann, "Appraise: An open-source toolkit for manual evaluation of machine translation output," *The Prague Bulletin of Mathematical Linguistics*, vol. 98, pp. 25–35, September 2012.

[20] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 11th IWSLT evaluation campaign, IWSLT 2014," in *Proc. of IWSLT*, Lake Tahoe, US-CA, 2014.

[21] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, R. Cattoni, and M. Federico, "The IWSLT 2015 evaluation campaign," in *Proc. of IWSLT*, Da Nang, Vietnam, 2015.

[22] ——, "The IWSLT 2016 evaluation campaign," in *Proc. of IWSLT*, Seattle, US-WA, 2016.

[23] E. W. Noreen, *Computer Intensive Methods for Testing Hypotheses: An Introduction*. Wiley Interscience, 1989.

[24] N. Chinchor, L. Hirschman, and D. D. Lewis, "Evaluating message understanding systems: An analysis of the third message understanding conference (muc-3)," *Computational Linguistics*, vol. 19, no. 3, pp. 409–449, 1993.

[25] S. Riezler and J. T. Maxwell, "On some pitfalls in automatic evaluation and significance testing for MT," in *Proc. of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, US-MI, 2005.

[26] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Chapman and Hall, 1993.

Appendix A. Automatic Evaluation for the Multilingual Task

· Table scores refer to the official testset (*tst2017.mltlng*)
· *BLEU* and *TER* scores are given as percent figures (%)
· ML, BL, SD, LD and ZS stand for multilingual, bilingual, small-data, large-data and zero-shot conditions, respectively
· BL SD systems were developed by three participants on explicit request of the organizers for comparison purposes

system	cond.	BLEU	NIST	TER	BLEU	NIST	TER
		Dutch-German			**German-Dutch**		
FBK	ML SD	18.59	5.177	65.24	19.16	5.583	61.45
	ML ZS	16.96	4.931	67.04	17.17	5.297	63.25
	BL SD	17.93	5.139	66.09	–	–	–
GTCT	ML ZS	19.00	5.208	65.17	19.59	5.565	61.27
KIT	ML SD	20.47	5.542	62.56	19.77	5.735	59.37
	ML LD	21.06	5.657	61.33	20.00	5.763	59.21
KYOTO	ML SD	20.27	5.487	63.39	19.64	5.733	60.24
	ML ZS	19.68	5.368	64.33	20.31	5.751	59.99
	BL SD	19.50	5.390	63.81	19.86	5.754	59.93
UDSDFKI	ML SD	18.28	5.133	66.34	18.96	5.492	63.50
	ML ZS	16.28	4.874	68.36	17.38	5.375	62.72
	BL SD	16.43	4.767	70.30	–	–	–
		Dutch-Italian			**Italian-Dutch**		
FBK	ML SD	19.33	5.471	62.88	20.27	5.568	61.78
	ML ZS	19.76	5.422	62.99	20.00	5.548	61.91
GTCT	ML ZS	21.21	5.722	60.84	21.80	5.784	60.09
KIT	ML SD	20.41	5.599	61.64	22.14	6.005	58.34
	ML LD	20.94	5.706	60.18	21.95	6.003	58.21
KYOTO	ML SD	19.86	5.530	62.07	22.32	5.922	59.16
	ML ZS	20.74	5.602	61.85	22.76	5.911	59.16
UDSDFKI	ML SD	19.12	5.419	63.69	20.08	5.560	62.02
	ML ZS	19.39	5.435	63.68	19.88	5.563	61.92
		English-Dutch			**Dutch-English**		
FBK	ML SD	26.72	6.536	53.45	29.79	7.078	50.27
	ML ZS	26.11	6.501	54.34	30.04	7.081	50.04
GTCT	ML ZS	29.08	6.805	51.47	32.78	7.422	47.35
KIT	ML SD	29.15	6.903	51.08	31.79	7.340	47.84
	ML LD	30.22	6.984	50.45	31.95	7.399	46.88
KYOTO	ML SD	28.80	6.824	52.16	30.49	7.131	49.04
	ML ZS	30.18	6.963	50.71	30.63	7.158	48.94
UDSDFKI	ML SD	26.49	6.529	53.72	29.53	7.112	49.64
	ML ZS	26.37	6.534	54.19	29.69	7.073	50.03
		English-Italian			**Italian-English**		
FBK	ML SD	29.60	6.821	50.74	34.24	7.618	44.45
	ML ZS	28.86	6.687	51.80	34.16	7.638	44.38
GTCT	ML ZS	32.84	7.222	47.63	37.84	8.100	41.06
KIT	ML SD	32.04	7.147	48.36	36.30	7.945	41.97
	ML LD	32.32	7.219	48.11	36.46	7.980	41.89
KYOTO	ML SD	30.79	6.921	50.48	34.73	7.631	45.07
	ML ZS	30.99	6.989	49.69	35.28	7.679	44.51
UDSDFKI	ML SD	29.62	6.855	50.48	33.77	7.644	44.07
	ML ZS	29.68	6.849	50.55	33.77	7.596	44.71
		German-Italian			**Italian-German**		
FBK	ML SD	16.84	5.094	65.67	16.88	4.92	68.38
	ML ZS	16.28	4.971	66.76	16.13	4.828	69.22
GTCT	ML ZS	18.56	5.363	63.44	18.09	5.091	67.28
KIT	ML SD	17.79	5.265	63.81	19.32	5.344	64.71
	ML LD	18.04	5.280	63.01	19.85	5.414	64.16
KYOTO	ML SD	17.54	5.262	64.32	19.10	5.339	64.73
	ML ZS	17.67	5.227	64.77	19.20	5.287	65.31
UDSDFKI	ML SD	16.66	5.096	66.12	16.48	4.870	69.15
	ML ZS	16.73	5.106	66.09	16.27	4.873	68.79

system	cond.	BLEU	NIST	TER	BLEU	NIST	TER
		Italian-Romanian			**Romanian-Italian**		
FBK	ML SD	19.06	5.155	64.87	21.34	5.811	59.65
	ML ZS	16.58	4.783	67.53	18.32	5.296	63.32
	BL SD	–	–	–	21.71	5.776	60.73
GTCT	ML ZS	18.62	5.027	65.54	20.39	5.573	61.11
KIT	ML SD	21.08	5.566	61.31	22.54	6.0209	58.28
	ML LD	21.09	5.629	60.68	22.35	6.013	57.87
KYOTO	ML SD	20.60	5.446	62.76	21.89	5.820	59.75
	ML ZS	20.37	5.385	62.79	21.85	5.789	60.26
	BL SD	–	–	–	23.14	6.026	58.55
UDSDFKI	ML SD	17.77	5.001	66.40	21.22	5.743	60.77
	ML ZS	16.07	4.752	68.21	18.67	5.352	63.97
	BL SD	–	–	–	18.94	5.345	63.40
		Dutch-Romanian			**Romanian-Dutch**		
FBK	ML SD	16.54	4.759	68.32	18.92	5.396	63.48
	ML ZS	15.88	4.698	68.57	17.72	5.272	64.51
GTCT	ML ZS	18.11	4.966	66.55	20.02	5.586	61.87
KIT	ML SD	17.43	5.067	64.98	19.28	5.674	60.93
	ML LD	17.52	5.103	64.48	19.19	5.645	61.10
KYOTO	ML SD	17.65	5.055	65.84	20.24	5.745	60.90
	ML ZS	17.74	5.056	65.75	20.47	5.699	61.14
UDSDFKI	ML SD	14.83	4.529	71.33	17.58	5.281	65.16
	ML ZS	14.93	4.532	71.79	17.26	5.286	64.44
		English-German			**German-English**		
FBK	ML SD	20.88	5.501	63.50	25.62	6.528	54.05
	ML ZS	20.67	5.471	63.80	25.22	6.453	54.54
GTCT	ML ZS	23.08	5.861	60.63	28.04	6.851	51.42
KIT	ML SD	23.86	6.029	59.22	26.76	6.694	52.43
	ML LD	25.49	6.212	57.75	27.47	6.803	51.26
KYOTO	ML SD	23.25	5.924	60.23	26.45	6.609	52.65
	ML ZS	23.63	5.936	60.22	27.08	6.678	52.49
UDSDFKI	ML SD	20.63	5.535	63.37	24.75	6.445	54.74
	ML ZS	20.20	5.504	63.49	24.54	6.442	55.22
		English-Romanian			**Romanian-English**		
FBK	ML SD	21.95	5.600	61.40	28.93	6.964	49.91
	ML ZS	21.54	5.575	61.41	28.52	6.925	50.57
GTCT	ML ZS	23.89	5.906	58.81	31.79	7.368	47.22
KIT	ML SD	25.09	6.132	56.92	30.71	7.208	48.18
	ML LD	25.25	6.133	56.95	30.69	7.242	48.01
KYOTO	ML SD	24.66	6.059	57.70	29.58	7.063	49.10
	ML ZS	24.49	6.073	57.16	30.23	7.102	48.78
UDSDFKI	ML SD	20.35	5.425	63.30	27.99	6.877	51.44
	ML ZS	20.25	5.353	63.99	28.25	6.902	51.09
		German-Romanian			**Romanian-German**		
FBK	ML SD	14.62	4.479	70.96	15.87	4.762	69.04
	ML ZS	13.93	4.400	71.10	15.47	4.695	69.87
GTCT	ML ZS	16.23	4.689	69.04	17.95	5.057	67.03
KIT	ML SD	14.99	4.690	67.59	18.01	5.181	66.01
	ML LD	15.31	4.737	67.12	18.14	5.198	65.44
KYOTO	ML SD	16.27	4.794	68.08	17.94	5.135	66.44
	ML ZS	16.08	4.822	67.76	18.40	5.152	66.24
UDSDFKI	ML SD	13.89	4.381	72.13	15.30	4.667	71.66
	ML ZS	13.83	4.287	72.97	15.01	4.652	71.37

Appendix B. Automatic Evaluation for the Lecture Task

ASR: Talk English and German
Results in Word Error Rate (WER)

German		English	
Testset	KIT	**Testset**	KIT
lecture 01	16.6	lecture 01	9.9
lecture 03	33.8	lecture 02	11.7
lecture 04	22.7	ted 2403	6.6
		ted 2429	10.6
		ted 2438	6.6
		ted 2439	15.5
		ted 2440	4.1
		ted 2442	6.7
		ted 2447	6.0
		ted 2507	6.2
All lectures	22.8	All lectures	10.3
All ted	–	All ted	7.7
All	22.8	All	8.5

SLT: Lecture translation task
Results in BLEU

German - English			English - German		
Testset	KIT	UEDIN	**Testset**	KIT	UEDIN
lecture 01	17.31	18.86	ted 2403	18.67	16.48
lecture 03	7.66	8.39	ted 2413	17.06	13.91
lecture 04	15.32	17.58	ted 2429	23.87	16.17
			ted 2438	17.14	8.05
			ted 2439	14.95	8.71
			ted 2440	13.52	13.28
			ted 2442	20.89	16.30
			ted 2447	11.59	7.73
			ted 2478	17.67	12.69
			ted 2507	16.64	14.15
			lecture 01	23.40	23.56
			lecture 02	18.75	22.70
All	12.50	13.99	ALL	18.59	15.98

Going beyond zero-shot MT: combining phonological, morphological and semantic factors. The UdS-DFKI System at IWSLT 2017

Cristina España-Bonet and Josef van Genabith

University of Saarland and DFKI, Saarbrücken, Germany

{cristinae,Josef.Van_Genabith}@dfki.de

Abstract

This paper describes the UdS-DFKI participation to the multilingual task of the IWSLT Evaluation 2017. Our approach is based on factored multilingual neural translation systems following the small data and zero-shot training conditions. Our systems are designed to fully exploit multilinguality by including factors that increase the number of common elements among languages such as phonetic coarse encodings and synsets, besides shallow part-of-speech tags, stems and lemmas. Document level information is also considered by including the topic of every document. This approach improves a baseline without any additional factor for all the language pairs and even allows beyond-zero-shot translation. That is, the translation from unseen languages is possible thanks to the common elements —especially synsets in our models— among languages.

1. Introduction

Neural machine translation systems (NMT) are currently the state of the art for most language pairs [1] and, among other advantages with respect to other paradigms, they can be easily extended to multilingual systems (ML-NMT) [2, 3]. ML-NMT systems usually use a common vocabulary where some words are shared and, more importantly, they project all the languages into the same embedding space clustering sentences according to their meanings. However, the clustering is not perfect and especially distant languages or those with fewer data are more difficult to group by semantics [4].

With the aim of facilitating the semantic clustering of languages, we enrich words with several levels of annotation. The highest level of annotation is represented by *Babel synsets*. BabelNet (BN) is a multilingual semantic network connecting concepts via synsets [5]. Each concept, or word, is identified by its ID irrespective of its language, effectively turning these IDs interlingua. At a lower level, we start from the premise that languages, especially within families, share roots that have evolved with time. We use stems and lemmas to capture common roots and phonetic coarse encodings for phonetic similarities.

On the other hand, we also take advantage of the coherent structure of the training data composed by a collection of TED talk transcriptions in several languages. We inform the system about the topic of every word according to the document it belongs to, expecting to improve lexical selection in this way. Previous research modifies a standard encoder-decoder architecture to deal with extra-sentence information [6, 7]. Here we take the opposite approach and modify (annotate) the data in order to capture relevant knowledge.

Technically, we include all the aforementioned information as factors in a ML-NMT system. Following [8], each feature has its own word vector which is concatenated to the BPE'd token vectors to build the hidden states. So, when including factors in the source representation of a token, every source token embedding has the top-k elements describing the distribution of the word and the remaining elements describing other features.

This is not the first time that linguistic factors are used in NMT. We use the implementation in [8], but also the authors in [9, 10] use part of speech tags and grammatical information in their systems. However, to our knowledge, this is the first time that factors are designed to try to reduce distances between different source languages and therefore to mimic the effect of having a larger corpus. Also, BPE subunits are expected to be more descriptive since semantic information of the complete word is added to its representation. A vocabulary expansion is naturally produced by the concatenation of the different features. Our approach specifically targets multi- and interliguality in translation and, as an extreme effect, we show how *beyond-zero-shot translation* can be possible, that is, the translation from unseen source languages thanks to interlingual factors.

The rest of the paper is organised as follows. Section 2 describes all the factors used in this work and which tools and methodologies we use to obtain them. In Section 3 we analyse the characteristics of the training corpus with respect to these factors. Section 4 briefly describes the parameters of the ML-NMT systems and Section 5 reports the results in the small data and zero-shot training conditions. Finally, we summarise and draw our conclusions in Section 6.

2. Linguistic and Semantic Annotations

Coarse-Grained Part of Speech (p). We use a coarse-grained part-of-speech (PoS) tag set with 10 elements: {NOUN, VERB, PREPOSITION, PRONOUN, DETERMINER, ADVERB, ADJECTIVE,

Table 1: Statistics for the number of elements in the monolingual TED corpora (all, top block; $unique$, bottom block). Monolingual corpora have been built as the concatenation of all the parallel counterparts eliminating duplicates.

	West Germanic Languages			Latin Languages			
	en	de	nl	ro	it	es	fr
Sentences	545,270	303,668	444,287	225,980	513,693	151,631	140,717
Tokens	9,768,374	5,148,199	6,894,438	3,732,679	8,367,940	2,494,336	2,473,040
uToken	141,013	221,459	187,148	213,670	200,697	148,366	131,015
uLemma	73,048	101,003	85,846	72,535	52,525	52,052	53,088
uStem	50,128	94,126	85,560	54,227	44,691	35,307	40,504
uM3	57,630	79,029	60,534	30,576	32,828	31,840	32,234
uBN	28,445	34,022	27,720	24,375	27,172	23,567	23,856

Table 2: Coverage of the different subcorpora (in $\%$) by the common elements among the languages. The number in parenthesis shows the absolute number of common elements.

	Germanic	Latin	ALL	SMALL	ZERO
Token	40% (17,185)	32% (11,690)	30% (8,279)	30% (13,150)	32% (13,150)
Lemma	56% (14,576)	37% (9,096)	40% (7,922)	41% (11,835)	43% (11,835)
Stem	57% (12,029)	52% (8,114)	46% (4,971)	45% (7,452)	47% (7,452)
M3	87% (9,961)	87% (8,164)	84% (5,922)	69% (7,506)	70% (7,506)
BN	15% (5,507)	27% (6,104)	12% (2,367)	12% (3,291)	12% (3,291)

CONJUNCTION, ARTICLE, INTERJECTION}. This tag set is defined so as to be compatible with the one in the BabelNet ontology, so this set does not exactly correspond to the Universal Part-of-Speech Tagset [11] although the granularity is similar. We use the IXA pipeline [12] to annotate English, German, Spanish and French documents with PoS and TreeTagger [13] for Dutch, Romanian and Italian. The original tags are then mapped to our common reduced tagset[1].

Lemma (l). As with PoS, we use the IXA pipeline for English, German, Spanish and French; and TreeTagger for Dutch, Romanian and Italian.

Stem (s). Stems are obtained with the Snowball API which implements the Porter algorithm [14].

Approximate Phonetic Encoding (m). We use a phonetic algorithm to encode words by their pronunciation. The purpose is to bring close languages together by taking advantage of similar pronunciations in a similar way lemmas and stems do for close spellings. Phonetic algorithms that provide a coarse encoding of a word are more appropriate for this task than the real phonetic transcription which would be too discriminative.

Phonetic algorithms like Soundex [15] or Metaphone [16] are usually developed for a particular language with possible adjustments to deal with specific features of another one such as matching names. As an approximation, in our experiments we use Metaphone 3 for English on all the languages. Metaphone 3 (M3) is a phonetic algorithm that takes into account irregularities in English coming from several languages including Germanic and Latin ones. As generic features, the encoding converts all the initial vowels into an A and pairs of unvoiced and voiced consonants are encoded by the same letter. The algorithm is commercially available also for Spanish and German, but the only open source resource that we know of is for the English version[2].

Babel Synset (b). BabelNet [5] is a multilingual semantic network connecting concepts via *Babel synsets*. We enrich TED data content words with their synset information. For this, we select (*i*) nouns (including named entities, foreign words and numerals), (*ii*) adjectives, (*iii*) adverbs and (*iv*) verbs following the mappings to our coarse-grained part-of-speech tags. In addition, we explicitly mark negation particles with a tag NEG and include them here to account for their semantics.

A word can have several Babel synsets. We retrieve a synset according to the lemma and PoS of a word. In case there is still ambiguity, as it is in most of the cases, we select the BabelNet ID as the first ID according to its own sorting: *(a) puts WordNet synsets first; (b) sorts WordNet synsets based on the sense number of a specific input word; (c) sorts Wikipedia synsets lexicographically based on their main sense*.

Topic (t). TED talks are tagged with a set of English keywords that describe the topic of a document. Topic information can be relevant under two points of view: (*i*) given

[1]The mappings and the full annotation pipeline can be obtained here: https://github.com/cristinae/BabelWE

[2]Metaphone 3 is available within the OpenRefine tool, https://github.com/OpenRefine/OpenRefine

Table 3: Characterisation of the 20 topics learned with a BTM system. The percentage and absolute value of documents in the training corpus of every topic is shown together with the top-5 keywords that describe them.

Label	Proportion	Top-5 keywords
t1	10.6% (206)	science (6.5%) biology (5.6%) health (3.9%) medical research (3.9%) medicine (3.8%)
t2	10.0% (193)	culture (6.6%) entertainment (5.9%) technology (5.8%) design (5.3%) business (4.1%)
t3	8.2% (160)	culture (5.2%) entertainment (4.8%) art (4.0%) storytelling (3.1%) humor (3.1%)
t4	7.3% (141)	brain (7.3%) science (5.8%) neuroscience (5.3%) psychology (5.1%) mind (5.0%)
t5	6.6% (128)	global issues (5.0%) future (3.9%) society (3.9%) government (3.7%) politics (3.6%)
t6	6.6% (127)	environment (6.0%) science (5.0%) ecology (4.2%) plants (4.1%) nature (3.9%)
t7	6.0% (116)	technology (9.2%) computers (5.5%) design (4.7%) Internet (3.5%) TEDx (3.1%)
t8	5.9% (113)	technology (6.0%) environment (5.5%) science (4.7%) sustainability (4.6%) global issues (4.6%)
t9	5.3% (102)	science (7.1%) animals (5.4%) environment (5.0%) oceans (4.6%) biodiversity (4.5%)
t10	4.7% (90)	global issues (10.9%) politics (6.8%) war (5.8%) culture (4.8%) TEDx (4.0%)
t11	4.2% (81)	design (9.5%) technology (8.6%) invention (7.5%) innovation (5.8%) creativity (4.4%)
t12	4.0% (78)	global issues (9.2%) business (9.0%) economics (6.9%) culture (5.6%) Africa (4.5%)
t13	4.0% (77)	science (9.8%) technology (6.5%) space (6.1%) universe (5.7%) astronomy (5.4%)
t14	3.9% (77)	health (10.6%) healthcare (8.9%) medicine (8.2%) science (6.5%) technology (4.8%)
t15	3.7% (72)	technology (7.0%) science (6.4%) biology (4.2%) design (3.7%) robots (3.7%)
t16	2.7% (53)	women (7.3%) social change (5.7%) culture (5.1%) education (5.0%) activism (5.0%)
t17	2.1% (40)	design (13.5%) cities (10.1%) architecture (8.1%) art (4.9%) infrastructure (4.2%)
t18	1.8% (35)	music (14.7%) performance (13.8%) entertainment (12.9%) live music (10.2%) piano (3.6%)
t19	1.2% (24)	work (7.5%) business (5.6%) motivation (5.4%) personal growth (5.3%) success (4.4%)
t20	1.2% (23)	culture (12.9%) religion (9.5%) global issues (8.0%) philosophy (5.6%) science (5.1%)

a document, it is shared across languages, so it can help the NMT system to locate together in the embedding space the same sentence across languages, and (*ii*) it may improve document-level translation since it can help to disambiguate word translations according to its topic.

With a total of 390 different keywords and a mean of 6.5 per document, considering all of them as input information for the NMT system would lead to too much diversity. Besides, some keywords such as *technology*, *science*, *culture* or *global issues* are very frequent and could put in irrelevant information. Therefore, we decided to learn a topic model on the keywords and tag each document with a single interlingua label. Since a document is then only the short set of keywords in English, we apply a monolingual biterm topic model (BTM) for short texts [17] for the purpose.

As an alternative, we also apply polylingual topic models learned with Mallet [18] on all documents using the full vocabulary. However, after inferring the topic of each document, we obtained a mixture of top-k topics that did not allow a unique labelling of the same document across languages and the use of a single label would not be an interlingua tag as desired. Since keywords are always available for TED talks we used the first approach.

3. Corpus Characteristics

We use the corpus provided for the IWSLT 2017 multilingual task [19]. It comprises transcripts and manual translations of the TED talks accessible on April 26th, 2017. Two sets, *dev2010* and *tst2010*, are available for validation and testing purposes. The corpus includes documents in five languages, en-de-ro-it-nl, summing up to 9161 talks. The intersection of talks among languages is high, 7945 documents are common to all of them. In addition, we also use TED talks in French and Spanish obtained from previous IWSLT campaigns[3]. This data is not used for training, but we include them in the analysis of the corpus because in a subsequent section we explore the translation from unseen languages.

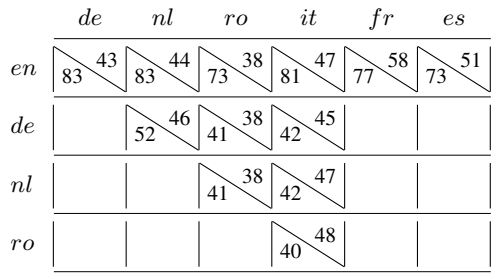
Table 1 shows the general statistics of the TED corpus by language. Languages are divided into two families: West Germanic with en, de and nl, and Latin with ro, it, es and fr. Notice that en, ro and it have significantly more sentences and that could benefit the translation from/to these languages, but the number of unique tokens ($uToken$) is quite homogeneous with the exception of fr and es.

The number of unique elements in the corpus decreases when going from words, to lemmas, stems, M3 encodings and BN synsets. The only exception is en, where we obtain more unique M3 encodings than stems. The number of unique elements is an indication of the ambiguity given by the factor: words are the least ambiguous linguistic factor but too many to be fully covered by the vocabulary of ML-NMT systems, and M3 encodings are the most ambiguous elements up to the point that they frequently erase the differences between unrelated words. In English, `anyone` and `union` share the same M3 encoding `ANN` but not the meaning. The same encoding applies to the German words `eine` and `ihnen` or the Italian ones `unione` or `annoiano`, some of them are translations, some of them not. BN synsets are not directly comparable because they are only obtained for a subset of PoS tags.

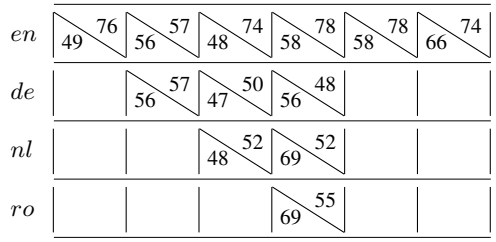
Our main interest is to observe the intersection of these elements in different languages. Table 2 reports the percentage of a corpus that is covered by the common elements among all the languages that build up such corpus. We show these figures for five corpora: Germanic including en, de and

[3] `https://wit3.fbk.eu/mt.php?release=2014-01`

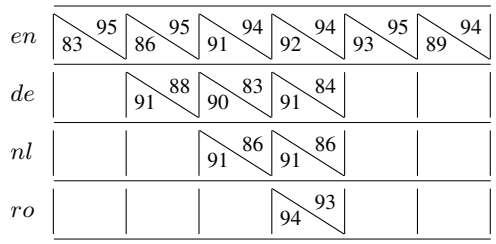
Figure 1: Percentage of TED corpora covered by the common elements in a language pair. A cell represents the language pair row–column, with the coverage of row language given by the bottom subcell and the coverage of the column language given by the upper subcell.



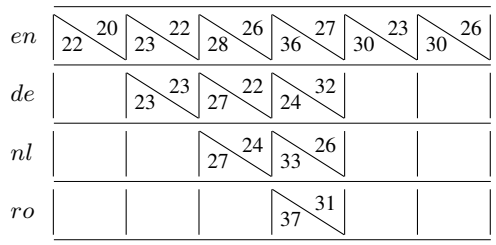
(a) Lemmas



(b) Stems



(c) Metaphone 3 encodings



(d) Babel synsets

nl; Latin with ro, it, es and fr; ALL with the sum of Germanic and Latin; and SMALL and ZERO with the languages considered for the multilingual translation task en, de, nl, ro and it. In general, Germanic languages share more vocabulary (tokens, lemmas and stems) than Latin languages; the disparity in lemmas is more marked in Latin languages: whereas $9,096$ common lemmas cover only a 37% of the corpus, $8,114$ common stems cover a 52% of it. It is remarkable to notice the percentage of common vocabulary in the ALL corpus (30% for tokens, 40% for lemmas and 46% for stems). These high values justify their usage in multilingual systems.

M3 encodings clearly show an excess of ambiguity: 87% of the Germanic and Latin corpora are covered by the common encodings, 70% of the SMALL and ZERO ones. Still, since the information is complementary to the previous elements, we employ it in the translation systems. Finally, the percentage of common BN synsets is higher for the Romance languages (27% vs. 15%). Joining all the languages together decreases this to 12%. Differently to the other factors, BN synsets only cover 4 out of the 10 PoS tags. Besides, they suffer from a *sense effect*: whereas `investigación` in Spanish and `investigation` in English share stem and M3 encoding, the top BabelNet ID is `bn:00067280n` for Spanish and `bn:00047355n` for English because the first sense of the word in the two languages is different.

Figure 1 shows the equivalent analysis per language pair. Notice that the English corpus is the best covered by common lemmas, stems and M3 encodings and that differences between languages can be large, especially when English is involved. According to these numbers, this is the language least rich in lemmas, stems and diversity of pronunciations.

Finally, we analyse the data according to their theme. To do so, we infer the most probable topic for each document with a BTM model learned for 20 topics, so that each topic is the main topic of at least 1% of the training documents. Table 3 shows the characterisation of each topic and the percentage of the corpus described by them. Note that although the extracted topics define different themes, they share keywords. In other words, the diversity in the TED talks is low and themes are close to each other.

4. NMT Systems

Our system is a many-to-many NMT engine trained with Nematus [20]. As done in [3] and similarly to [2], the engine is trained on parallel corpora for the several language pairs simultaneously, 16 pairs for the zero-shot training condition (ZERO) and 20 for the small data training condition (SMALL), with the only addition of a tag in the source sentence to account for the target language "<2trg>". SMALL includes all the pairs generated from the en-de-ro-it-nl languages and ZERO excludes the de-nl and it-ro pairs. In both cases, we only consider those sentences with less than 50 tokens for training, that is 2.113.917 parallel sentences (39.393.037 tokens) in the first case, 1.692.594 sentences (31.671.455 tokens) in the second one.

We consider each token in a source sentence to be represented by (a subset of) the features introduced in the previous sections. The final representation of a word is the concatenation of all its features. This has been named *factored translation* by their similarities with factored translation in statistical machine translation [21] and we use the implementation available in Nematus [8]. The same work [8] explores the inclusion of PoS and subword tags, morphological features, lemmas and syntactic dependency labels as input features for bilingual NMT systems involving, en, de and ro. Here, we

Table 4: Dimensions per factor in the final word embedding for the systems shown in the most-left column.

	token w	PoS p	lemma l	stem s	M3 m	BN b	topic t
w	506	0	0	0	0	0	0
wl	300	0	206	0	0	0	0
ws	300	0	0	206	0	0	0
wm	300	0	0	0	206	0	0
wb	300	0	0	0	0	206	0
wt	496	0	0	0	0	0	10
$wpsm$	300	6	0	100	100	0	0
$wpsmb$	275	6	0	75	75	75	0
$wpsmt$	290	6	0	100	100	0	10
$wpsmbt$	265	6	0	75	75	75	10

extend the model to use more generic factors such as stems and M3 encodings, and interlingual factors such as Babel synsets. The next example shows a truecased phrase annotated with token|PoS|stem|M3|BN in English and German:

en: that|DETERMINER|that|0T|-
's|***VERB***|'s|S|bn:00083181v the|***DETERMINER***|the|0|-
problem|***NOUN***|problem|***PRPLM***|***bn:00048242n***
de: das|PRONOUN|das|TS|-
ist|***VERB***|ist|AST|- das|***DETERMINER***|das|TS|-
Problem|***NOUN***|probl|***PRPLM***|***bn:00048242n***

where boldface emphasises differences and boldface plus italics emphasises similarities. The examples belong to two close languages that share vocabulary and roots. However, problem and Problem would not match without the information on PoS, M3 encoding and BN synset. The example displays other characteristics such as differences of PoS between languages (DETERMINER vs. PRONOUN) for that/das, or lacking BN synset in a language. Differences in the retrieved BN sense are not seen here but should also be considered (portrait|NOUN|portrait|PRTRT|bn:00063682n vs. Porträt|NOUN|porträt|PRTRT|bn:00063683n).

All our systems employ a common vocabulary of $150\,K$ tokens plus $2\,K$ for subword units segmented using Byte Pair Encoding (BPE) [22]. Subwords in the source sentence are annotated with the same factors as the complete word has. As for the parameters, we use a learning rate of 0.0001, Adadelta optimisation, 800 hidden units, a mini-batch size of 100, and drop-out only for hidden layers and input embeddings. We also tie the embeddings in the decoder side to reduce the size of the translation models. The dimension of the word embeddings is always 506, but every model has a different distribution of the dimensions per factor. We refer the reader to Table 4 to see the distribution, where models are named using the letters that represent the factors included.

5. Results and Discussion

Below we report the translation performance for several systems under the small data and zero-shot training conditions. We evaluate systems that combine word tokens (w) with the individual linguistic or semantic factors (wp, wl, ws, wm, wb and wt) and the combination of additional factors ($wpsm$, $wpsmb$, $wpsmt$ and $wpsmbt$). As BabelNet was not within the allowed resources, our submissions for both training conditions were: $wpsm$ (primary, SUB1) and $wpsmt$ (SUB2) and $wpsmbt$ (SUB3) as contrastive.

Results are broken down according to the training condition and language pair: Table 5 shows the BLEU scores on truecased and tokenised translations under the zero-shot training condition and Table 6 shows the equivalent under the small one. First of all, we obtain the results for three different decoding settings on our baseline with only words: two beam sizes, 5 ($w5$) and 10 ($w10$); and an ensemble with the last four models with a beam size of 10 (w). Increasing the beam size is the major source of improvement (1.5 BLEU points on the concatenated test set), and this number is further increased by the ensemble up to 2.4 BLEU points. We analyse the effect of the designed factors over this strong baseline. Since conclusions are analogous, the most detailed analysis is only reported for the zero-shot training condition.

Notice that the global BLEU score for SMALL systems is better than for ZERO mainly because of the zero-shot pairs *de-nl* and *it-ro*. For the other pairs, the enlargement of the multilingual corpus is even harmful both in a baseline with only words and with factored models. When considering the performance of the systems on all the languages simultaneously, the best system is the one exploiting all the features ($wpsmbt$), with a BLEU of 25.46 for ZERO and 25.72 for SMALL. These scores are close to but below our primary submission (25.38 for ZERO and 25.70 for SMALL) which does not consider BN synsets or topic labels.

In general and for most language pairs, BN synsets are the only factor that is able to produce translation improvements by itself, the other ones are in average below the baseline but help to break degeneracies when combined and produce a beneficial effect. PoS tags also achieve a small improvement, but it is non-significant and much less than the one obtained by the authors in [8, 9] for bilingual NMT systems. Stems and lemmas perform equally well in average with only few exceptions: stems are better for translating from *de* or into *nl*, while lemmas are better for translating into *de*. For other language pairs differences are either non-systematic or insignificant. M3 encodings alone are too ambiguous as shown by the high percentage of the corpus covered by common encodings already at the bilingual level (see Figure 1c). Note that in the case of *de*, where the percentage is lower, the encodings do help to increase the performance. As expected, topic information does not imply relevant changes probably due to the low diversity in the topic characteristics (Table 3). However, the fact that contrary to previous research [8, 9, 10] neither PoS tags nor lemmas have a positive impact in the ML-NMT system motivates further experiments with bilingual NMT systems enriched with M3 encodings, BN synsets and topic information.

Figure 2: 2D t-SNE representation of the context vectors of the first 8 source sentences of *tst2010* for system w, wb and $wpsmb$ under the zero-shot training condition. The same sentence has the same colour in different languages.

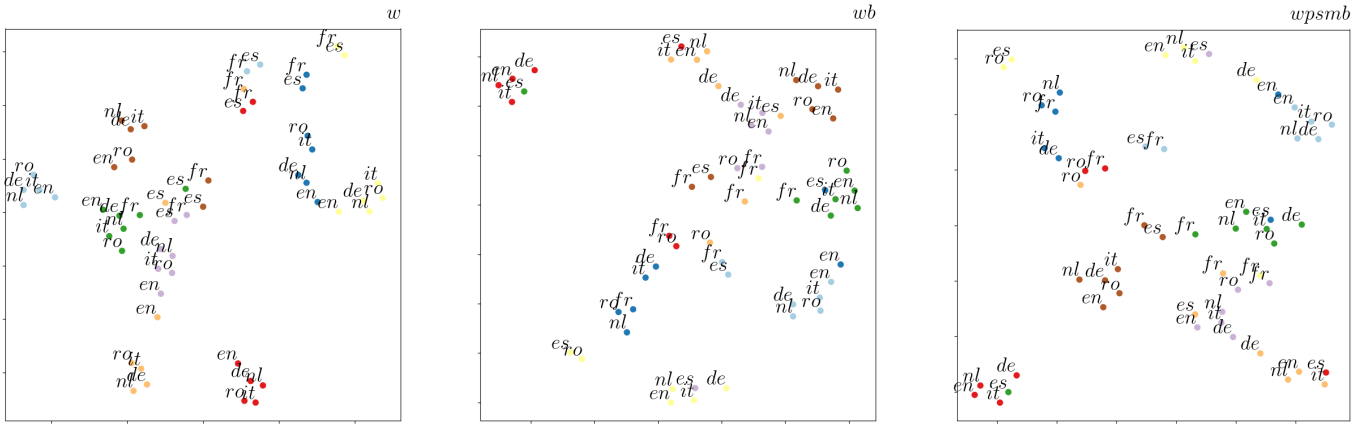


Table 5: BLEU scores on the TED talks *tst2010* obtained with several systems under the zero-shot training condition. The zero-shot pairs, *de-nl* and *it-ro*, are shown at the end. `SUB1`, `SUB2` and `SUB3` were submitted to the shared task.

	beam size		factors + 4-ensembles (beam size 10)										
	w5	*w10*	*w*	*wp*	*wl*	*ws*	*wm*	*wb*	*wt*	*wpsmb*	*wpsm* (`SUB1`)	*wpsmt* (`SUB2`)	*wpsmbt* (`SUB3`)
de2it	18.02	19.20	19.78	19.85	18.95	20.07	19.67	20.28	19.67	**20.35**	20.10	20.05	20.33
it2de	18.05	19.49	19.90	20.03	20.03	19.98	19.98	20.42	20.30	20.22	20.42	20.06	**20.45**
de2ro	15.85	17.57	18.23	17.98	17.16	17.73	17.96	18.46	18.19	**18.60**	18.23	18.00	18.40
ro2de	18.56	20.05	20.87	21.04	**21.52**	21.06	20.93	21.23	20.78	21.34	21.49	21.12	21.41
de2en	30.11	31.67	32.65	32.62	31.66	32.74	32.47	32.97	32.71	33.34	33.11	32.91	**33.51**
en2de	24.61	26.06	27.02	**27.53**	27.30	26.80	27.10	27.26	26.97	27.36	27.15	27.10	27.44
en2it	26.33	27.90	28.88	28.66	28.74	28.97	28.41	**29.35**	28.69	29.06	28.99	28.94	29.34
it2en	31.22	32.56	33.46	33.59	33.85	33.15	32.95	33.20	33.25	33.49	33.53	33.33	**33.87**
en2nl	28.60	30.24	31.27	31.21	31.12	30.87	30.85	31.08	31.26	30.80	30.90	31.17	**31.44**
nl2en	33.86	35.39	36.20	36.61	36.34	36.56	36.16	36.57	36.03	36.92	36.82	36.55	**37.40**
en2ro	23.65	25.28	26.38	26.37	25.67	25.83	25.19	26.18	25.76	26.37	25.85	26.08	**26.47**
ro2en	32.02	33.59	34.34	34.33	34.60	34.40	34.28	34.82	34.34	**35.31**	34.87	34.89	35.09
it2nl	19.03	21.05	21.58	21.65	21.65	21.23	21.25	**21.91**	21.48	21.41	21.79	21.77	21.54
nl2it	19.80	21.23	21.72	21.56	21.34	21.62	21.16	**21.97**	21.71	21.81	21.61	21.84	21.83
nl2ro	17.28	18.42	19.09	18.89	18.98	18.69	18.78	19.39	19.07	19.35	19.09	**19.45**	19.42
ro2nl	19.28	21.21	21.70	21.72	21.76	21.79	21.74	21.65	22.00	22.21	**22.61**	22.20	22.50
de2nl	18.82	21.11	21.75	21.58	20.78	21.76	21.66	22.51	21.62	21.73	**22.29**	22.10	21.90
nl2de	18.82	20.76	21.52	21.81	21.86	21.46	21.62	21.99	21.56	**22.04**	21.81	21.99	21.77
it2ro	16.42	18.14	19.16	19.06	18.94	18.47	18.59	18.94	18.68	**19.51**	19.29	19.13	18.73
ro2it	17.37	19.50	20.04	20.17	20.61	20.38	19.97	20.84	20.28	20.60	**20.94**	20.74	20.32
Concatenation	22.68	24.31	25.08	25.10	24.93	24.96	24.82	25.32	25.01	25.38	25.33	25.30	**25.46**

It is interesting to notice that the final effect of the most *interlingual* factors has not been a better clustering of sentences according to their meaning. Figure 2 shows a qualitative example using a 2D t-SNE representation [23] of the context vectors of 8 sentences in 3 cases. The baseline ML-NMT system w (most-left plot) does already a very good job in locating the sentences in consonance with their semantics. The sentences for the languages used in training lie together for the different languages, while sentences in the unknown languages fr and es group in two specific regions of the space irrespective of their meaning. The effect of BN synsets (middle plot) and M3 encodings (not shown in Figure 2) is to locate fr and es sentences close to the the other Latin languages ro and/or it. By looking at the examples, that means that similarities of the M3 encodings across close languages are too strong to be compared with the most distant languages, and that the top-1 BN synset for a term usually depends on the family that the language belongs to. So,

Table 6: BLEU scores on the TED talks *tst2010* obtained with several systems under the small data training condition. SUB1, SUB2 and SUB3 were submitted to the shared task.

	WZERO	WSMALL	wpsm (SUB1)	wpsmt (SUB2)	wpsmbt (SUB3)
$de2it$	19.78	19.55	20.53	20.14	**20.58**
$it2de$	19.90	19.92	20.05	19.49	**20.26**
$de2ro$	18.23	18.07	18.21	**18.45**	18.05
$ro2de$	20.87	20.82	21.13	20.51	**21.33**
$de2en$	32.65	32.08	**33.44**	32.71	33.24
$en2de$	27.02	26.82	27.22	26.71	**27.37**
$en2it$	28.88	28.83	29.01	28.76	**29.07**
$it2en$	33.46	33.03	33.81	33.70	**33.85**
$en2nl$	31.27	30.72	31.10	31.02	**31.39**
$nl2en$	36.20	35.90	**37.00**	36.48	36.79
$en2ro$	**26.38**	25.57	26.09	25.86	25.99
$ro2en$	34.34	33.86	34.82	34.58	**34.89**
$it2nl$	**21.58**	21.16	21.36	21.30	21.49
$nl2it$	21.72	21.27	**21.82**	21.56	21.72
$nl2ro$	19.09	18.87	19.14	**19.35**	19.16
$ro2nl$	21.70	21.74	21.89	21.61	**22.27**
$de2nl$	21.75	22.97	23.67	**23.90**	23.46
$nl2de$	21.52	23.19	**23.92**	23.64	23.56
$it2ro$	19.16	20.31	**20.84**	20.79	20.67
$ro2it$	20.04	22.41	23.36	22.94	**23.70**
Concat.	25.08	25.12	25.70	25.50	**25.72**

the features designed in this way would maximise their effectiveness within a multilingual system for related languages and, at the light of current results, a better disambiguation and mapping between languages of synsets is necessary for a real interlingual setting. However, the current implementation already achieves statistically significant improvements when used in the en-de-ro-it-nl-NMT system and we show in the following how these features are useful to translate from unseen languages, es and fr. Translation into a new language is still not possible because the system cannot create new words beyond a combination of BPE subunits.

Table 7 summarises the results for es/fr–en translations using the multilingual system under the zero-shot training condition. When translating from English, the BLEU score is close to 1 for all system irrespective of the information they consider —also irrespective of the beam size an number of ensembled models. This score accounts mainly for the common words between the two languages. But, when translating into English, one can obtain a BLEU of 7.25 for $es2en$ translation (5.07 for $fr2en$). The baseline is higher in this case because, as seen in Section 3, English is more sparse than the other languages. Even then, the baseline is improved by more than 4 points of BLEU for $es2en$ and almost 3 points of BLEU for $fr2en$. The major contribution comes from the inclusion of Babel synsets (models wb and $wpsmb$ outperform $wpsm$).

Table 7: BLEU scores for translations involving languages not seen at all in training, es and fr, on the *tst2010* under the zero-shot training condition.

	w	wp	wl	ws	wm	wpsm	wb	wpsmb
$en2fr$	1.11	**1.13**	1.05	0.98	0.98	1.00	1.04	1.04
$fr2en$	2.41	2.77	1.77	3.14	2.84	3.63	**5.07**	5.02
$en2es$	1.29	1.04	1.02	0.98	0.92	0.99	1.02	**1.36**
$es2en$	3.09	3.67	2.61	4.22	3.88	4.87	6.75	**7.25**

6. Conclusions

This paper describes the UdS-DFKI participation at IWSLT 2017. Besides the description of the engines, we analyse the multilingual TED corpus regarding our six characterisation factors: parts of speech, lemmas, stems, Metaphone 3 encodings, Babel synsets and topics.

The most promising feature turned to be BN synsets, especially when combined with other factors. However, our primary submission does not include them as the resource is not allowed in the small data training conditions. Our primary submission, the $wpsm$ system, almost reaches the performance of our best system $wpsmbt$ without any information on the topic and the sense of a token.

BN synsets are the most expensive factor to obtain and they are only queried for a subset of PoS; the common IDs cover between 20% and 40% of the parallel corpora, depending on the language pair. Even then, they improve translations for a 75% of the language pairs and allow beyond-zero-shot translation. Further efforts to deal with multiword expressions and resolve ambiguities in the retrieval of the synsets will be made to enhance the description of the data and facilitate a multilingual learning. Constraining other factors such as M3 encodings and topics to content words could also improve the performance and will be further researched.

7. Acknowledgements

This work was partially funded by the Leibniz Gemeinschaft (SAW-2016-ZPID-2), the BMBF through the project ALL SIDES (01IW14002) and the European Unions Horizon 2020 grant agreement No. 645452 (QT21).

8. References

[1] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, P. Koehn, V. Logacheva, C. Monz, M. Negri, M. Post, R. Rubino, L. Specia, and M. Turchi, "Findings of the 2017 Conference on Machine Translation," in *Proceedings of the Second Conference on Machine Translations (WMT 2017)*, September 2017, pp. 169–214.

[2] T. Ha, J. Niehues, and A. H. Waibel, "Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder," in *Proceedings of the International*

Workshop on Spoken Language Translation, Seattle, WA, November 2016.

[3] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. T. and1 Fernanda B. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," *Transactions of the Association for Computational Linguist*, vol. 5, pp. 339–351, October 2017.

[4] C. España-Bonet, A. C. Varga, A. Barrón-Cedeño, and J. van Genabith, "An Empirical Analysis of NMT-Derived Interlingual Embeddings and their Use in Parallel Sentence Identification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1340–1350, December 2017.

[5] R. Navigli and S. P. Ponzetto, "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network," *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.

[6] L. Wang, Z. Tu, A. Way, and Q. Liu, "Exploiting Cross-Sentence Context for Neural Machine Translation," *CoRR*, vol. abs/1704.04347, 2017.

[7] J. Zhang, L. Li, A. Way, and Q. Liu, "Topic-Informed Neural Machine Translation," in *COLING*, 2016, pp. 1807–1817.

[8] R. Sennrich and B. Haddow, "Linguistic input features improve neural machine translation," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 83–91.

[9] J. Niehues and E. Cho, "Exploiting linguistic resources for neural machine translation using multi-task learning," in *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark*, 2017, pp. 80–89.

[10] M. García-Martínez, L. Barrault, and F. Bougares, "Neural machine translation by generating multiple linguistic factors," in *Proceedings of the 5th International Conference on Statistical Language and Speech Processing (SLSP), Le Mans, France*, 2017, pp. 21–31.

[11] S. Petrov, D. Das, and R. McDonald, "A universal part-of-speech tagset," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012.

[12] R. Agerri, J. Bermudez, and G. Rigau, "IXA pipeline: Efficient and Ready to Use Multilingual NLP Tools," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), may 2014.

[13] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *International Conference on New Methods in Language Processing*, Manchester, UK, 1994, pp. 44–49.

[14] M. F. Porter, "Snowball: A language for stemming algorithms."

[15] M. K. Odell, "The profit in records management," *Systems*, vol. 20, no. 20, 1956.

[16] L. Philips, "Hanging on the metaphone," *Computer Language Magazine*, vol. 7, no. 12, pp. 39–44, December 1990.

[17] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22Nd International Conference on World Wide Web*. New York, NY, USA: ACM, 2013, pp. 1445–1456.

[18] A. K. McCallum, "MALLET: A Machine Learning for Language Toolkit," 2002, http://mallet.cs.umass.edu.

[19] M. Cettolo, M. Federico, L. Bentivogli, J. Niehues, S. Stüker, K. Sudoh, K. Yoshino, and C. Federmann, "Overview of the IWSLT 2017 Evaluation Campaign," in *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, 2017.

[20] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. V. Miceli Barone, J. Mokry, and M. Nadejde, "Nematus: a toolkit for neural machine translation," in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain: Association for Computational Linguistics, April 2017, pp. 65–68.

[21] P. Koehn and H. Hoang, "Factored translation models," 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. 868–876.

[22] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, August 7-12, 2016, Berlin, Germany, Volume 1*, 2016.

[23] L. Van Der Maaten, "Accelerating t-SNE Using Tree-based Algorithms," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, January 2014.

The Samsung and University of Edinburgh's submission to IWSLT17

Pawel Przybysz[1], Marcin Chochowski[1], Rico Sennrich[2], Barry Haddow[2] and Alexandra Birch[2]

[1]Samsung R&D Institute Poland
[2]School of Informatics, University of Edinburgh
{m.chochowski,p.przybysz}@samsung.com
bhaddow@inf.ed.ac.uk, {rico.sennrich,a.birch}@ed.ac.uk

Abstract

This paper describes the joint submission of Samsung Research and Development, Warsaw, Poland and the University of Edinburgh team to the IWSLT MT task for TED talks. We took part in two translation directions, en-de and de-en. We also participated in the en-de and de-en lectures SLT task. The models have been trained with an attentional encoder-decoder model using the BiDeep model in Nematus. We filtered the training data to reduce the problem of noisy data, and we use back-translated monolingual data for domain-adaptation. We demonstrate the effectiveness of the different techniques that we applied via ablation studies. Our submission system outperforms our baseline, and last year's University of Edinburgh submission to IWSLT, by more than 5 BLEU.

1. Introduction

This paper describes the system submission of Samsung R&D Institute Poland and the University of Edinburgh team. The models have been trained with a deep attentional encoder-decoder neural machine translation model using Nematus [1]. In this year's submission, we focused on the core NMT architecture, for which we selected the BiDeep model by [2], training data filtering via language identification and MT-based sentence alignment scores, and domain adaptation with back-translated, domain-filtered monolingual training data, and fine-tuning towards the in-domain training set with MAP-L2 regularization towards the baseline model [3].

Corpus	raw	aligned	filtered
Commoncrawl [4]	2.40M	2.22M	1.62M
Europarl v7 [5]	1.92M	1.90M	1.85M
GoldAligment	509	508	486
MultiUN [6]	0.16M	0.16M	0.15M
News Com. v12 [4]	0.27M	0.26M	0.26M
Opensubtitles2016 [7]	13.88M	12.08M	9.04M
QED Corpus	0.07M	0.07M	0.06M
Rapid 2016	1.33M	1.28M	1.12M
Wikipedia Corpus	2.46M	2.16M	1.18M
WIT3 (in-domain) [8]	0.22M	0.21M	0.20M
Total	22.72M	20.35M	15.47M

Table 1: Admissible parallel corpora used for training, with number of sentences before and after filtering

2. Training data and data selection

2.1. Parallel corpora

For the English-German language pair, we used the corpora listed in Table 1. IWSLT provides a large amount of permissible parallel training data. We performed filtering based on sentence alignment and language identification.

To obtain a sentence alignment score, we follow the idea that we can automatically translate the source text, and use BLEU between the automatic translation and the target side as a feature to predict probable alignments [9]. We trained a Phrase-based Statistical MT model, using significance filtering [10] to remove improbable phrases. Then we translated German sentences into English with a fast Statistical MT engine. Then, a sentence aligner BLEU-Champ[1] was applied to score each parallel training sentence. We also scored each sentence pair with a sentence-level language

[1]https://github.com/emjotde/bleu-champ

recognition tool. After these operations each sentence pair had assigned BLEU-Champ scores and language recognition scores. We selected small subset of 3k sentences from the corpora and performed manual evaluation for each sentence pairs scoring from 1 (very bad) to 5 (very good). Then we trained a regression model to predict human score based on BLEU-Champ and language recognition scores. Finally, we used the regression model to score whole parallel corpora and select potentially good sentences (predicted score above 2). We also removed lines with Wiki markup as we observed negative impact of such lines in our baseline model. Corpus sizes after these steps are shown in column **aligned** and **filtered**. The filtering method removed less than 5% of high quality corpora like News Commentary, but it removed over 50% of Wiki corpus. Additionally monolingual training data from the Commoncrawl [11] was used for creating synthetic parallel training data, see section 2.2 and 2.3 for details.

2.2. Selecting pseudo in-domain monolingual data

In order to reduce the amount of training data and possibly improve domain-adaptation effects, we decided to select data that matched the domain of TED talks based on Moore-Lewis filtering [12]. We followed the procedure described in Edinburgh's submission to IWSLT16 [13]. We used the TED talk data from WIT3 as seed data to create the in-domain language model and a matching amount of randomly chosen out-of-domain data for the contrasting language model.

Lang.	Total	Selected	Avg. score	Sel. score
de	2.9G	20M	0.4639	-0.0935
en	3.0G	20M	0.3797	-0.0394

Table 2: Selected monolingual data. Interpretation of figures is the same as for parallel data.

As seen in Table 2 we selected 20M sentences for back-translation from much larger original corpora of 2.9G and 3.0G sentences.

2.3. Preprocessing and subword units

To avoid the large-vocabulary problem in NMT models [14], we used byte-pair-encoding (BPE) to achieve open-vocabulary translation with a fixed vocabulary of subword symbols [15]. For all languages we set the number of merge operations to 90k. Segmentation into subword units was applied after any other preprocessing step for joint source and target vocabulary. We set vocabulary threshold to 50.

2.4. Back-translation

Corpus	size	oversapling
WIT3 (in-domain) [8]	0.20M	4.17M
Other parallel	15.27M	15.27M
Synthetic	19.57M	19.57M
Total	-	39.01M

Table 3: Final corpora used for training including admissible, filtered parallel corpora, oversampled in-domain corpus and synthetic, backtranslated data.

Back-translated monolingual in-domain data has been shown to be very beneficial when added to the parallel training data [16]. We back-translated the selected monolingual data with shallow, single layer NMT model trained on raw, permissible parallel data. We call it a baseline model hereafter. The model was trained with Nematus and translation was done with Marian [17]. We present the size of the final training corpora in table 3.

3. Neural translation systems

The neural machine translation system is an attentional encoder-decoder [18], which has been trained with Nematus [1]. There have been a number of papers showing that deeper models in machine translation lead to higher quality output. We apply the BiDeep model [2], which is a combination of stack RNNs and deep recurrent RNNs. Each cell in the stack RNN consists of multiple GRU cells, as illustrated in Figure 1. We use 4 stacks of RNNs with deep recurrent GRUs with a transition depth of 2.

In these experiments we followed the implementation details described in Edinburgh's WMT 2017 submission [19]. Important features which we used were: layer normalisation, BPE Version 2 with filtering of rare subwords, dynamic batching and using tied embeddings.

Additionally, to reduce training time we experimented with data parallelism on multi-GPU. Most of the approaches ([20],[21]) use SGD optimizer with centralized parameters which all workers read and

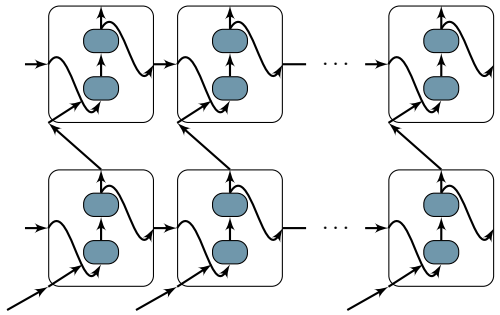


Figure 1: Illustration of BiDeep RNN architecture [2]. The architecture consists of a stack of layers of recurrent cells; each cell is composed of multiple GRU transitions.

update. We decided to use Adam optimizer instead as it was shown to converge faster. Unfortunately, the additional parameters in Adam make centralized parameter approach ineffective due to large copy overhead (one needs to store also means and variances of all parameters). In our implementation there is no centralized copy, instead each worker holds its own copy of all parameters. Each worker computes gradients on its own batch and only the gradients are summed over all workers and shared among them synchronously (we use nccl library[2]). Next, each worker independently updates its model parameters using the shared gradient. The data copied between workers is thus minimal. Since all workers are initialized equally, after the update they all still hold the same parameter values. Using N workers in this implementation can be seen as single worker case with N-times larger batch size. Thus to compare results one needs to set the training parameters like validation frequency accordingly. Table 4 compares trainings results for different number of GPUs. The results refer to de2en, BiDeep model training on filtered corpus using single node server with 8 GeForce 1080Ti. We did not present the 8-GPU case due to hardware problems with one card. The results show that our approach scales well. With increasing number of cards the throughput measured in words per second scales nearly linearly while the training time significantly reduces and the achieved BLEU is in close range.

To train the models for IWSLT 2017 submission we used 3 servers with 8 GPUs each (1 with GeForces 1080Ti, and 2 others with Teslas K80). The number of GPU used in particular training varied between 1 and 8 depending on resources availability.

[2]https://github.com/NVIDIA/nccl

	BLEU	time [h.]	words/sec.	overhead
1GPU	35.01	162.2	1082	8.3%
2GPU	34.85	103.0	1890	15.6%
3GPU	35.45	80.2	2950	17.6%
4GPU	35.30	67.1	3586	18.6%
6GPU	35.16	48.2	5315	23.0%

Table 4: Multi-GPU BiDeep model traininig statistics for different number of GPU. Training performed on de2en filtered corpora. The first column reports the best BLEU, the second convergence time, the third number of processed words per second. The last one is the overhead added by using the multi-gpu mechanism (reduce-all synchronization). Note the non-zero overhead even with 1 GPU.

3.1. Training, tuning and ensembling

We perform several steps of fine-tuning of the general models, using continued training with a new selection of training data and training parameters.

For each translation direction we run several independent trainings with slightly different data and parameters to get variety of final models for most successful ensembling. In all trainings we used TED test set from 2015 as a validation set.

As an example, the training of two of the de2en models (Fig. 2) used in the final ensemble, was started on the filtered parallel training data with 20 million in-domain backtranslated sentences and TED corpus oversampled 20 times. We trained this to convergence. After convergence, we enabled dropout, with both embedding dropout and hidden layer dropout set to 0.2, and continued training until results converged again.

We repeated this procedure two times for each direction, hoping that two independent runs will give us a better ensemble model than a checkpoint ensemble.

Finally we performed a fine-tuning step where we tuned to just the TED corpus with dropout and MAP-L2 regularization towards the previous model [3]. We also performed careful validation and early stopping. For fine-tuning we selected best and second best models for each independent run from the previous step.

For the final system we choose the 4 fine-tuned models that gave the best ensembles. In de2en direction

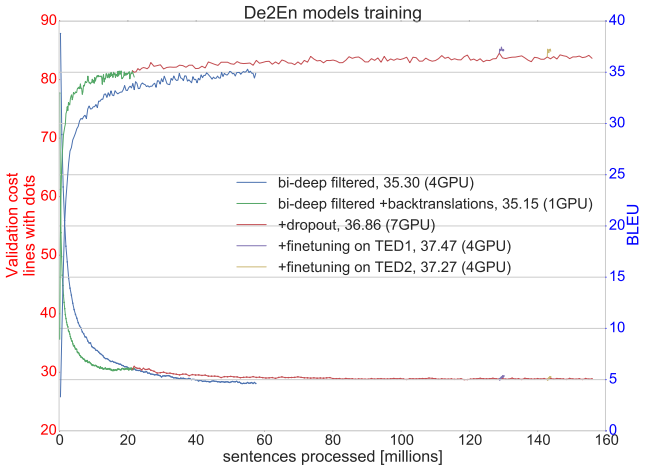


Figure 2: German-to-English models training progress. Plot shows the BLEU (straight lines) and validation error (lines with dots) on tst2015. Colors represent successive training parameters modifications. The number in plot label is the best BLEU score for particular training configuration.

the best model was a checkpoint ensemble and in en2de independent ensemble (from 2 independent models).

3.2. Spoken Language Translation

We also participated in the IWSLT Lectures spoken language translation task. This task consisted of three German lectures and two English lecutres and ten English TED talks. The test sets had no segmentation or punctuation. Our submission used an English and a German punctuation model provided by the SUMMA project. These models are essentially neural machine translation models which are trained to predict commas, full stops, question marks, exclamation marks and three dots [22, 23]. The punctuated source was then translated using the reranked ensembles described above.

4. Results

We present results in table 5. For the progress set, we also report BLEU of the University of Edinburgh submission to IWSLT16 [13], which was ranked first for en-de, and third for de-en. For results comparing our MT and SLT submissions to other systems in IWSLT please see the overview paper [24].

We performed extensive ablation studies. Our results confirm the effectiveness of deep models, which yield an improvement of 0.8-1.4 BLEU. They also give evidence for the sensitivity of our models towards

Translation	Progress set (2016)		Test set (2017)	
	de-en	en-de	de-en	en-de
IWSLT16 [13]	32.56	-	27.34	-
baseline	32.52	26.05	27.84	24.33
BiDeep raw	33.92	27.27	29.28	25.14
BiDeep filtered	34.07	27.66	29.94	25.61
+backtranslations	36.27	28.81	30.93	25.24
+dropout	36.50	29.83	31.41	26.66
+finetune on TED	37.08	30.21	32.26	27.38
+checkpoint ens.	37.61	30.34	32.37	27.56
independent ens.	37.56	29.91	32.71	27.23
+right to left	**37.85**	**30.93**	**33.08**	**28.00**

Table 5: Results for the IWSLT TED translation task (BLEU). Submitted system highlighted in bold.

training data noise, and models trained on filtered data outperform those trained on the full training coropra by 0.5–0.7 BLEU. Our use of back-translated data improved performance for de-en (+1 BLEU), and on the 2016 progress set for en-de (+0.8 BLEU), but not on the 2017 test set (-0.4 BLEU). Fine-tuning towards the TED training data remains an effective strategy (+0.8 BLEU), as does ensembling and right-to-left reranking.

In total, we report improvements of over 5 BLEU over last year's IWSLT submission by the University of Edinburgh [13], which was also based on Nematus and used a similar strategy for preprocessing and training. We attribute this to technical improvements in our neural network architecture, such as layer normalisation and BiDeep networks, better regularization during fine-tuning, and the use of more out-of-domain training data, and the use of reranking with a right-to-left model. We note that even our best single model outperforms last year's ensemble of 5 models by more than 4 BLEU.

5. Conclusions

This paper describes the joint submission of Samsung R&D Institute Poland and the University of Edinburgh team to the IWSLT MT task for TED talks, for the translation directions en-de and de-en. We report strong baseline results that are on par with last year's University of Edinburgh submission to IWSLT. Our experimental results confirm the effectiveness of the BiDeep NMT architecture, and of domain adaptation

via back-translated monolingual training data, and regularized fine-tuning towards an in-domain training set. Our results also highlight the importance of clean training data for NMT training, and we obtain better translation quality with a filtered subset of the permissible parallel training data. Our submission system outperforms our baseline, and last year's University of Edinburgh submission to IWSLT, by more than 5 BLEU.

6. Acknowledgments

The research presented in this publication was conducted in cooperation with Samsung Electronics Polska sp. z o.o. - Samsung R&D Institute Poland. We thank Antonio Valerio Miceli Barone for his illustration of the BiDeep architecture. This work was supported by the H2020 project SUMMA, under grant agreement 688139.

7. References

[1] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. V. Miceli Barone, J. Mokry, and M. Nadejde, "Nematus: a Toolkit for Neural Machine Translation," in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain: Association for Computational Linguistics, April 2017, pp. 65–68. [Online]. Available: http://aclweb.org/anthology/E17-3017.pdf

[2] A. V. Miceli Barone, J. Helcl, R. Sennrich, B. Haddow, and A. Birch, "Deep Architectures for Neural Machine Translation," in *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers*. Copenhagen, Denmark: Association for Computational Linguistics, 2017. [Online]. Available: https://arxiv.org/pdf/1707.07631

[3] A. V. Miceli Barone, B. Haddow, U. Germann, and R. Sennrich, "Regularization techniques for fine-tuning in neural machine translation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, 2017. [Online]. Available: https://arxiv.org/pdf/1707.09920.pdf

[4] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, M. Huck, C. Hokamp, P. Koehn, V. Logacheva, C. Monz, M. Negri, M. Post, C. Scarton, L. Specia, and M. Turchi, "Findings of the 2015 Workshop on Statistical Machine Translation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 1–46. [Online]. Available: http://aclweb.org/anthology/W15-3001

[5] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," in *Conference Proceedings: the tenth Machine Translation Summit*, AAMT. Phuket, Thailand: AAMT, 2005, pp. 79–86. [Online]. Available: http://mt-archive.info/MTS-2005-Koehn.pdf

[6] A. Eisele and Y. Chen, " MultiUN: A Multilingual Corpus from United Nation Documents," in *Proceedings of the Seventh conference on International Language Resources and Evaluation*, 2010, pp. 2868–2872.

[7] P. Lison and J. Tiedemann, "OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, N. C. C. Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, Eds. Paris, France: European Language Resources Association (ELRA), may 2016.

[8] 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.

[9] R. Sennrich and M. Volk, "MT-based Sentence Alignment for OCR-generated Parallel Texts," in *The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*, Denver, Colorado, USA, 2010. [Online]. Available: https://amta2010.amtaweb.org/AMTA/papers/2-14-SennrichVolk.pdf

[10] W. Ling, J. Graça, I. Trancoso, and A. Black, "Entropy-based Pruning for Phrase-based Ma-

chine Translation," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, ser. EMNLP-CoNLL '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 962–971. [Online]. Available: http://dl.acm.org/citation.cfm?id=2390948.2391054

[11] C. Buck, K. Heafield, and B. van Ooyen, "N-gram Counts and Language Models from the Common Crawl," in *Proceedings of the Language Resources and Evaluation Conference*, Reykjavk, Icelandik, Iceland, May 2014.

[12] R. C. Moore and W. Lewis, "Intelligent Selection of Language Model Training Data," in *Proceedings of the ACL 2010 Conference Short Papers*, ser. ACLShort '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 220–224. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858842.1858883

[13] M. Junczys-Dowmunt and A. Birch, "The University of Edinburgh's systems submission to the MT task at IWSLT," in *Proceedings of IWSLT*, 2016.

[14] T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the Rare Word Problem in Neural Machine Translation," in *ACL*, 2015.

[15] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 1715–1725. [Online]. Available: http://www.aclweb.org/anthology/P16-1162.pdf

[16] R. Sennrich, B. Haddow, and A. Birch, "Improving Neural Machine Translation Models with Monolingual Data," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 86–96. [Online]. Available: http://www.aclweb.org/anthology/P16-1009.pdf

[17] M. Junczys-Dowmunt, T. Dwojak, and H. Hoang, "Is Neural Machine Translation Ready for Deployment? A Case Study on 30 Translation Directions," in *Proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*, Seattle, WA, 2016. [Online]. Available: http://workshop2016.iwslt.org/downloads/IWSLT_2016_paper_4.pdf

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *CoRR*, vol. abs/1409.0473, 2014.

[19] R. Sennrich, A. Birch, A. Currey, U. Germann, B. Haddow, K. Heafield, A. V. Miceli Barone, and P. Williams, "The University of Edinburgh's Neural MT Systems for WMT17," in *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, Copenhagen, Denmark, 2017. [Online]. Available: https://arxiv.org/pdf/1708.00726

[20] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, *et al.*, "Large scale distributed deep networks," in *Advances in neural information processing systems*, 2012, pp. 1223–1231.

[21] S. Zhang, A. E. Choromanska, and Y. LeCun, "Deep learning with elastic averaging sgd," in *Advances in Neural Information Processing Systems*, 2015, pp. 685–693.

[22] O. Klejch, P. Bell, and S. Renals, "Punctuated transcription of multi-genre broadcasts using acoustic and lexical approaches," in *SLT*, 2016.

[23] O. Klejch, P. Bell, and S. Renals, "Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features," in *ICASSP*, 2017.

[24] M. Cettolo, M. Federico, L. Bentivogli, J. Niehues, S. Stüker, K. Sudoh, K. Yoshino, and C. Federmann, "Overview of the IWSLT 2017 Evaluation Campaign," in *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, 2017.

The RWTH Aachen Machine Translation Systems for IWSLT 2017

Parnia Bahar, Jan Rosendahl*, Nick Rossenbach and Hermann Ney*

Human Language Technology and Pattern Recognition Group
Computer Science Department
RWTH Aachen University
Aachen, Germany

<surname>@cs.rwth-aachen.de

*Authors contributed equally

Abstract

This work describes the Neural Machine Translation (NMT) system of the RWTH Aachen University developed for the English↔German tracks of the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2017. We use NMT systems which are augmented by state-of-the-art extensions. Furthermore, we experiment with techniques that include data filtering, a larger vocabulary, two extensions to the attention mechanism and domain adaptation. Using these methods, we can show considerable improvements over the respective baseline systems and our IWSLT 2016 submission.

1. Introduction

We describe the Neural Machine Translation (NMT) system of the RWTH Aachen University developed for the evaluation campaign of International Workshop on Spoken Language Translation (IWSLT) 2017. We have participated in the unofficial bilingual Machine Translation (MT) track for the German→English and English→German language pairs. The in-house NMT system incorporates various state-of-the-art extensions.

For the IWSLT 2016 evaluation campaign, RWTH Aachen utilized different translation systems [1] including a state-of-the-art phrase-based system, a neural machine translation system and the joint translation and reordering (JTR) model [2]. Furthermore, last year's system applied feed-forward and recurrent neural language and translation models for reranking. The attention-based approach had been used for reranking the n-best lists for both the phrase-based and the hierarchical setups. On top of these systems, a system combination enhances the translation quality by combining individually trained systems. For the IWSLT 2017 evaluation campaign, we developed the systems only based on the NMT approach as it has shown the most promising results among all.

This paper is organized as follows. In Section 2, we briefly address our preprocessing which differs from our previous submissions [3, 1]. Section 3 describes the details of the NMT systems, the baseline, our optimization techniques as well as two extensions to the attention mechanism. Our experiments for each track are summarized in Section 4.

2. Preprocessing

2.1. Preprocessing

Recent studies [4] showed that attention-based neural network systems do not benefit from several established preprocessing features such as compound splitting and POS-based word reordering. Therefore, we decided to employ a simpler version of preprocessing which uses only tokenization, frequent casing, and simple categories. In this approach, numbers are not mapped to a specific category-token but are treated like regular words instead.

All words and numbers are split into subword units using byte-per-encoding (BPE)[1] introduced by [5]. We use 90k BPE merging operations trained jointly on the concatenated source and target training data. In the preprocessing, we do not distinguish if a language is seen as a source or target language.

2.2. Data Filtering

In order to remove incorrectly aligned sentence pairs, we drop all training samples for which the length of the source sentence exceeds the length of the target sentence by more than about 70%. We applied this method for both translation directions. In the following we describe the effects for the English→German task. The length comparison is executed on the word level and results in the total removal of 1.7M sentences, i.e. 8% of the total training data. The majority of removed sentence pairs are part of the Common Crawl (300k sentences i.e. 14% of Common Crawl) and the OpenSubtitles corpora (1000k sentences i.e. 8% of OpenSubtitles). The removal rates for the individual corpora can be found in Table 1.

[1] https://github.com/rsennrich/subword-nmt

Table 1: Effect of filtering on the individual training corpora on the example of English→German.

	Common Crawl	Europarl	UN	News Comment	OpenSub	QED	TED	Wiki	**Total**
# sentences	2,399,123	1,920,209	162,981	216,190	13,430,645	72,747	206,112	2,459,662	**20,867,669**
# removed	336,248	38,032	3,323	3,547	1,056,628	4,343	2,560	218,860	**1,663,541**
% removed	14.02%	1.98%	2.04%	1.64%	7.87%	5.97%	1.24%	8.90%	**7.97 %**

3. Neural Machine Translation System

The best performing system provided by RWTH Aachen is an attention-based recurrent neural network similar to [6]. Provided with a source f_1^J and a target sequence e_1^I, NMT models the conditional probability of the target given the source. The model itself consists of an encoder which produces a continuous representation of the input sequence f_1^J, an attention mechanism which allows the system to focus on certain words during the translation and a decoder which returns a probability distribution over all possible target tokens for every time step.

3.1. Baseline System

We use the attention-based NMT system as our baseline. In our setup, all words are projected into a 620-dimensional embedding space both on the source and on the target side. The bidirectional encoder and the unidirectional decoder consist of LSTM nodes [7] with peephole connections using 1000 cells. The output layer of the networks is a two-layer maxout [8] followed by a softmax operation that creates a probability distribution over the target vocabulary. We use the additive attention with tanh activation function as proposed in [6] followed by the softmax to compute the attention weights.

3.2. Stacked Layers

In this architecture, we experiment with two stacked LSTM layers in both encoder and decoder to build a deeper model. We connect all internal states of the first LSTM layer to the second. This approach is applied both in the bidirectional encoder and the unidirectional decoder.

3.3. Optimization

Since the learning trajectory considerably depends on the optimization technique, the optimizer plays an important role in fast convergence, training stability and reliable performance. It is desired to have a fast convergence to a zone in which a good local minimum is located. After that, the algorithm shrinks the learning rate to get a finer search pattern and converge to a suitable model within the located area.

As proposed in [9], we start the training using Adam [10] with a learning rate of 0.001 up to 600k iterations. Afterwards the learning rate of the Adam optimizer is scaled down by the factor of 0.75 every 20k iterations. In the following, we refer to this approach by annealing Adam.

3.4. Fertility Feedback

One of the problems arising from the attention-based sequence-to-sequence model, which is used as our baseline, is that there is no explicit alignment or coverage information. The attention weights are included in the context vector and there is no guarantee that the network can extract this information in the next attention computation. One of the proposed solutions [11] is to feed back the sum of the alignments over the past decoder steps. This information is added to the computation of the attention energies for each source position. Hence, in each decoder step this sum indicates how much attention has been given to the source position j up to step i. The feedback term $\hat{\beta}_{i,j}$ is expressed as:

$$\hat{\beta}_{i,j} = \sum_{k=1}^{i-1} \alpha_{k,j} \tag{1}$$

One might simply use $\hat{\beta}_{i,j}$ as an additional information in order to compute the attention energies. Instead, we use a fertility parameter that determines how many target words should be generated by a single source word. The concept of fertility has been introduced in IBM Model 3 and can be integrated into neural networks [11, 12].

Let's assume a single word should be translated twice, then $\hat{\beta}_{i,j}$ can be divided by a factor of 2. This normalizes the sum presented in Equation 1, such that the network can use the information whether the current word is over- or under-translated. Therefore, $\beta_{i,j}$ is defined as:

$$\beta_{i,j} = \frac{1}{\phi_j} \sum_{k=1}^{i-1} \alpha_{i,j} \tag{2}$$

where ϕ_j refers to the fertility of f_j. This term depends on the encoder states, because it can vary if the word is used in a different context. Like [11] in our model ϕ_j is defined as:

$$\phi_j = N \cdot \sigma(\upsilon_\phi^\top \cdot h_j) \tag{3}$$

where N specifies the maximum value for the fertility which is set to 2 in our experiments. This value is included in the calculation of the attention energies $e_{i,j}$:

$$e_{i,j} = \upsilon^\top \tanh\left(W s_{i-1} + U h_j + V \beta_{i,j}\right) \tag{4}$$

where h_j and s_i denote the output of the encoder and the decoder state respectively. W, U, V and υ are the weight matrices.

3.5. Convolutional Feedback

In the standard attention-based model, there is no dependency on the source position while computing the attention weights. Several authors argue [13, 14] that this independence assumption does not hold for monotonous alignments as can be found in speech recognition. Although the alignments in machine translation are not monotonous in general, we still encounter many cases of local monotonicity in many languages. Convolutional attention feedback tries to encounter such problems by putting an explicit focus on the source positions around j when generating the j-th target word. Formally, it computes feature vectors γ_i by applying a one-dimensional convolutional operation over the attention weights from the last decoder step:

$$\gamma_i = G * \alpha_{i-1} \tag{5}$$

where $G \in \mathbb{R}^{N \times 2k+1}$. This leads to N vectors, one for each filter that has been applied. Every filter moves over a window of size $2k+1$ that is centered at position j, i.e.:

$$\gamma_{i,j} = \sum_{l=j-k}^{j+k} G_{n,j-l} \cdot \alpha_{i,l} \qquad \text{for all } n = 1, \cdots, N. \tag{6}$$

The result of this is used as a feedback term to compute the attention weights in the current decoder step:

$$e_{i,j} = \upsilon^\top \tanh\left(W s_{i-1} + U h_j + V \gamma_{i,j}\right). \tag{7}$$

We use 5 filters with a window of size 5 in our experiments that include convolutional feedback. Again we use h_j respectively s_i to denote the output of the encoder and the internal state of the decoder.

4. Experimental Evaluation

For the evaluation, we carry out experiments on two translation tasks: German→English and English→German. The translation systems are built using our in-house implementation of the attention-based NMT approach which relies on the Blocks[2] framework [15] and Theano[3] [16].

All systems are trained on the filtered bilingual data as described in Section 2.2 and no monolingual data. In order to adapt our system to the domain of TED Talks, we add the TED corpus eleven times and the QED corpus six times to our training data. This results in a training set of 21.6M parallel sentence pairs.

Before training, we shuffle the training samples once and use mini-batches of 50 sentence pairs while sentences longer than 65 subwords are dropped. The processing of one mini-batch is called an iteration. The networks are trained for up to 600k iterations and equipped with the various features presented in Section 3. We evaluate the models every 10k iterations.

Throughout our experiments, we observe that employing the Adam annealing scheme consistently gives us strong improvements of at least 1.5% BLEU over the pure Adam optimizer. Similar gains can be achieved by averaging the weights among the four best models of a single training run as described in the beginning of [17]. Both methods are applied to improve upon a weak Adam endpoint. Hence, we always pick the option that leads to a better average BLEU score. The results of the other method are omitted in this paper for the sake of brevity.

We try to fine-tune the models on the indomain data which consists of the TED corpus to which `TED.tst2011`, `TED.tst2012` and `TED.tst2013` sets were added.

Decoding is performed using beam search with a beam size of 12 and the scores are normalized w.r.t the length of the hypotheses.

We use `TED.dev2010` consisting of 888 sentences as our validation set and evaluate our models on `TED.tst2010`, `TED.tst2014` and `TED.tst2015` as unseen test sets. The systems are evaluated using case-sensitive BLEU [18] computed by `mteval-v13a` [4], TER [19] computed by `tercom` [5] and CharacterTER [20] which we abbreviate to CTER[6].

To avoid the out of vocabulary problem, we use the joint BPE [5] to convert sentences into the sequences of subwords on both the source and the target side. In both tasks, the number of joint-BPE merging operations is 90k.

4.1. German→English

Based on the work done in [4], we equip the German→English baseline with two layers of stacked LSTMs in both the encoder and the decoder which is referred to as `multilayer enc-dec baseline`. The total number of parameters for this setup is about 220M. All networks are trained with 30% of dropout for better regularization. The results are depicted in Table 2, Row 1. After training the network and reaching convergence, we apply annealing Adam as mentioned in Section 3.3 for additional 300k iterations (Row 2 in Table 2). As shown, this strategy results in improvements up to 2.4% BLEU score, 1.4% TER and 1.4% CTER averaged over the four test sets.

Using additional information from previous attention states by employing fertility feedback, we gain 0.5% BLEU and 0.1% TER on average. The results in Row 3 of Table 2 have been obtained by applying annealing Adam. On top of this model, we fine-tune the system. Here, we pick the best model and retrain it using the indomain TED data discussed before for around 20 epochs. Surprisingly, fine tuning does not help and even hurts slightly in terms of TER. One of the reasons is that we have already weighted our indomain data in the training data such that any further fine tuning does not affect the learning trajectory. In the other words, the model

[2] https://github.com/mila-udem/blocks-examples

[3] http://deeplearning.net/software/theano/

[4] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl

[5] http://www.cs.umd.edu/ snover/tercom/

[6] https://github.com/rwth-i6/CharacTER

Table 2: Results measured in BLEU [%], TER [%] and CTER [%] for the individual systems for the German→English MT task.

#	System	TED.dev2010			TED.tst2010			TED.tst2014			TED.tst2015		
		BLEU	TER	CTER	BLEU	TER	CTER	BLEU	TER	CTER	BLEU	TER	CTER
1	multilayer enc-dec baseline	34.3	44.9	46.1	33.0	46.0	48.1	31.3	49.0	51.6	31.1	48.2	50.6
2	+ annealing Adam	36.4	43.2	45.2	35.0	44.3	46.3	33.8	46.2	49.3	34.0	46.0	48.3
3	+ fertility feedback	37.0	42.4	44.8	35.6	43.7	45.3	33.9	46.2	49.0	34.6	45.4	48.4
4	+ fine tuned	36.2	43.2	44.9	35.6	44.0	45.2	33.9	46.6	48.3	34.5	45.7	48.6
5	+ convolutional feedback (averaged4)	36.2	43.0	45.4	34.9	44.1	46.5	33.1	46.4	49.2	33.1	45.8	49.0
6	ensemble 2, 3, 5	38.3	41.2	43.4	37.3	42.3	44.3	35.5	44.9	47.8	35.5	44.5	47.6

is smoothly adapted to the TED domain from the first iterations. We also apply convolutional feedback as explained in Section 3.5, and average the best four models of a single training run (see Row 5). As it can be seen, convolutional feedback is slightly better in terms of TER and hurts in terms of BLEU compared to Row 2.

Finally, we build an ensemble [21] of different architectures including two multilayer enc-dec baseline, fertility feedback and convolutional feedback models. Ensemble improves the translation performance compared to the best system (Row 3) by 1.4% in terms of BLEU, 0.6% TER and 0.4% CTER on average.

4.2. English→German

For the English→German task we start with a simple baseline described in Section 3.1 which employs a single LSTM-layer for both the bidirectional encoder and the decoder. The model is trained using the Adam algorithm for 600k iterations and by default, no dropout is applied.

On top of this baseline, we add various feature combinations. Results are shown in Table 3. Adding dropout to the baseline system yields an average improvement of 0.7% BLEU. Based on this, we continue the training with the annealing Adam for 300k iterations which gives us an improvement of 2.5% BLEU.

Furthermore, we train a series of models that utilize the fertility feedback presented in Section 3.4. Adding this feature on top of the baseline system yields an improvement of 0.3% BLEU (Table 3, Row 4). Adding a second LSTM-layer to both the encoder and the decoder leads to an average gain of 0.2% BLEU and 1.1% TER.

Again, we observe that it is important to keep on training for 300k iterations with a small learning rate as this boosts our performance by 2.3% BLEU (Table 3, Row 7). Usually, the models extracted from a training run are among the last models saved during the 600k iterations. Therefore, the effect of the annealing Adam scheme can be attributed to an insufficiently small learning rate or a model that is not fully converged. However, it hurts the performance of the model if we further continue training on the regular training data.

We fine-tune the models either on the indomain data or an expanded version which contains the QED corpus as well. Both approaches led to almost no change w.r.t BLEU and TER. As in the case of the German→English system, we conclude that due to the weighting of the TED data, additional domain adaptation is of little use. However the models that are fine-tuned on the TED corpus perform a little bit stronger in the final ensemble which is why we decide to keep them.

In total, we combine fertility feedback, multi-layered encoder and decoder as well as dropout with an annealing version of Adam to get an improvement of 3.3% BLEU (Table 3, Row 9). Surprisingly, by averaging the four best fertility feedback models (Table 3, Row 5), we obtain a smaller model that has been trained for a much shorter period of time but performs only 0.3% BLEU worse than to the fine-tuned one on average.

Combining several of the systems in one ensemble led to an average improvement of 1.5% BLEU and 1.4% TER over our single best system.

4.3. Final Results

Compared to last year's submission, we have completely moved towards pure neural MT systems. Although last year's system contains a phrase-based system in combination with the JTR model [2], neural language and translation models as well as NMT systems, the results are improved by 2.3% BLEU and 1.8% TER for the `TED.tst2010` set and by 1.3% BLEU and 1.6% TER on the `TED.tst2014` set as shown in Table 4. Furthermore, the pure NMT system for 2017 submission shows a huge improvement compared to the 2015 submission in which the NMT model had only been used in the reranking of the *n*-best lists for both phrase-based and hierarchical setups.

The performance on the `TED.tst2016` and `TED.tst2017` test sets is shown in Table 5. We evaluate our hypothesis via the IWSLT 2017 evaluation server.

5. Conclusion

The RWTH Aachen has participated in two bilingual MT tracks for the German→English and English→German IWSLT 2017 evaluation campaign. The 2017 submission includes neural models only opposed to last year's system included the NMT system and the phrase-based system. The

Table 3: Results measured in BLEU [%], TER [%] and CTER [%] for the individual systems for the English→German MT task.

#	System	TED.dev2010			TED.tst2010			TED.tst2014			TED.tst2015		
		BLEU	TER	CTER	BLEU	TER	CTER	BLEU	TER	CTER	BLEU	TER	CTER
1	baseline	26.0	55.3	50.6	26.4	54.3	51.1	24.6	57.1	53.7	27.2	55.2	51.1
2	+ dropout	26.7	53.7	50.5	27.4	53.0	50.8	25.3	56.4	53.9	27.4	55.1	51.3
3	+ annealing Adam	28.6	52.1	47.1	30.2	51.0	48.5	27.9	53.9	50.7	30.2	53.0	48.4
4	+ fertility feedback	26.4	54.3	50.6	26.7	54.0	51.5	25.3	56.7	53.8	27.0	55.5	50.9
5	+ average4	28.9	50.9	47.0	29.9	50.9	47.7	27.2	54.3	50.5	29.9	52.4	47.9
6	+ multilayer enc-dec	26.5	53.5	50.0	27.1	53.5	51.3	25.2	56.2	53.5	27.3	54.5	50.7
7	+ annealing Adam	28.8	51.1	46.8	29.6	51.2	48.0	27.6	54.0	50.0	29.9	52.6	48.3
8	+ fine tuned	28.4	51.4	47.0	29.8	51.2	47.6	27.5	54.3	49.8	29.9	52.7	47.4
9	+ dropout	28.9	51.4	47.3	30.1	50.8	47.4	27.6	54.1	50.3	30.5	52.3	46.8
10	ensemble 3, 5, 8, 9	30.3	49.9	45.2	31.8	49.5	45.9	29.2	52.8	48.8	31.5	51.1	45.9

Table 4: Comparison to last years' German→English MT task submissions. Results measured in BLEU [%], TER [%] and NIST.

System	TED.tst2010			TED.tst2014		
	BLEU	TER	CTER	BLEU	TER	CTER
2015-Submission [3]	31.9	47.6	45.5	-	-	-
2016-Submission [1]	35.0	44.1	42.7	34.2	46.5	46.9
2017-Submission	37.3	42.3	44.3	35.5	44.9	47.8

Table 5: Results measured in BLEU [%], TER [%] and NIST on `TED.tst2016` and `TED.tst2017`.

MT Task	TED.tst2016			TED.tst2017		
	BLEU	TER	NIST	BLEU	TER	NIST
De→En	35.38	44.48	7.8947	30.22	49.44	7.1608
En→De	28.09	55.23	6.5995	25.12	59.09	6.1239

baseline systems for the MT track utilize our state-of-the-art attention-based neural machine translation. We are able to further improve translation by applying a multilayer encoder and decoder and increasing the number of subword units. Using refinements of the attention mechanism to feedback more alignment information leads to better results. A significant gain is achieved by the annealing scheme based on Adam and the ensemble of different NMT systems.

In total, we achieve a performance of 35.5% BLEU and 44.5% TER on the `TED.tst2015` data set of the German→English task. Compared to our 2016 submission, this is an improvement by 1.3% BLEU and 1.6% TER. For the English→German task our state-of-the-art system produces a score of 31.5% BLEU and 51.1% TER on `TED.tst2015`.

6. Acknowledgements

The work reported in this paper has been funded by three projects, SEQCLAS, QT21 and DFG-Core-Tec. SEQCLAS has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement n^o 694537. QT21 has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n^o 645452. It was partially supported by the Deutsche Forschungsgemeinschaft (DFG) under Contract No NE572/8-1. The work reflects only the authors' views and neither the European Commission nor the European Research Council Executive Agency nor the DFG are responsible for any use that may be made of the information it contains.

7. References

[1] J.-T. Peter, A. Guta, N. Rossenbach, M. Graça, and H. Ney, "The rwth aachen machine translation system for iwslt 2016," in *International Workshop on Spoken Language Translation. Seattle, WA, USA*, 2016.

[2] A. Guta, T. Alkhouli, J.-T. Peter, J. Wuebker, and H. Ney, "A Comparison between Count and Neural Network Models Based on Joint Translation and Reordering Sequences," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015.

[3] J.-T. Peter, F. Toutounchi, S. Peitz, P. Bahar, A. Guta, and H. Ney, "The rwth aachen german to english mt system for iwslt 2015," in *International Workshop on Spoken Language Translation*, Da Nang, Vietnam, Dec. 2015, pp. 15–22.

[4] J.-T. Peter, A. Guta, T. Alkhouli, P. Bahar, J. Rosendahl, N. Rossenbach, M. Graça, and H. Ney, "The rwth

aachen university english-german and german-english machine translation system for wmt 2017," in *Proceedings of the Second Conference on Machine Translation*, 2017, pp. 358–365.

[5] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016. [Online]. Available: http://aclweb.org/anthology/P/P16/P16-1162.pdf

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," May 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[8] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio, "Maxout networks," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, 2013, pp. 1319–1327. [Online]. Available: http://jmlr.org/proceedings/papers/v28/goodfellow13.html

[9] P. Bahar, T. Alkhouli, J.-T. Peter, C. J.-S. Brix, and H. Ney, "Empirical investigation of optimization algorithms in neural machine translation," *The Prague Bulletin of Mathematical Linguistics, The 20th Annual Conference of the European Association for Machine Translation*, vol. 108, no. 1, pp. 13–25, 2017.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[11] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Coverage-based neural machine translation," *CoRR*, vol. abs/1601.04811, 2016. [Online]. Available: http://arxiv.org/abs/1601.04811

[12] T. Cohn, C. D. V. Hoang, E. Vymolova, K. Yao, C. Dyer, and G. Haffari, "Incorporating structural alignment biases into an attentional neural translation model," *CoRR*, vol. abs/1601.01085, 2016. [Online]. Available: http://arxiv.org/abs/1601.01085

[13] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *CoRR*, vol. abs/1506.07503, 2015. [Online]. Available: http://arxiv.org/abs/1506.07503

[14] S. Feng, S. Liu, M. Li, and M. Zhou, "Implicit distortion and fertility models for attention-based encoder-decoder NMT model," *CoRR*, vol. abs/1601.03317, 2016. [Online]. Available: http://arxiv.org/abs/1601.03317

[15] B. van Merriënboer, D. Bahdanau, V. Dumoulin, D. Serdyuk, D. Warde-Farley, J. Chorowski, and Y. Bengio, "Blocks and fuel: Frameworks for deep learning," *CoRR*, vol. abs/1506.00619, 2015. [Online]. Available: http://arxiv.org/abs/1506.00619

[16] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[17] M. Junczys-Dowmunt, T. Dwojak, and R. Sennrich, "The AMU-UEDIN submission to the WMT16 news translation task: Attention-based NMT models as feature functions in phrase-based SMT," in *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, 2016, pp. 319–325. [Online]. Available: http://aclweb.org/anthology/W/W16/W16-2316.pdf

[18] 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, 2002, pp. 311–318.

[19] 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, 2006, pp. 223–231.

[20] W. Wang, J.-T. Peter, H. Rosendahl, and H. Ney, "Character: Translation edit rate on character level," in *ACL 2016 First Conference on Machine Translation*, Berlin, Germany, Aug. 2016.

[21] S. Jean, O. Firat, K. Cho, R. Memisevic, and Y. Bengio, "Montreal neural machine translation systems for wmt'15," in *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, 2015, pp. 134–140.

FBK's Multilingual Neural Machine Translation System for IWSLT 2017

Surafel M. Lakew[1,2], *Quintino F. Lotito*[2], *Marco Turchi*[1], *Matteo Negri*[1], *Marcello Federico*[1]

[1]Fondazione Bruno Kessler, Trento, Italy
[2]University of Trento, Trento, Italy
federico@fbk.eu

Abstract

Neural Machine Translation has been shown to enable inference and cross-lingual knowledge transfer across multiple language directions using a single multilingual model. Focusing on this multilingual translation scenario, this work summarizes FBK's participation in the IWSLT 2017 shared task. Our submissions rely on two multilingual systems trained on five languages (*English, Dutch, German, Italian, and Romanian*). The first one is a 20 language direction model, which handles all possible combinations of the five languages. The second multilingual system is trained only on 16 directions, leaving the others as zero-shot translation directions (*i.e* representing a more complex inference task on language pairs not seen at training time). More specifically, our zero-shot directions are Dutch↔German and Italian↔Romanian (resulting in four language combinations). Despite the small amount of parallel data used for training these systems, the resulting multilingual models are effective, even in comparison with models trained separately for every language pair (*i.e.* in more favorable conditions). We compare and show the results of the two multilingual models against a baseline single language pair systems. Particularly, we focus on the four zero-shot directions and show how a multilingual model trained with small data can provide reasonable results. Furthermore, we investigate how pivoting (*i.e* using a bridge/pivot language for inference in a source→pivot→target translations) using a multilingual model can be an alternative to enable zero-shot translation in a low resource setting.

1. Introduction

Recently, multilingual translation across different languages using a single model showed to perform in a comparable way with single language pair systems. In [1, 2], a multilingual model has been successfully trained using a standard Neural Machine Translation (NMT) architecture by applying a simple preprocessing step on the source side of the training data. It consists in prepending an artificial language token indicating the target language id at the beginning of each sentence. This information guides the system towards a specific target language both at training and inference time. This mechanism of guiding the multilingual model is referred to as *target-forcing* [2].

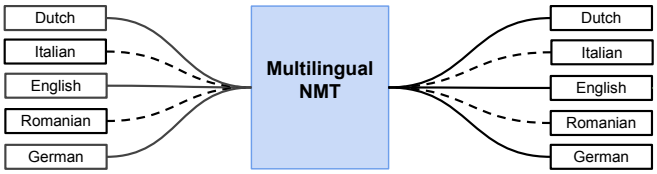


Figure 1: The multilingual system source→target association. A parallel data exists for all the 20 directions in the first multilingual model, where as zero-shot model the Dutch↔German and Italian↔Romanian pairs (dashed line) are excluded

In this work, we present our participation in two IWSLT2017[1] [3] shared tasks: *i)* a multilingual translation task in a small data condition for twenty language directions, and *ii)* a multilingual zero-shot task in a similar small data condition. For convenience, throughout the paper, we refer to the models trained for the two tasks respectively as *Multilingual* and *Zero-shot* models. We trained the two models separately, by sharing a common configuration. The only difference, at training time, is that we removed the four language directions involved in the zero-shot task. Figure 1, shows the twenty possible associations between the source and target pairs, avoiding (*source* = *target*) condition. We trained the models following the same preprocessing and training procedures described in [1]. Note that, due to its small size ($\approx 200K$ for each language pair), the training data set becomes even more sparse after preprocessing and dropping sentences above a certain length (which becomes necessary in order to facilitate and speed-up the training process).

For comparing the performance of the multilingual and zero-shot models, we trained 20 single language pair models. For a fair comparison, the preprocessing and training procedures are similar to the multilingual models. The same models are also used for the comparison against the pivoting method, in which *English* is fixed as the bridging language. In terms of evaluation results, the overall performance of the zero-shot model is satisfactory even if, unsurprisingly, lower than the multilingual model. The largest distance is observed in Romanian→Italian (-3.02 BLEU points), while the smallest difference is observed in the Dutch→German

[1]https://sites.google.com/site/iwsltevaluation2017/

direction (-1.63 BLEU points).

In the following sections of this paper, we begin by introducing the main concepts related to NMT (§2). Then, we review the related work in a multilingual (§3.1) and zero-shot (§3.2) translation domains. In Section 4, we describe the training details (§4.1), the dataset, the preprocessing procedures (§4.2), as well as the results of the single language pair (§4.3) and the multilingual (§4.4) models. For comparing the different approaches, we focus on the zero-shot directions in section (§4.5). Then, we give further analysis in Section 5 and conclude the work in Section 6.

2. Neural Machine Translation

NMT comprises an encoder, a decoder, and an attention-mechanism, which are all trained with maximum likelihood in an end-to-end fashion [4]. The encoder is a recurrent neural network (RNN) that encodes a source sentence into a sequence of hidden state vectors. The decoder is an RNN that uses the representation of the encoder to predict words in the target language [5] [6]. The *attention* mechanism is used to improve the translation by deciding which part of the source sentence can contribute mostly in the prediction process at each time step.

As shown in Figure 2, which simplifies the NMT architecture, first the encoder (green colored section) takes the source words left to right, maps them to vectors and feeds them into the RNN. When the <eos> (*i.e* end of sentence) symbol is seen, the final time step initializes the decoder RNN (blue colored). At each time step, the attention mechanism is applied over the encoder hidden states and combined with the current hidden state of the decoder to predict the next target word. Then, the prediction is fed back to the decoder RNN to predict the next word until the <eos> symbol is generated [7].

In order to build a multilingual model, in this work we used a standard encoder-decoder NMT architecture with a general attention mechanism that combines via dot product the decoder hidden state and a linear transformation of the encoder state [8]. Furthermore, we used four layers of RNN both on the encoder and decoder side.

3. Related Work

3.1. Multilingual NMT

Early works in multilingual NMT are characterized by the use of separate encoder, decoder, and an attention mechanism for every language direction [9] [10]. Firat et al. [11] introduced a way to share the attention mechanism in a many-to-many translation setting still keeping separate encoders and decoders for each source and target language. In a more closely related approach to the one, we utilized in our systems, [1] and [2] introduced a way to share not only the attention mechanism but also a single encoder-decoder. In both works, an artificial language token is prepended at a preprocessing stage to the source sentences in order to enable multilingual translation. In a rather different way, the approach in [2] appended a language-specific code to differentiate words from different languages. The word and subword level language-specific coding mechanism is proved to be expensive, by creating longer sentences that can deteriorate the performance of NMT [5]. In addition, they appended the artificial token as a prefix and postfix on the source side of the training and validation data. In [1], however, only one artificial token is prepended at the beginning of the source sentences. This single token, which specifies the target language proved to work in a comparable performance as specifying two (prefix and postfix) tokens. In this work, we follow the Johnson et al. [1] approach for prepending.

3.2. Zero-Shot Translation

Firat et al. [12], suggested a zero-resource translation by extending their approach in [11] with a shared attention mechanism and a separate encoder-decoder architecture for every language pair. They leverage a pre-trained multi-way multilingual model, and then fine tune it with synthetic parallel data generated by the model itself. Their approach, however, does not allow a zero-shot translation. Instead, they proposed a *many-to-one* translation setting and used the idea of generating a pseudo-parallel corpus [13] for fine-tuning purposes Moreover, also in this case, the need of separate encoders and decoders for every language pair significantly increases the model complexity. So far, though simple, the most effective approach proposed for zero-shot translation is the one based on *target-forcing* at preprocessing stage [1] [2]. The most attractive benefit of the *target-forcing* comes from the possibility to perform zero-shot translation with the same multilingual setting as in [1, 2].

However, recent experiments have shown that the mechanism fails to achieve reasonable zero-shot translation performance for low-resource languages [14], due to the fact that the target-forcing mechanism requires more examples at training time to effectively handle zero-shot at inference stage. This is particularly visible in case of zero-shot target language which appears only once in comparison with other source $\rightarrow$ target pairs. The promising results in [1] and [2] hence require further investigation to verify if their method can work in various language settings, particularly for low resourced and across distant languages.

As an alternative strategy, pivoting is a rather intuitive way to approach zero-shot translation, especially when it involves low-resourced languages. The idea is to translate from/into under-resourced languages (L_{source} and L_{target}) by leveraging data available for a high-resourced one (L_{pivot}) used as "bridge" between the two languages (*i.e.* $L_{source} \rightarrow L_{pivot} \rightarrow L_{target}$) [15]. However, results in the pivoting framework are strictly bounded to the performance of the two combined translation engines, and especially to that of the weaker one. In contrast, multilingual models that leverage knowledge acquired from data for different language combinations (similar to multi-task learn-

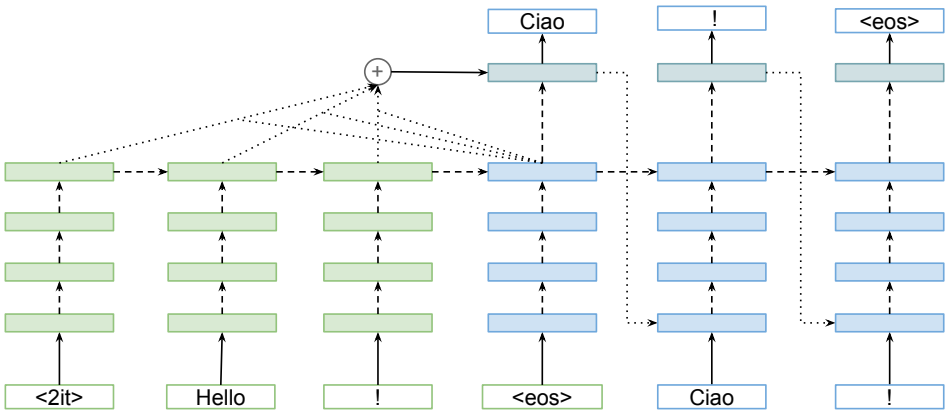


Figure 2: *NMT architecture with encoder-decoder and an attention mechanism, showing an example input "Hello !" translated to Italian "Ciao ! using a <2it> target-forcing mechanism". The first two layers of the encoder (green) are a bidirectional RNN with two additional forward layers. On the decoder side (blue), however, all the layers are forward. The attention mechanism is shown for the first time step of the prediction. Input feed is used to pass the context vector as additional input to the decoder.*

ing) can potentially compete or even outperform the pivoting ones. Taking the different approaches to perform zero-shot in consideration, in Section 4.5, we show the comparison between the zero-shot strategies (i.e direct source→target zero-shot translation and using a pivot language) employing the zero-shot model and the single language pair models.

4. Experiments

4.1. Training Details

For training the multilingual and the single language pair systems we used a standard encoder-decoder NMT architecture with attention mechanism [8][16]. The encoder and decoder sides of the network consist of four layers, where the first two layers of the encoder are bidirectional. As shown on the right side of Figure 2, at each time step an input-feeding mechanism is applied to pass the context vector as an additional input to the decoder by concatenating it with the embedding of the predicted word [8]. Table 1, shows the parameters used for training both the multilingual and single language pair systems.

For optimization, based on preliminary experiments and following best practices from previous work [1], we used Adam [17] with a learning rate of 0.001. Learning rate decay of 0.5 is applied if the perplexity does not decrease on the validation set or the number of epoch passes 8. For reducing perplexity and the network size, we also share the word and softmax embedding of the decoder as suggested by Press and Wolf [18]. To prevent overfitting [19], particularly for the training dataset in this low-resource setting, we applied a dropout of 0.3 on all layers [20]. At time of inference, a beam search of size 10 is utilized to balance decoding time and accuracy of the search. Where each decoding step takes a batch of 128 evaluation set. The experiments are carried out using the

Parameter	Value
RNN type	LSTM
RNN size	1024
embedding	512
encoder	bidirectional
encoder depth	4
decoder depth	4
beam size	10
batch size	128
optimizer	adam
dropout	0.3

Table 1: *Parameters used to train both single language pair and multilingual models.*

open source OpenNMT-py[2] toolkit [7].

With the aim to compare the performance of the multilingual models, we trained twenty single language pair models with the same amount of training data used by each direction of the multilingual models (see Table 2 for details). For every direction of the multilingual models and every single pair model we report case sensitive detokenized (*i.e* using the internal tokenization of the scorer) BLEU scores [21] computed using `mteval-v13a.pl`.

4.2. Dataset and Preprocessing

In the source-target pair of the five languages considered in this work, there are $\approx 200k$ parallel training sentences in each pair. As shown in Table 2, test2010 is used for evaluating the models, whereas test2017 is used for comparison purposes and as the official submission test set. For training both the multilingual and single language pair models, the same number of sentences are used.

[2]https://github.com/OpenNMT/OpenNMT-py

Direction	Training	test2010	test2017
English ↔ German	197,489	1,497	1,138
English ↔ Italian	221,688	1,501	1,147
English ↔ Dutch	231,669	1,726	1,181
English ↔ Romanian	211,508	1,633	1,129
German ↔ Italian	197,461	1,502	1,133
German ↔ Romanian	194,257	1,626	1,121
Dutch ↔ Italian	228,534	1,623	1,183
Dutch ↔ Romanian	199,762	1,637	1,123
German ↔ Dutch	209,169	1,729	1,174
Italian ↔ Romanian	209,668	1,605	1,127

Table 2: *Number of sentences used for training and evaluation in a source↔target combination. The German↔Dutch and Italian↔Romanian four language directions shown in the third row is removed from the training data of the zero-shot multilingual model.*

To prepare the data for training, we first prepare a tokenized version. Then, using a shared byte pair encoding (BPE) model, we segment the tokens into sub-word units [22]. The BPE model is trained on a joint source and target dataset covering all the language directions. For this operation we used $8,000$ BPE merging rules. A frequency threshold of 30 is used to apply the segmentation. For choosing the BPE segmentation rules, we follow the suggestion of Denkowski and Neubig [23] in such small data condition. When training the multilingual models, we add the *target-forcing* language token at the source side of each parallel data, both for training and validation sets [1]. Apart from the data set provided by the IWSLT17 shared task [24], for the multilingual small data condition no additional data are utilized, neither for the preprocessing stage nor for the experiments.

4.3. Single Language Pair Models

To compare the two evaluation tasks (multilingual and zero-shot model), we trained twenty single language pair models. As discussed in training details (4.1), these models are trained in a similar setting with the multilingual models. Table 3, summarizes the performance each of the twenty models on *test2017*. Except for the slight gain in the Romanian→Italian direction over the results of the multilingual model (see Table 4), the performance of the single language pair models (see Table 3), are poorer in the rest of the other 19 directions.

4.4. Multilingual Models

In this experiment, we present the multilingual 20 direction and zero-shot 16 direction models. Note: in case of the zero-shot model the training data for the German↔Dutch and Italian↔Romanian directions are dropped. As in the single language pair models, the rest of the training follows the steps described in Section 4.1. The results shown in Table 4, are the primary runs of the official submission for the multilingual and zero-shot small data condition tasks. The term of comparison between these two multilingual models is focused on the four zero-shot directions. As expected, the zero-shot model performed poorly than the multilingual model in all of the four directions.

Particularly, we see a larger gap of 3.02 for the Romanian → Italian, whereas the Italian → Romanian direction has a difference of 2.48 BLEU score. In case of German → Dutch and Dutch → German the gap closes to 1.99 and 1.63 respectively. For the other 16 non-zero-shot directions, the multilingual model performed slightly better than the zero-shot model. However, in case of Dutch → English and Italian→Dutch there exists a pattern where the zero-shot model performed better. In Table 5, we separately reported additional results for the multilingual small-data condition task evaluated using a model from an on-time submission. Except for the reporting purpose the results from Table 5, are not included in any of the comparisons made in this work.

4.5. Zero-shot Vs. Pivoting

In this analysis, we compare zero-shot translation mechanisms using the Zero-shot multilingual model and models trained on single language pair. Specifically, we compared three different results of a zero-shot translation on the IWSLT *tst2017*. The first is a direct zero-shot from a source → target language using the Zero-shot multilingual model. The other two results are acquired through a pivoting translation mechanism in a *two-step* translation. Hence, pivoting using single language pair models requires a source→pivot and a pivot→target model. However, this is not the case for the Zero-shot model which assumes to already have the pivot paired with the source and target languages. In both cases, we use English as a pivot language. Thus, for the Italian ↔ Romanian zero-shot directions we follow Italian ↔ English ↔ Romanian, whereas the German ↔ Dutch translation is done as German ↔ English ↔ Dutch *two-step* translations.

Approaches	De→Nl	Nl→De	It→Ro	Ro→It
Zero-shot	17.17	**16.96**	16.58	18.32
Zero-shot Pivot	**17.67**	16.84	**17.3**	**19.57**
Single Pair Pivot	15.3	14.9	15.22	17.2

Figure 3: A BLEU score comparison of German ↔ Dutch and Italian ↔ Romanian four language directions using three different zero-shot translation mechanisms. The first row is a direct zero-shot translation using the Zero-shot model, while the last two rows show the results of the pivoting mechanism.

The results in Table 3, shows better performance of the Zero-shot model using a pivoting mechanism (except the Nl→De direction). In a surprising way, the pivoting using two separate single language pair models for each translation direction perform worse than the direct zero-shot and the pivoting zero-shot using the multilingual model in row 1 and 2.

Single Pair	En−De	En−Nl	En−It	En−Ro	De−Nl	De−It	De−Ro	Nl−It	Nl−Ro	It−Ro
→	19.84	26.41	29.90	21.41	18.93	15.52	12.52	18.47	14.71	18.67
←	24.69	30	34.03	28.03	17.93	15.47	13.81	20.13	16.78	**21.71**

Table 3: BLEU score on IWSLT *tst2017* from twenty single language pair models that are trained separately. The bold highlighted Romanian→Italian direction is the only gain over the multilingual system.

Multilingual	En−De	En−Nl	En−It	En−Ro	De−Nl	De−It	De−Ro	Nl−It	Nl−Ro	It−Ro
→	20.88	26.72	29.6	21.95	19.16	16.84	14.62	19.33	16.54	19.06
←	25.62	29.79	34.24	28.93	18.59	16.88	15.87	20.27	18.92	21.34
Zero-shot										
→	20.67	26.11	28.86	21.54	17.17	16.28	13.93	**19.76**	15.88	16.58
←	25.22	**30.04**	34.16	28.52	16.96	16.13	15.47	20.00	17.72	18.32

Table 4: BLEU scores on the IWSLT *tst2017* using the multilingual model trained on 20 directions and the zero-shot model trained using the dataset of the 16 directions. Bold highlighted Nl→En and Nl→It are the only cases where the zero-shot model performed better than the multilinugal.

5. Discussion

The experiments in this work showed that a single multilingual system can perform better than independently trained single language pair systems. Hence, training a single system on the concatenation of all the language directions helps to maximize the parameter sharing in the common representation space. The consistent gain of the multilingual model in 19 directions except for the slight loss for the Romanian→Italian shows the potential behind multilingual approaches. Unlike the scenario in previous work [1], we showed the improvements in a low resource setting, without any additional data to tune the system. In case of the zero-shot model, we considered the non-zero-shot 16 directions for comparison with the bilingual models. In an equivalent way with the multilingual model, the zero-shot model has shown gains over the single language pair models.

Even though the zero-shot model showed a comparable performance with the multilingual model in the 16 non-zero-shot directions, there is a slight performance degradation in all but the Dutch→English direction. For instance, a 29.6 BLEU score for English→Italian of the multilingual model decreases to 28.86 BLEU with the zero-shot model. However, for the translation directions Source→English the maximum loss for the zero-shot model is 0.41 BLEU in the Romanian→English direction. As we expected initially, these results reflect a condition where the number of language pairs with English (on the encoder and decoder side) stayed the same in both multilingual models. Whereas the absence of the four zero-shot (source↔target) combinations influenced the translation performance of the Zero-shot model even for the language pairs seen at training time.

The pivoting experiments discussed in Section 3, is another way of showing the reasonable performance of the zero-shot model. The two-step inference (i.e source → pivot, and then pivot→target) for zero-shot translation provided a better performance in three directions out of four (see Table 3), in comparison with a direct zero-shot translation. We observed that using English (the only language that has a pair and better performance with all the zero-shot directions) as the bridge language played the major role for the gain. However, as discussed in Section 3, pivoting using two separate bilingual systems is found to be weaker (see the third row of Table 3) in leveraging the pivot language. This can be observed from the weaker bilingual systems in comparison with the zero-shot model. Particularly, both in the source→English, and as well in the English→target the bilingual model performance is poor in comparison with the zero-shot model, see Table 3 and 4.

Overall the reasonable performance of the zero-shot model shows the potential of a multilingual approach. In the subsequent comparisons using a pivoting method, it becomes clearer that in a multilingual setting it is possible to train a more robust model that can handle the noise from the output of the first step translation.

6. Conclusion

In this work, we showed how a multilingual system can deliver better performance over bilingual systems in twenty different directions. In addition, we explored the performance of a multilingual model for a zero-shot translation task in a direct source-to-target translation and using a pivot language in a two-step translation. The Zero-shot model proved to be an effective way of achieving a zero-shot translation for German ↔ Dutch and Italian ↔ Romanian directions, while showing a comparable performance in the non-zero-shot directions with the Multilingual model trained on the full training dataset. In addition to avoiding training several independent systems, multilingual model showed to be beneficial in such low-resource setting.

In future works, we plan to thoroughly investigate the

Multilingual	En−De	En−Nl	En−It	En−Ro	De−Nl	De−It	De−Ro	Nl−It	Nl−Ro	It−Ro
→	20.28	25.68	29.32	21.12	18.67	15.85	13.43	19.25	15.48	17.89
←	24.27	30.16	33.86	28	17.65	15.98	14.99	18.77	17.5	21.28

Table 5: BLEU scores for the twenty language directions evaluated using a multilingual model on *tst2017* (results are using a model from an on-time submission of the multilingual small data condition task).

behavior of the multilingual systems, seeing that the target-forcing mechanism plays the main role in redirecting the translation to the right target language, and susceptible to ambiguities in a low-resource setting. In addition, we plan to explore a better way to balance the training dataset for the different language directions. Particularly, for achieving a zero-shot translation we expect that finding the right language combinations, amount of dataset, and the number of languages require further investigation. Furthermore, a human evaluation on the outputs of the bilingual and the multilingual models would be interesting to assess the translation quality, in addition to confirming the evaluation scores, reported in this work.

7. Acknowledgements

This work has been partially supported by the EC-funded projects ModernMT (H2020 grant agreement no. 645487) and QT21 (H2020 grant agreement no. 645452). The Titan Xp used for this research was donated by the NVIDIA Corporation. This work was also supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1 and by a donation of Azure credits by Microsoft.

8. References

[1] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *arXiv preprint arXiv:1611.04558*, 2016.

[2] T.-L. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," *arXiv preprint arXiv:1611.04798*, 2016.

[3] M. Cettolo, M. Federico, L. Bentivogli, J. Niehues, S. Stüker, K. Sudoh, K. Yoshino, and C. Federmann, "Overview of the IWSLT 2017 Evaluation Campaign," in *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, 2017.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[5] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[7] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:1701.02810*, 2017.

[8] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[9] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation." in *ACL (1)*, 2015, pp. 1723–1732.

[10] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," *arXiv preprint arXiv:1511.06114*, 2015.

[11] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," *arXiv preprint arXiv:1601.01073*, 2016.

[12] O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Y. Vural, and K. Cho, "Zero-resource translation with multilingual neural machine translation," *arXiv preprint arXiv:1606.04164*, 2016.

[13] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[14] S. M. Lakew, A. D. G. Mattia, and F. Marcello, "Multilingual neural machine translation for low resource languages," in *CLiC-it 2017 – 4th Italian Conference on Computational Linguistics, to appear*, 2017.

[15] H. Wu and H. Wang, "Pivot language approach for phrase-based statistical machine translation," *Machine Translation*, vol. 21, no. 3, pp. 165–181, 2007.

[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[17] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] O. Press and L. Wolf, "Using the output embedding to improve language models," *arXiv preprint arXiv:1608.05859*, 2016.

[19] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[21] 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 on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[22] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[23] M. Denkowski and G. Neubig, "Stronger baselines for trustable results in neural machine translation," *arXiv preprint arXiv:1706.09733*, 2017.

[24] M. Cettolo, C. Girardi, and M. Federico, "Wit3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, vol. 261, 2012, p. 268.

KIT's Multilingual Neural Machine Translation systems for IWSLT 2017

†*Ngoc-Quan Pham,* †*Matthias Sperber,* *†*Elizabeth Salesky,* †*Thanh-Le Ha,* †*Jan Niehues,* *†*Alexander Waibel*

†Institute for Anthropomatics and Robotics
KIT - Karlsruhe Institute of Technology, Germany
firstname.lastname@kit.edu
* Language Technologies Institute
Carnegie Mellon University
firstname.lastname@cmu.edu

Abstract

In this paper, we present KIT's multilingual neural machine translation (NMT) systems for the IWSLT 2017 evaluation campaign machine translation (MT) and spoken language translation (SLT) tasks.

For our MT task submissions, we used our multi-task system, modified from a standard attentional neural machine translation framework, instead of building 20 individual NMT systems. We investigated different architectures as well as different data corpora in training such a multilingual system. We also suggested an effective adaptation scheme for multilingual systems which brings great improvements compared to monolingual systems.

For the SLT track, in addition to a monolingual neural translation system used to generate correct punctuations and true cases of the data prior to training our multilingual system, we introduced a noise model in order to make our system more robust. Results show that our novel modifications improved our systems considerably on all tasks.

1. Introduction

In recent works, attention-based neural networks has been considered the state-of-the-art approach for machine translation. More importantly, this framework can be efficiently adapted or customized to fit in a multilingual setting, so that one model can be trained to translate from and to multiple languages. In this evaluation campaign, we empirically explore different architectures which have been exploited in various previous works, in order to find the best combination for the multilingual setting.

Specifically, we break down the neural machine translation architecture into its main components: embedding layers, encoders, decoders, attention and output layers. Our analysis indicates which components can be shared to benefit from multilingual data. We also employed an adaptation strategy which is proved to be beneficial for multi-task learning. Our best systems are the ensembles of individual architectures.

2. Data Processing

The data is preprocessed prior to training and translation. Sentence longer than 50 words and aligned sentence pairs having a big difference in length are removed. Special dates, numbers and symbols are normalized. Smartcasing is applied as well. Afterwards, we apply byte pair encoding [1] to model the translation of rare words. We build corpora using 40K codes and 80K codes. Since we did not see a large difference in performance, all reported results use a byte pair encoding size of 40K.

2.1. Sentence alignment

While for the in-domain TED corpus, parallel data was provided for all directions, the out-of-domain EPPS data was only available from and to English. For all language direction that do not include English, no additional data was available. In order to generate this data, we used English as a pivot language, sentence-aligning the English sides of source-English and English-target data in order to extract source-target sentence pairs. In the two tracks that we participated in, the Small data consists of 4.2 million sentences while the Large data has 26 million for 20 language directions.

3. Multilingual NMT

In previous works, the encoder-decoder architecture with attention mechanism has been used in a multilingual setting [2, 3, 4, 5] with various architectural choices. While most authors decided to share the encoder and decoder weights between languages, the attention module remains controversial, as [2] negates the use of attention in the multi-task models, [4] uses explicit attention layers for each language pair, and in [3], one single model is shared between all pairs. In this work, we explore the possibilities of architectural sharing between encoder, decoder and attention layers across languages.

3.1. Architectures

3.1.1. Neural Machine Translation

Our base model is the encoder-decoder with attention mechanism [6, 7], in which both of the encoder and the decoder are Long-Short Term Memory networks [8]. The attention module is a two-layer feed-forward neural network that we found to work better than simple dot-product or bilinear models [7].

In the multilingual setting, we investigate the effectiveness of sharing different parts of the model. The break-down of the neural machine translation models is illustrated as in Figure 1

- **The embedding layers** project the discrete words into dense vectors. We also consider the output linear layer as an embedding one. These layers are language specific and their parameters cannot be shared across languages.
- **The encoders** encode the representation of the source sentences into a set of vectors S. We can share this component by using one single encoder to encode sentences regardless of the language.
- **The attention layer** reads the encoded source S and learns to focus on important information at every time step which is used for decoding. The attention layer depends on both the source and target languages.
- **The decoder** receives the context information from the attention layer and learns to generate target sentences.

3.1.2. Sharing Embeddings

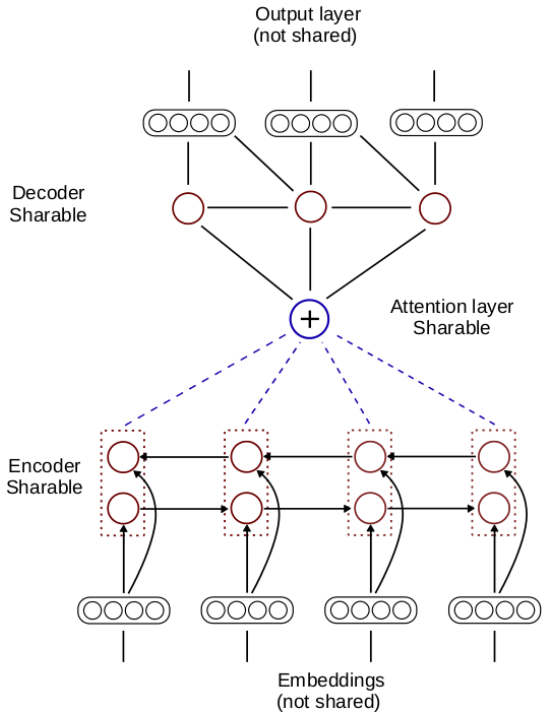
In this multiway, multilingual scenario, we have in total 5 languages on the both source and target sides. We want to ensure that the model has the same view of the embeddings on the source and target side, i.e a German word on the source data has the same embedding values as the same German word on the target sentences. Therefore, we construct one single projection matrix for each language, and use them according to the language of the sentences in the mini-batch.

For the output layer which computes the probabilities of the words, there are two different scenarios: if we use distinct vocabularies for each language, we then end up constructing five different output layers. Because of this architectural choice, each minibatch only contains sentences from one single language pair. In the second scenario, the probability distribution is computed of all words of all languages, then the output layer is not separated as in the first one. The two output layer scenarios are almost equivalent, but the former is much computationally faster than the latter, because the softmax layer required for each mini-batch is considerably smaller.

3.1.3. Sharing Encoder and Decoder

The encoder and decoder are fundamentally built by recurrent neural networks which learn the structural dependency

Figure 1: Neural Machine Translation architecture with shared components



of the words in sentences. For each language at the source and at the target, we assign a separate RNN encoder and a separate decoder. Similar to [4, 2], the specific language encoder weights are only updated when they are used during learning a particular mini-batch. In the sharing scenario, we just need to tie the weights and their gradients to the encoders and decoders.

3.1.4. Sharing Attention Mechanism

The attention layer consists of one feed-forward neural network which connects the hidden layers of the encoders with the hidden layer of the decoders. When being shared, the same network is used across twenty language pairs, while if attention is not shared, each language pair is assigned to one attention layer. Notably, sharing attention has been used in most multilingual setups [4, 3, 5] since the number of attention layers increases quadratically with respect to the number of language pairs, and it is believed that the shared attention layer can benefit better from multilingual sources [4].

4. Speech Translation

4.1. Punctuation Generation

Automatic speech recognition (ASR) systems typically do not generate punctuation marks or reliable casing. Using the raw output of these systems as input to MT causes a performance drop due to mismatched train and test condi-

tions. We used a monolingual NMT system to recase, insert proper punctuation, and add sentence boundaries to ASR output where necessary before translating [9].

To train, we created parallel data where the source sentence is the target sentence lowercased with all punctuation removed. Rare words were replaced with POS tags. The training data was randomly segmented so that segment boundaries and punctuation types were well-distributed throughout the corpus. For the English→German and German→English lecture data, segment boundaries are given, but for TED, they are not. At test time, we used a sliding window of length 10 to observe each word in multiple contexts as described in [9].

We used single-layer biLSTMs for the encoder and decoder, with 256 hidden units for the encoder/decoder/attention layers. Models were trained with Adam. We restarted the algorithm twice and applied early stopping.

4.2. Noised Training

Our speech translation model is applied to noisy and erroneous speech recognition outputs, despite never having been exposed to noisy data during the training process. The result is a harmful mismatch between training and test data that further aggravates the difficulty of having to transform malformed inputs in the first place. Sequence-to-sequence models have been observed to be especially sensitive to corrupted inputs due to erroneous ASR [10]. To improve robustness at test-time, we experiment with inducing a suitable form of noise during the training process. Specifically, we corrupt the source side of the parallel training data by randomly introducing substitution, insertion, and deletion errors. In this way, training data is made more similar to the testing condition, and the model potentially learns to handle noisy inputs at test-time in a more robust fashion.

The noise model is described in detail in [11]. Here, we used the simplified noise model sampling deletions only, at a noise rate of $\tau = 0.01$.

5. Results and Analysis

In this section, we present a summary of our experiments we have carried out for the IWSLT 2017 evaluation[12]. All the reported scores are case-sensitive BLEU scores.

5.1. Machine Translation tracks

5.1.1. Training details

System overview We built a neural machine translation framework which is customized with multiple encoders-decoders-attention for this multilingual task using PyTorch [1]. For the small data task, we use a small network configuration with word embedding and hidden layer size of 512 for all experimented architectures, except for the Share-All one which we found that layer size of 1024 is required to avoid underfitting. For the big data task, all of the models are trained with a larger config, with layer size of 1024. We applied Dropout between the vertical connection of the recurrent networks [13] with probability 0.5. We sampled minibatches containing sentences from only one language-pair so that the model can observe all sentences once every epoch. The parameters are updated using Adam optimizer [14] with the gradients clipped at 5. It is noteworthy that certain models with separated components can suffer from sparse updates since the unused components gradients are treated normally by Adam for the stat computation steps. We observed the training progress with the average *perplexity* on the validation sets, and used the models with the lowest perplexity to translate the test sets.

Adaptation We employed two different strategies of adaptation: in-domain (only applicable for large data task) and language-specific adaptation. Concretely, for in-domain adaptation after our models converge on the training set, we fine-tuned them further on the TED data as proposed in [15]. For language-specific adaptation, after we obtain the best performing model on the validation data, we continue training on each language pair. In the later section, the experimental results indicate that the language-specific adaptation is beneficial.

5.1.2. Main Results

For both tasks, we report the system performance on the test set with the tokenized BLEU (tBLEU) as well as the case-sensitive BLEU (cBLEU) scores. We explore three different architectures, based on our model design described in Section 3.1:

- Share-All : We tie all parameters of the encoders, decoders and attention layers across language pairs
- Share-RNN : The encoders and decoders parameters are shared, but explicit attention layers are separated for each language pair
- Separate-All : Encoders and decoders are language-specific, and the attention layers are separated.

Besides, we also employed the multilingual architecture from [5], here after referred to as 'Language-coded Multilingual'. The most similar architecture to Language-coded Multilingual is Share-All, where all components of the NMT system are shared. Language-coded Multilingual relies on preprocessing steps to share information while keeping the NMT architecture unchanged. In Language-coded Multilingual, however, the output is a big softmax layer, considering all distinct target words in all languages at the same time. Thus, Language-coded Multilingual is quite expensive to train and decode compared to our aforementioned architecture. We reported the result of Language-coded Multi-

[1] http://pytorch.org/

Table 1: Average BLEU scores on the test set for Small task

System	tokenized BLEU	case-sensitive BLEU
Separate-All	24.7	22.6
+ Lang-adapted	25.8	24
Share-RNN	26.0	24.2
+ Lang-adapted	26.3	**24.5**
Share-All	25.2	23.5
+ Lang-adapted	26.2	24.2
Language-coded Multilingual	25.6	23.8
Share-All + Lang-adapted + Average	25.7	23.8
Ensemble	27.4	**25.6**

lingual only on the Small task and without any adaptation scheme.

We also applied some strategies on top of Language-coded Multilingual systems to effectively improve the zero-shot translation. First we built two Language-coded Multilingual-based *Zero* systems, one used 18 language pairs excepts German↔Dutch, the other used 18 language pairs excepts Italian↔Romanian following the architecture suggested by [5]. Then we built other systems employing two strategies: *Target Dictionary Filtering* and *Language as a Word Feature* For greater details of those strategies, please refer to [16].

Small task The translation scores on the test data reflected that sharing the RNN encoders and decoders is clearly effective in multilingual setups. Both the architectures with shared RNNs outperformed their Separate-All counterpart, by 0.9 and 1.6 cBLEU. For the attention mechanism, we found out that sharing the attention reduces translation performance by 0.7 BLEU. Even with the shared recurrent networks, the context vectors from different languages are distinguishable, which is advantageous for the separate attention layers.

Also, as illustrated from table 2, language-specific adaptation helped us to improve the score, which is most clearly seen on the Separate-All model. The gain is also observed on the other two architectures, but not significant. This finding is in-line with [17], which shows that task-specific adaptation is necessary in for multi-task learning with neural encoder-decoders. Our final system to be submitted is the ensemble of three models after adaptation. Notably, the ensemble of Share-All and Share-RNN yields the same performance as the ensemble of all six models, showing that the adapted model dominates the others.

Meanwhile, the Language-coded Multilingual model performed best. Unsurprisingly, the scores from that system are similar to Share-All's. Due to its expensive training and different preprocessing pipeline, however, we did not attempt to employ adaptation and ensemble on that architecture.

The language-specific adaptation method is disadvantageous in that we have to store one model for each direction. Therefore, we tried to take all of the 20 models and average their parameters; interestingly, the averaged model performs better than the pre-adapted one.

Large task Moving over the large data set, we observe the same phenomenon as the small one, in which the Separate architecture fell behind the other two. Interestingly, Shared-All and Shared-RNN produce the same translation performance. One reason why may be that the shared-attention mechanism requires more data to become robust to language-specific mappings.

Even after adaptation (TED in-domain and language specific), the addition of the Europarl corpus only manages to improve the BLEU score by 0.4 for the best system. However, we reckon that further improvement can be achieved by increasing the model size and better parameter search, as was observed in the Small task.

Table 2: Average BLEU scores on the test set for Large task All systems are language-specifically adapted

System	tokenized BLEU	case-sensitive BLEU
Share-All	26.9	25.1
Share-RNN	26.9	25.1
Separate-All	25.1	23.4
Ensemble	27.8	**26.0**

Zero-shot task. We conducted the zero-shot translation for 4 directions asked by IWSLT'17 organizers: German↔Dutch and Italian↔Romanian. The results are shown in Table 3. We can see that *Language as a Word Feature* greatly improves our zero-shot translation systems.

5.2. Spoken Language Translation tracks

Our main translator for this task is the multilingual Share-All model trained with large data (which is also adapted on TED

System	DE→NL		NL→DE		IT→RO		RO→IT	
	dev2010	tst2010	dev2010	tst2010	dev2010	tst2010	dev2010	tst2010
Zero [5]	15.87	19.46	14.03	19.59	11.61	15.44	16.18	17.11
Zero Filtered Dict	15.79	19.48	13.96	19.59	11.52	15.45	16.21	17.20
Zero Lang Feature	**16.65**	**19.68**	**14.50**	**20.67**	**12.70**	**16.22**	**17.26**	**17.79**

Table 3: Effectiveness of proposed strategies on performance of zero-shot translation systems

data as well as language-specific data). However, there is a mismatch between the cleaned text data which is used for MT training and the noisy speech recognition output which can be disfluent, repetitive or lack of punctuations. Therefore, our effort to alleviate this problem is to apply a noisy model on the TED training data and then to adapt the translation models, as described in section 4.2.

The model is tuned with noisy data for ten more epochs (due to sampling, the data is actually slightly modified after each epoch) with learning rate of 0.0001. We conduct experiments on tst2013 sets on two directions: German→English and English→German. The experimental results are shown in Table 4 with case-sensitive BLEU scores. Using the noise models, we can improve the translation scores on both test sets by 0.5 and 0.3 respectively.

5.3. Other findings

In this section, we report the experimental findings that were not considered in the submission systems, including the configurations that we did not afford to finish.

Model capacity Initially we used layer size of 512 for all models for the Small task. With such capacity, the *Separate* was indeed the best model. However, when we scale the layer size to 1024, both of the two shared models improved drastically while the *Separate* model suffered from over-fitting despite of the high Dropout value. We express that, in order to fit the amount of training data that is quadratically larger than a single direction, the model capacity also needs to be scaled accordingly.

Such observation can also potentially explain the lackluster of the models trained on Large data. Such amount of data probably requires a larger/deeper model to utilise, which has been empirically experimented by [3]. However, as the larger model is much slower to train, we decided to keep the same configuration to have a reasonable training time.

Dropout Dropout is also one of the important factor to the model quality. We found out that on the small data, albeit the training data is 20 times larger than a single direction, a larger dropout value of 0.5 helps the model to regularise better than lower values such as 0.2, which is applicable for all architectures.

BPE size In the earlier experiments on the TED data, we tried out different BPE sizes of 40000 and 80000 merging operations, which were done over the concatenated data of all languages. We did not see any improvement of translation and proceeded to use 40000 in the later experiments.

Table 4: BLEU scores on tst2013 for Spoken Language Translation task

System	tst2013 EN-DE	tst2013 DE-EN
baseline	17.9	15.7
noise	18.4	16.0

6. Conclusions

In this paper, we described several innovative techniques that we applied to our multilingual neural machine translation systems, submitted to the IWSLT 2017 Evaluation Campaign. In order to use a single multilingual system instead of many individual systems, we tailored a standard neural translation framework to perform multi-task learning, where each language takes the role of one task. By doing so, we investigated different architectures with different shared components. Our experiments show that ensembling those systems improves the translation performance of the multilingual task further. In addition, a new training technique, the noise model, proved to be beneficial in the SLT task by making the translation system more robust on spoken data.

7. Acknowledgements

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 645452. The research by Thanh-Le Ha was supported by Ministry of Science, Research and the Arts Baden-Württemberg.

8. References

[1] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Association for Computational Linguistics (ACL 2016)*, Berlin, Germany, August 2016.

[2] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," in *International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2016.

[3] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. B. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *CoRR*, vol. abs/1611.04558, 2016. [Online]. Available: http://arxiv.org/abs/1611.04558

[4] O. Firat, K. Cho, and Y. Bengio, "Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism," *CoRR*, vol. abs/1601.01073, 2016. [Online]. Available: http://arxiv.org/abs/1601.01073

[5] T.-L. Ha, J. Niehues, and A. Waibel, "Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder," *Proceedings of the 13th International Workshop on Spoken Language Translation (IWSLT 2016)*, 2016.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[7] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," in *Proceedings of ACL-IJNLP 2015*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 11–19. [Online]. Available: http://www.aclweb.org/anthology/P15-1002

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[9] E. Cho, J. Niehues, and A. Waibel, "Nmt-based segmentation and punctuation insertion for real-time spoken language translation," *Proc. Interspeech 2017*, pp. 2645–2649, 2017.

[10] N. Ruiz, M. A. Di Gangi, N. Bertoldi, and M. Federico, "Assessing the Tolerance of Neural Machine Translation Systems Against Speech Recognition Errors," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Stockholm, Sweden, 2017, pp. 2635–2639.

[11] M. Sperber, J. Niehues, and A. Waibel, "Toward toward robust neural machine translation for noisy input sequences," in *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT 2017)*, Tokyo, Japan, 2017.

[12] M. Cettolo, M. Federico, L. Bentivogli, J. Niehues, S. Stüker, K. Sudoh, K. Yoshino, and C. Federmann, "Overview of the IWSLT 2017 Evaluation Campaign," in *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, 2017.

[13] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 285–290.

[14] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domains," in *Proceedings of the International Workshop on Spoken Language Translation*, 2015.

[16] T.-L. Ha, J. Niehues, and A. Waibel, "Effective Strategies in Zero-Shot Neural Machine Translation," *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT 2017)*, 2017.

[17] J. Niehues and E. Cho, "Exploiting linguistic resources for neural machine translation using multi-task learning," *CoRR*, vol. abs/1708.00993, 2017. [Online]. Available: http://arxiv.org/abs/1708.00993

Towards better translation performance on spoken language

Chao Bei, Hao Zong

Global Tone Communication Technology Co.,Ltd.

{beichao,zonghao}@gtcom.com.cn

Abstract

In this paper, we describe GTCOM's neural machine translation(NMT) systems for the International Workshop on Spoken Language Translation(IWSLT) 2017. We participated in the English-to-Chinese and Chinese-to-English tracks in the small data condition of the bilingual task and the zero-shot condition of the multilingual task. Our systems are based on the encoder-decoder architecture with attention mechanism. We build byte pair encoding (BPE) models in parallel data and back-translated monolingual training data provided in the small data condition. Other techniques we explored in our system include two deep architectures, layer nomalization, weight normalization and training models with annealing Adam, etc. The official scores of English-to-Chinese, Chinese-to-English are 28.13 and 21.35 on test set 2016 and 28.30 and 22.16 on test set 2017. The official scores on German-to-Dutch, Dutch-to-German, Italian-to-Romanian and Romanian-to-Italian are 19.59, 17.95, 18.62 and 20.39 respectively.

1. Introduction

This paper describes the submission of the Global Tone Communication Technology Co., Ltd. (GTCOM) for the first participation in IWSLT evaluation. We participated in the zero-shot condition in the multilingual task and the English-to-Chinese and Chinese-to-English tracks in the small data condition of the bilingual task. Our neural machine translation systems are developed as encoder-decoder architecture [1] with attention mechanism [2] and the experiment toolkit we used in the evaluation is Nematus [3].

The intuition of this participation is to verify whether the model architechture and techniques we applied in our generic system [1] with large training data is also effictive in spoken language domian with small training data. In bilingual task, since the training data is very small in both Chinese-to-English and English-to-Chinese directions, Chinese word segmentation, tokenization, binary pair encoder(BPE), different size of hidden layer, deep transition model and back-translation are involved in our experiments. In multilingual task, we uesd different pre-processing strategy and annealing Adam to enhance the translation performance.

This paper is arranged as follows. We firstly describe the task, including the data size and evaluation method. Then we introduce the techniques used in our system. After that, we present the experiments for the two task, including data preprocessing and model architecture. Finally, we analysis the experiment results and draw the conclusions.

[1]Our generic translation system covers 10 languages and is available at http://translateport.yeekit.com:4305/index.html

2. Task Description

The task focuses on bilingual and multilingual text translation in spoken language domain; the provided data is mainly collected form TED talks. We participated in Chinese-to-English and English-to-Chinese directions of the bilingual task, as well as zero-shot translation of the multilingual task.

2.1. Bilingual task

For the bilingual task, we focused on Chinese-to-English and English-to-Chinese directions of the small data condition, which only the in-domain training and development data is allowed to use. The detail information about the data is shown in Table 1. In addition, Chinese texts were evaluated at character level. Before evaluation, texts are splitted into Chinese characters, but sequences of non-Chinese characters are kept as they are.

2.2. Multilingual task

For multilingual task, we focused on zero-shot translation which using one model to translate any pair between English, Dutch, German, Italian and Romanian trained with the in-domain training and development data. In addition, training data synthesis from other pair and pivoting are allowed as contrastive conditions. But the directions, which included Dutch-to-German, German-to-Dutch, Italian-to-Romanian and Romanian-to-Italian, must be excluded from the training and development sets. The statistic of the parallel data is shown in Table 2.

3. Methology

This section introduces the techniques we used in our systems.

Table 1: *Number of sentences summary for in-domain training and development data for bilingual task.*

NMT direction	training data	development data 2013 2014 2015	monolingual data(target)
en-zh	231K	1,372 1,297 1,205	520K
zh-en	231K	1,372 1,297 1,205	234K

Table 2: *Number of sentences summary for in-domain training and development data for zero-shot multilingual task.*

language	de-en	de-it	de-ro	en-it	en-nl	en-ro	it-nl	nl-ro
training data	204K	203K	200K	230K	236K	219K	232K	205K
development set	1,138	1,133	1,121	1,147	1,181	1,129	1,183	1,123

3.1. Layer normalization and weight normalization

Layer normalization [4] is helpful to accelerate the convergence of model and improve the performance. [5] showed layer normalization is very effective in neural machine translation, especially with deep model. It is known that deep model for neural machine translation is difficult to converge. Weight normalization [6] is another method to accelerate the convergence and improve the performance, especially for recurrent models. Therefore, we used layer normalization in all the models and explore whether weight normalization play a further role on the models with layer normalization in neural machine translation.

3.2. Subword segmentation

To avoid unknow words, we used BPE-based splitting algorithm [7] to segment the word sequence to subword units sequence. This algorithm iteratively merges the most frquent pair of symbols into a single symbol. Therefore, the most frequent words in the corpus remain intact while the rare words are segmented into subunits. Joint BPE were used for the zero-shot condition, while we trained two separate BPE models for bilingual task due to different alphabet shared.

3.3. Back-translation

Monolingual in-domain data is also important for small training data condition. Monolingual data was back-translated with a shallow model trained with parallel data from target to source [8]. So we get translated source text and in-domain target text as synthetic parallel data. Then we mixed synthetic data and provided parallel data together to train our model.

3.4. Deep model

Deep model always gets better performance but is harder to converge. We use two architectures, stacked model [9] and deep transition model [10], which has been used in WMT 2017 by [5]. Even though the data size in [5] is larger than this task whose parallel data size is only 231K, deep model was still used to explore the adaptation on small data condition.

3.5. Annealing Adam

A strong baseline [11] gives a training trick, annealing Adam, which is significantly faster than SGD with annealing and obtains better performance. Adam [12] is an optimization algorithm, which applies momentum on a per-parameter basis and automatically adapts step size subject to a user-specified maximum. It speeds up the convergence and is a popular choice for researches. However, the models with Adam are slightly underperform compared to annealing SGD [13]. Thus, we halved learning rate after early stop and trained from the previous best model. We did this operation twice.

4. Experiment setup

4.1. Bilingual task

In this small data condition, we trained our systems using the in-domain data sets. Althrough, Chinese texts are evaluated at character level, we used Jieba [14], a Chinese word segmentation tool, to segment Chinese text in both parallel data and monolingual data. For English text, tokenizer and truecase in Moses [15] toolkit were applid. We applied BPE on both tokenized Chinese and English text. Before that, we calculated the word frequency on the training data and then get the number of words whose frequency is larger 10. Thus, the merge operation is calculate as
$N_{operation} = number\ of\ words(word\ frequency > 10)$
In our experiments , merge operation for English is set to 18000 and to 20000 for Chinese.

We used a 2-layer model trained with in-domain parallel data to translate the monolingual data as synthetic parallel data and mixed it with real parallel data. Translating Enlish monolingual data and Chinese monolingual data took about 4 days.

Our neural machine transition system is an encoder-decoder leverage GRU [16] cell in each layer with attention mechanism. The main model configuration is shown in Table 3. The mini-batches size is set to 64. The models were optimized using Adam with initial learning rate 0.0001 dur-

Table 3: *Model configuration for bilingual task.*

Type	value
English vocabulary size	19623
Chinese vocabulary size	25377
word embedding	512
hidden units	1024
embedding dropout	0.2
hidden dropout	0.2
source dropout	0.1
target dropout	0.1
layer normalization	True
maximum sentence length	100

ing training procedure, we also shuffed the training data after each epoch. For decoding we set the beam size to 10. In general, we trained 4-layer model and deep transition model with transition depth 4 for real parallel data and synthetic parallel data. Beside, the right-to-left model [17] with 4-layer architecture and deep transition architecture respectively were trained to rerank the n-best-list. It[17] showed a complementary target context will be seen at each time step and therefore the expected averaged probabilities will be more robust. In detail, We increase the size of the n-best-list to 50 for the reranking experiments.

4.2. Zero-shot condition in multilingual task

Different from bilingual task, in this zero-shot condition, the training data set consists of in-domain data from any pair between in English, Dutch, German, Italian and Romanian, except German-to-Dutch, Dutch-to-German, Italian-to-Romanian and Romanian-to-Italian data. We applied tokenizer and truecase script in Moses toolkit to preprocess all the corpora.

Zero-shot model aims to translate different langauage directions using the same model. Therefore, BPE segmentation is more useful than bilingual task. It can not only reduce the vocabulary size but also reduce the unknown words drastically. The merge operation of joint BPE model is 39500.

At the end of pre-processing, we add a label which consists of source language label and target language label at the start of each source sentence according [18]. Our processing for the language label is slightly different from [18]. And the model can translate from one specified source language to another specified target language learned from this label, although the model architecture didn't change.

Similar to bilingual task, the main model configuration is shown in Table 4. The mini-batch size is set to 80. And models were trained with Adam with initial learning rate 0.0001, the training data will be shuffed during each epoch. The Beam size in decoding is set to 10. We generally trained shallow model and deep transition model whose transition depth is 4 for all in-domain data. Beside, the right-to-left model with shallow model and deep transition architecture

Table 4: *Model configuration for multilingual task.*

Type	value
Source vocabulary size	40000
target vocabulary size	40000
word embedding	512
hidden units	1024
embedding dropout	0.2
hidden dropout	0.2
source dropout	0.1
target dropout	0.1
layer normalization	True
maximum sentence length	80

Table 5: *Results on Official Test Sets for binglingual task.*

direction	tst2016	tst2017
en-zh	28.13	28.30
zh-en	21.35	22.16

respectively were trained to rerank the n-best-list, which is the same in bilingual task.

5. Result and analysis

5.1. Results of bilingual task

Table 6 shows the case-insensitive BLEU score in development set of Chinese-to-English and Table 7 is for English-to-Chinese. We observed the improvement of 0-0.81 BLEU score from annealing Adam training trick and 0 to 0.88 BLEU score from training with a mix of parallel and synthetic data. But we find a fluctuation of -0.57 to 0.81 BLEU score from weight normalization especially in deep transition model. Weight normalization is not robust based on layer normalization in this condition. Ensembling of the independent models gives further imporvement by 0.97-1.28 BLEU score. Finally, our submitted system was reranked by right-to-left models with 50 n-best-list output of ensembling decoding of left-to-right models. This improved 0.3 to 0.55 BLEU score. Table 5 shows the official test results.

5.2. Results of multilingual task

Table 8 shows the case-insensitive BLEU score for development set of the zero-shot condition. It can be observed that adopting annealing Adam training algorithm also gets improvement of 0.28 to 0.36 BELU points, while weight normalization gets the worse performance. Ensemble decoding improves 1.93 BLEU points, compared shallow model. Then, we found in this condition, right-to-left reranking didn't improve the performance of model. We think that the zero-shot condition is a complex problem, which can translate from multilingual source language to multilingual target language. The model of right-to-left reranking may be hard

Table 6: *Case-insensitive BLEU score in development set of Chinese-to-English in small data condition. WN means weight normalization and SD means synthetic data.*

	tst2013	tst2014	tst2015	average
2 layers	20.32	18.07	21.48	20.03
+ annealing Adam	20.85	18.39	22.04	20.47
4 layers	20.89	17.91	21.87	20.33
+ annealing Adam	20.81	17.91	22.24	20.33
4 layers with WN	20.95	17.99	21.98	20.43
+ annealing Adam	21.24	18.1	21.81	20.48
4 layers with SD	21.05	18.4	21.94	20.49
+ annealing Adam	20.94	18.57	22.41	20.65
4 layers with SD and WN	21.34	18.72	22.5	20.91
+ annealing Adam	21.53	18.72	22.46	20.98
Deep transition	20.68	17.56	21.49	19.97
+ annealing Adam	21.11	17.66	21.64	20.28
Deep transition with WN	20.71	17.98	21.96	20.78
+ annealing Adam	21.40	18.33	22.30	20.80
Deep transition with SD	21.49	18.1	22.40	20.73
+ annealing Adam	21.75	18.83	22.77	21.16
Q Deep transition with SD and WN	21.31	18.78	22.07	20.78
+ annealing Adam	21.86	18.64	22.23	20.97
ensemble	22.83	19.72	23.73	22.13
+ r2l reranking	23.02	19.94	24.26	22.43

Table 7: *Case-insensitive BLEU score in development set of English-to-Chinese in small data condition. WN means weight normalization and SD means synthetic data.*

	tst2013	tst2014	tst2015	average
2 layers	23.71	21.03	26.80	23.83
+ annealing Adam	24.3	21.45	26.69	24.14
4 layers	23.94	21.63	27.34	24.30
+ annealing Adam	24.05	21.90	27.26	24.37
4 layers with WN	24.27	21.61	27.64	24.54
+ annealing Adam	24.46	21.8	27.42	24.54
4 layers with SD	24.43	21.89	28.00	24.74
+ annealing Adam	24.73	21.73	28.14	24.85
4 layers with SD and WN	24.39	21.47	27.61	24.47
+ annealing Adam	24.69	21.69	28.04	24.79
Deep transition	23.83	21.51	27.15	24.13
+ annealing Adam	23.75	21.37	27.06	24.03
Deep transition with WN	23.85	21.77	27.66	23.74
+ annealing Adam	24.21	21.92	27.43	24.49
Deep transition with SD	24.04	21.53	27.43	24.31
+ annealing Adam	24.47	22.1	27.98	24.82
Deep transition with SD and WN	23.7	21.7	26.5	23.74
+ annealing Adam	24.41	21.64	27.65	24.55
ensemble	25.86	23.21	29.41	26.13
+ r2l reranking	26.21	23.61	30.35	26.68

Table 8: *Case-insensitive BLEU score in development set of the zero-shot condition. WN means weight normalization.*

	en-de	en-nl	en-it	en-ro	de-en	de-it	de-ro	nl-en	nl-it
shallow model	28.29	32.22	29.67	27.56	34.43	20.60	19.47	38.01	22.42
+ annealing Adam	28.79	32.70	30.13	28.03	34.46	20.9	19.76	38.27	22.43
shallow model with WN	27.68	32.63	29.82	27.32	34.15	20.50	19.36	37.78	21.90
+ annealing Adam	27.79	32.56	30.15	27.72	34.42	20.82	19.81	38.03	22.05
deep transition	29.43	32.79	30.86	28.96	35.33	21.93	20.54	39.45	23.48
+ annealing Adam	29.9	32.85	31.56	28.78	35.72	22.18	20.91	39.79	23.67
deep transition with WN	28.85	33.19	30.98	28.37	34.83	22.07	20.28	38.96	23.06
ensemble	29.82	34.22	31.98	29.39	36.50	22.8	21.32	40.31	23.84
+ r2l reranking	29.60	32.70	31.58	28.77	35.76	22.48	21.45	39.50	24.22

	nl-ro	it-de	it-en	it-nl	ro-de	ro-en	ro-nl	average	
shallow model	20.79	20.75	34.22	22.1	22.05	35.81	23.15	27.28	
+ annealing Adam	21.31	20.85	34.61	22.22	22.26	36.06	23.34	27.56	
shallow model with WN	21.15	20.64	34.25	21.87	22.09	35.62	22.58	27.3	
+ annealing Adam	20.78	20.29	33.71	22.04	21.63	35.31	22.48	27.05	
deep transition	22.13	21.51	35.25	22.99	22.84	37.06	23.29	28.3	
+ annealing Adam	22.16	22.20	35.99	23.29	23.16	37.71	23.53	28.66	
deep transition with WN	21.83	21.55	35.13	22.86	22.73	37.09	23.63	28.17	
ensemble	22.93	22.56	36.15	23.93	23.35	38.05	24.49	29.21	
+ r2l reranking	22.74	24.41	35.74	23.76	23.68	37.47	24.61	28.99	

Table 9: *Results on Official Test Sets for multilingual task.*

direction	en-de	en-nl	en-it	en-ro	de-en	de-it	de-ro	de-nl	nl-en	nl-it
BLEU	23.08	29.08	32.84	23.89	28.04	18.56	16.23	19.59	32.78	21.21
Nist	5.86	6.81	7.22	5.91	6.85	5.36	4.69	5.57	7.42	5.72
Ter	60.63	51.46	47.63	58.81	51.41	63.43	69.04	61.26	47.34	60.83
direction	nl-ro	nl-de	it-de	it-en	it-nl	it-ro	ro-de	ro-en	ro-nl	ro-it
BLEU	18.11	17.95	18.09	37.84	21.80	18.62	17.95	31.79	20.02	20.39
Nist	4.97	5.06	5.09	8.10	5.78	5.03	5.06	5.59	5.59	5.57
Ter	66.55	67.02	67.28	41.05	60.09	65.53	67.02	41.22	67.81	61.11

to converge. In other words, we didn't get a good enough model of right-to-left reranking. Therefore, our submission was the results of ensemble decoding. And the result of the official test set is show in Table 9.

6. Summary

We presented our neural machine transition system for both bilingual task and multilingual task. The intution is mostly coming from the training of our generic translation system and the experiments shows the approaches we applied in our generic model is also effective in spoken langauge domain. Overall, the annealing Adam training algorithm and deep model always get a better performance, while weight normalization is not robust in this experiment. And right-to-left reranking for zero-shot model didn't help.

7. Acknowledgement

This work is supported by 2020 Cognitive Intelligence Research Institute[2] of Global Tone Communication Technology Co., Ltd.[3] We also want to thank Yiming Wang and Xinjie Li for their kindness help.

8. References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[3] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. V. M. Barone, J. Mokry, and M. Nadejde, "Nematus: a toolkit for neural machine translation," *CoRR*, vol. abs/1703.04357, 2017. [Online]. Available: http://arxiv.org/abs/1703.04357

[4] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[5] R. Sennrich, A. Birch, A. Currey, U. Germann, B. Haddow, K. Heafield, A. V. M. Barone, and P. Williams, "The university of edinburgh's neural mt systems for wmt17," *arXiv preprint arXiv:1708.00726*, 2017.

[6] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *CoRR*, vol. abs/1602.07868, 2016. [Online]. Available: http://arxiv.org/abs/1602.07868

[7] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[8] ——, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[9] J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu, "Deep recurrent models with fast-forward connections for neural machine translation," *CoRR*, vol. abs/1606.04199, 2016. [Online]. Available: http://arxiv.org/abs/1606.04199

[10] A. V. M. Barone, J. Helcl, R. Sennrich, B. Haddow, and A. Birch, "Deep architectures for neural machine translation," *CoRR*, vol. abs/1707.07631, 2017. [Online]. Available: http://arxiv.org/abs/1707.07631

[11] M. J. Denkowski and G. Neubig, "Stronger baselines for trustable results in neural machine translation," *CoRR*, vol. abs/1706.09733, 2017. [Online]. Available: http://arxiv.org/abs/1706.09733

[12] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[14] J. Sun, "jiebachinese word segmentation tool," 2012.

[15] 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 *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 2007, pp. 177–180.

[16] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014. [Online]. Available: http://arxiv.org/abs/1406.1078

[17] R. Sennrich, B. Haddow, and A. Birch, "Edinburgh neural machine translation systems for WMT 16," *CoRR*, vol. abs/1606.02891, 2016. [Online]. Available: http://arxiv.org/abs/1606.02891

[2] http://www.2020nlp.com/
[3] http://www.gtcom.com.cn/

[18] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. B. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *CoRR*, vol. abs/1611.04558, 2016. [Online]. Available: http://arxiv.org/abs/1611.04558

Kyoto University MT System Description for IWSLT 2017

Raj Dabre[1], Fabien Cromieres[2] and Sadao Kurohashi[1]

[1] Graduate School of Informatics, Kyoto University, Kyoto, Japan
[2] Japan Science and Technology Agency, Saitama, Japan

[1] dabre@nlp.ist.i.kyoto-u.ac.jp
[2] fabien@nlp.ist.i.kyoto-u.ac.jp
[1] kuro@i.kyoto-u.ac.jp

Abstract

We describe here our Machine Translation (MT) model and the results we obtained for the IWSLT 2017 Multilingual Shared Task. Motivated by Zero Shot NMT [1] we trained a Multilingual Neural Machine Translation by combining all the training data into one single collection by appending the tokens: "$< 2xx >$" (where xx is the language code of the target language) to the source sentences in order to indicate the target language they should be translated to. We observed that even in a low resource situation we were able to get translations whose quality surpass the quality of those obtained by Phrase Based Statistical Machine Translation by several BLEU points. The most surprising result we obtained was in the zero shot setting for Dutch-German and Italian-Romanian where we observed that despite using no parallel corpora between these language pairs, the NMT model was able to translate between these languages and the translations were either as good as or better (in terms of BLEU) than the non zero resource setting. We also verify that the NMT models that use feed forward layers and self attention instead of recurrent layers are extremely fast in terms of training which is useful in a NMT experimental setting.

1. Introduction

One of the most attractive features of neural machine translation (NMT) [2, 3, 4] is that it is possible to train an end to end system without the need to deal with word alignments, translation rules and complicated decoding algorithms, which are a characteristic of statistical machine translation (SMT) systems [5]. However, it is reported that NMT works better than SMT only when there is an abundance of parallel corpora. In the case of low resource domains, vanilla NMT is either worse than or comparable to SMT, due to overfitting on the small size of parallel corpora [6].

Although PBSMT is superior to NMT in low resource situations it leads to large models (phrase and reordering tables and language models) and thus is not an attractive approach, especially because it cannot lead to the development of models that are end to end. Recently, Google's multilingual system was made available to the public which was able to perform Zero Shot translation [1]. Although, it is possible to train a multilingual NMT model using a multi encoder and decoder setup [7], such a model contains a massive number of parameters and does not enable interaction between languages by means of shared encoders and decoders. Moreover, it is clear that the basic attention based encoder-decoder model is more than capable of accommodating multiple languages while keeping the number of parameters constant. Multilingual NMT (MLNMT) models are inherently more powerful than bilingual models especially when the target language for most pairs is common.

One major problem with MLNMT models is that they take a lot of time (ranging from several days to a few weeks) to train and thus it is very difficult to test out changes in approaches. This is because the original models are recurrent which need O(N) time for encoding followed by O(N) for decoding. Recently, models that use feed forward layers instead of recurrent layers [8] were proposed which are roughly an order of magnitude faster than their recurrent predecessors. Even without ensembling, they have also been shown to surpass ensembles of recurrent models by a significant amount. In a situation where time is limited and computing power (GPUs) such models (which we abbreviate as AIAYN[1]) can be a boon. It is important to note that although we refer to AIAYN as a feed forward model, the concept of self-attention is the central aspect of the overall architecture.

Since we had limited , we decided to work with the preprocessing based approach (prepending $< 2xx >$ tokens to source sentences) to train our multilingual AIAYN model. Internally, we compared our translations against those obtained using a PBSMT model and found them to be much superior.

2. Related Work

Our work can be viewed as an extension of Google's multilingual NMT work [1] with the main difference being that we used AIAYN [8]. Although, recurrent models that use multiple encoders and decoders [7] are an option, such models contain too many parameters and take even more time to

[1] The full form is Attention Is All You Need

train than bilingual models.

3. System Description

We trained MLNMT models for both the zero shot and non zero shot settings. For our models we followed the preprocessing approach [1]. For the non zero shot setting, for each language pair (20 pairs for the all pairs setting) we prepended the source language sentence with the tokens "$< 2xx >$" where xx could be any of the language codes for the languages under consideration. Following this we simply merged the corpora. Typically, it is a standard practice to oversample the smaller corpora but since all the corpora provided, were of the same size (in terms of number of lines), we skipped this step. For the zero shot setting we simply excluded the parallel corpora for the (bidirectional) language pairs German-Dutch and Italian-Romanian. While decoding, the input sentences are prepended with the token "$< 2xx >$" in order to force the model to translate to the target language whose language code is indicated by "xx". Apart from this we made no modifications to the NMT architecture or the decoding procedure.

We also created a multilingual PBSMT model by using a simple trick. We simply prepended every token in the source language sentences with the token "xx#" where xx indicates the target language. We also trained a joint language model on a concatenated corpora of the target side of all languages. This was enough to train a single multilingual SMT model. The working of such a model is as follows: Since each source word is marked by the "xx#" token, the phrase table contains unique entries for phrases for every language pair. During testing time, to translate from Dutch to Romanian, the input sentence will contain words marked with "ro#" and this sentence will match phrase pairs that are extracted from the Dutch-Romanian parallel corpus. Despite the non standard nature of this approach, it works well in practice. Since our focus was on NMT models we did not pursue this approach further, especially because it cannot be used to perform zero shot translation.

4. Experimental Settings

We worked on training a single NMT model for all the language directions in the multilingual task. The languages involved are German, English, Romanian, Italian and Dutch for which the language codes are de, en ,ro, it and nl" respectively. English, German and Dutch are Germanic languages whereas Romanian and Italian are Romance languages. Since they are all European languages and share cognates and grammatical structure, a multilingual model by means of parameter sharing can benefit greatly due to the language similarity.

For our experiments we used the parallel corpora provided to us by the organizers. For the non zero shot setting there are 20 parallel corpora for each language direction (5 languages and 4 targets per language leading to 20 pairs). For the zero shot setting (where the Italian-Romanian and German-Dutch corpora were to be excluded) we used only 16 out of the 20 parallel corpora. Kindly refer to the workshop overview paper for details on sizes. Apart from the official test set for this year's shared task we also evaluated our models using the "tst2010" test set that was provided to us along with the training data. Since the training, development and test sets are available in xml format we did preprocessing in the following order[2]:

1. Remove all XML tags so as to leave only raw sentences
2. Tokenize using the tokenizer in Moses[3].
3. Learn and apply a truecaser model[4] which deals with capitalization.
4. Optional 1: For the PBSMT models learn and apply a joint BPE model[5] to reduce data sparsity.

Following these steps we performed the following preprocessing steps to enable multilingual translations in a black box setting. For the PBSMT model we prepended every source language word with the token "xx#" corresponding to the target language. For the NMT models we prepended each source language sentence with the token "$< 2xx >$".

For training we used Moses[6] for the PBSMT model and Tensor2Tensor's implementation of AIAYN[7] for the NMT model.

For PBSMT the settings are:

- Subword vocabulary size of 32000 before appending the "xx#" tokens.
- A joint 7 gram KenLM model[8] [9] to account
- Default training settings for the phrase tables.
- Default settings for tuning using MIRA via MERT.

For NMT the settings are:

- Subword vocabulary size of 32000 which the subword tokenizer in the AIAYN implementation generates automatically.
- Embeddings and layer outputs of sizes 512 and the feed forward layer with a hidden later size of 2048.

[2]To generate the submission files we simply undid the preprocessing in the reverse direction

[3]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[4]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/recaser/truecase.perl

[5]https://github.com/rsennrich/subword-nmt

[6]https://github.com/moses-smt/mosesdecoder

[7]https://github.com/tensorflow/tensor2tensor/tree/master/tensor2tensor

[8]https://github.com/kpu/kenlm

L1/L2	de	en	it	nl	ro
de	-	26.45	17.54	19.64	16.27
en	23.25	-	30.79	28.80	24.66
it	19.10	34.73	-	22.32	20.60
nl	20.27	30.49	19.86	-	17.65
ro	17.94	29.58	21.89	20.24	-

Table 1: The official evaluation results for the multilingual NMT model task (non zero shot case).

L1/L2	de	en	it	nl	ro
de	-	27.08	17.67	**20.31**	16.08
en	23.63	-	30.99	30.18	24.49
it	19.20	35.28	-	22.76	**20.37**
nl	**19.68**	30.63	20.74	-	17.74
ro	18.40	30.23	**21.85**	20.47	-

Table 2: The official evaluation results for the multilingual NMT model task (zero shot case). The results for the zero shot pairs are marked in bold.

- Adam optimizer with a weight decay on the learning rate that increases for 16000 iterations and then decreases.
- Beam of size 4 with an alpha value of 0.6 for decoding the test sets.

We trained our models for 400000 iterations which is equivalent to roughly 10 epochs that required only 3-4 days on 5 GPUs. With 8 GPUs which is the default setting in the original AIAYN paper we can expect faster convergence. We did experience a slight amount of overfitting and could have eliminated it with dropout but will pursue such activities in the future. We also did not average the model checkpoints before decoding and instead only took the final model[9] for decoding. Decoding for all language pairs was done in parallel on multiple GPUs and took roughly an hour for all the test sets. The automatic evaluation measure we used was BLEU[10] [11] which we compute for the detokenized sentences.

5. Results

First we give the results of the official evaluation for the non zero shot and zero shot settings in Tables 1 and 2 respectively followed by the evaluations on the "tst2010" test set which was provided along with the training data in Table 3.

Since we are not aware of the BLEU scores for the runs submitted by the other participants we are unable to comment on how well our results are compared to others. However, we do have interesting observations regarding our zero shot results. Despite having no parallel corpora between the Italian-Romanian and German-Dutch language pairs, the zero shot NMT model performs almost as well for translations between these pairs. For German-Dutch the non zero shot model gives a BLEU of 19.64 whereas the zero shot model gives a BLEU of 20.31 which is a significant improvement. For the reverse direction though, the non zero shot model gave a BLEU of 20.27 against 19.68 BLEU for the the zero shot model. Although, there is a drop in translation quality it is not large. For the Italian-Romanian pair (both directions) the differences between the two settings is insignificant.

L1/L2	de	en	it	nl	ro
de	-	29.63 34.98	17.57 21.37	23.51 23.69	14.49 18.96
en	21.70 27.81	-	24.04 29.07	27.25 30.91	21.38 26.65
it	15.88 21.37	28.89 34.58	-	18.48 21.83	19.46 20.72
nl	21.57 24.45	34.79 38.86	18.84 23.02	-	15.99 20.68
ro	15.96 21.81	31.10 37.10	22.65 24.07	18.57 23.01	-

Table 3: The results for the "tst2010" set which we used as a test set for our local evaluations. Each cell contains 2 scores the one on the top is for the multilingual PBSMT system and the one on the bottom is for the multilingual NMT system.

Zero Shot NMT between a language pair is known to give relatively lower BLEU scores as compared to a non zero shot scenario and thus the outcomes above puzzled us initially. We decided to inspect the parallel corpora for any oddities. After some preliminary analysis we discovered that, although, the corpora are available in their bilingual form there are about 150,000 N-lingual sentences in the overall collection. For example, out of approximately 250,000 sentences for Italian-Romanian, 150,000 (60%) sentences contain translations to other languages. This means that even if the Italian-Romanian parallel corpus is excluded from the training set, there is an indirect parallel corpus of 150,000 sentences between the two languages. This also means that this setting is not truly zero shot because of the existence of the 150,000 multilingual sentences. It would be interesting to see what would happen in case all the bilingual corpora are disjoint [11].

Apart from this we also see that the zero shot models performed slightly better than the non zero shot models in a number of cases and we believe that since the non zero shot models had to work with a larger number of language pairs, the training process was no effective enough. It is possible to argue that using models with more parameters might be a good idea but we have already mentioned that our models actually overfit on the training data which means that it is better

[9] Such models overfit on the training data since they have a slightly lower BLEU on the development set than some of the past checkpoints.

[10] This is computed by the multi-bleu.pl script, which can be downloaded from the public implementation of Moses [10].

[11] In other words, these corpora come from different parts of the TED corpora with zero overlaps in their content.

consider approaches where we design better training schedules or work with better models that can incorporate multiple languages better than the kind of models we are currently using.

In Table 3 we can see how well the NMT system we trained is compared to the PBSMT system. In most cases the difference is over 4 BLEU points. The multilingual PBSMT system is simply a hack, as is the NMT system, in the sense that we only concatenated the corpora. However in the NMT system multiple languages share a common representation space which allow them to interact with each other and elevate the overall translation quality.

Although we do not mention it in the experimental section we did experiment with training a multilingual RNN model using Kyoto NMT[12] [12]. The model size was roughly the same but even after 2 weeks of training we were unable to obtain peak performance in terms of BLEU. Overall, we tried training models for about a month after which we gave up and moved over to AIAYN models and as a result were able to train high quality models within a matter of 3-4 days.

As we have mentioned our models are slightly overfitted on the training data and we also do not average various model checkpoints. We believe that the BLEU scores above can be further increased by a few points but since we were not aware of advanced techniques like model averaging and lacked the time and resources for trying out various model settings we were unable to train the best possible models. Note that we also do not do ensembling which is something that the authors of tensor2tensor do not implement and is particularly unnecessary since model averaging seems to mitigate the need for ensembling many models. We believe that in the future these AIAYN models can be exploited to their fullest extent and will replace the traditional RNN models.

6. Conclusions

We have described how we trained our zero and non zero shot multilingual NMT model for the IWSLT Multilingual MT tasks. We used the simple token based (appending "$< 2xx >$" to the source language sentence where xx is the target language) approach and observed that it is much superior to a PBSMT system. We observed that for the given corpora and settings the zero shot results are as good as the non zero shot results because of the existence of N-lingual sentences which constitute 60% of the bilingual corpora. We also verified that AIAYN models are extremely fast to train and yield models of high quality in a matter of days instead of weeks or months which the recurrent NMT models require.

7. Acknowledgements

We would like to thank the creators of tensor2tensor for making their code available since it allowed us to conduct several NMT experiments in a few days which would have required weeks, if not months, had we relied on recurrent NMT models. We would also like to thank the organizers and the anonymous reviewers for their efforts. We would also like to thank MEXT (Japan) since their scholarship is the source of funding for the first author.

[12] https://github.com/fabiencro/knmt

8. References

[1] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. B. Vigas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation." *CoRR*, vol. abs/1611.04558, 2016. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1611.html

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*. San Diego, USA: International Conference on Learning Representations, May 2015.

[3] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: http://www.aclweb.org/anthology/D14-1179

[4] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969173

[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: Association for Computational Linguistics, June 2007, pp. 177–180. [Online]. Available: http://www.aclweb.org/anthology/P/P07/P07-2045

[6] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4,*

2016, 2016, pp. 1568–1575. [Online]. Available: http://aclweb.org/anthology/D/D16/D16-1163.pdf

[7] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," in *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, 2016, pp. 866–875. [Online]. Available: http://aclweb.org/anthology/N/N16/N16-1101.pdf

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[9] K. Heafield, "Kenlm: Faster and smaller language model queries," in *WMT@EMNLP*, 2011.

[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*. The Association for Computer Linguistics, 2007.

[11] 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 on Association for Computational Linguistics*, ser. ACL '02. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 311–318. [Online]. Available: http://dx.doi.org/10.3115/1073083.1073135

[12] F. Cromierès, "Kyoto-nmt: a neural machine translation implementation in chainer," in *COLING*, 2016.

The 2017 KIT IWSLT Speech-to-Text Systems for English and German

Thai-Son Nguyen, Markus Müller, Matthias Sperber, Thomas Zenkel, Sebastian Stüker and Alex Waibel

Institute for Anthropomatics, Karlsruhe Institute of Technology
Karlsruhe, Germany
{first.lastname}@kit.edu

Abstract

This paper describes our German and English *Speech-to-Text* (STT) systems for the 2017 IWSLT evaluation campaign. The campaign focuses on the transcription of unsegmented lecture talks. Our setup includes systems using both the Janus and Kaldi frameworks. We combined the outputs using both ROVER [1] and confusion network combination (CNC) [2] to achieve a good overall performance. The individual subsystems are built by using different speaker-adaptive feature combination (e.g., lMEL with i-vector or bottleneck speaker vector), acoustic models (GMM or DNN) and speaker adaptation (MLLR or fMLLR). Decoding is performed in two stages, where the GMM and DNN systems are adapted on the combination of the first stage outputs using MLLR, and fMLLR.

The combination setup produces a final hypothesis that has a significantly lower WER than any of the individual subsystems. For the English lecture task, our best combination system has a WER of 8.3% on the tst2015 development set while our other combinations gained 25.7% WER for German lecture tasks.

1. Introduction

For many years now, the *International Workshop on Spoken Language Translation* (IWSLT) offers a comprehensive evaluation campaign on spoken language translation. The evaluation is organized in different evaluation tracks covering automatic speech recognition (ASR), machine translation (MT), and the full-fledged combination of the two of them into speech translation systems (SLT). Different from previous years, this year's installment mostly consists of real-life lectures e.g., real university lectures or talks at real symposia.

The goal of the ASR track is the automatic transcription of fully unsegmented lectures. The quality of the resulting transcriptions is measured in word error rate (WER).

This system paper describes our English and German ASR setups with which we participated in the lecture ASR tracks of the 2017 IWSLT evaluation campaign. Similar to previous years' evaluation [3], we used the Janus Recognition Toolkit (JRTk) [4] which features the IBIS single-pass decoder [5] to build several complementary subsystems and combined them with an additional system developed with the Kaldi toolkit [6]. Our Janus-based systems employ different speaker-adaptive features, acoustic models or speaker adaptation techniques. While the Kaldi-based system applies the same adaptation techniques but employs sequence training and big n-gram language models for rescoring.

The rest of this paper is structured as follows. Section 2 describes the data that our system was trained and tested on. This is followed by Section 3 which provides a description of the acoustic front-ends used in our system and Section 7 which describes our segmentation setup. An overview of the techniques used to build our acoustic models is given in Section 5. We describe the language model used for this evaluation in Section 6. Our decoding strategy and results are then presented in Sections 8 and 9. We conclude the paper with Section 10.

2. Data Resources

2.1. Training Data

Table 1 and Table 2 show the data sources we used for the acoustic model training of our systems. This year we included 80 hours of broadcast news which results in a total of 483 hours for the English systems. For the German systems, we used the same training data as last year.

Source	# Amount
Quaero from 2010 to 2012	200 hours
Broadcast news [7]	80 hours
TED-LIUM v2 [8] excluding disallowed talks	203 hours
Total	483 hours

Table 1: *English acoustic modeling data.*

2.2. Test Data

For this year's evaluation campaign, the evaluation test set "tst2017" as well as the development test sets "tst2015", "tst2013" and "dev2017" were provided for the English and German lecture tasks. All development test sets featured a pre-segmentation provided by the IWSLT organizers. For the

Source	# Amount
Quaero from 2009 to 2012	180 hours
Broadcast news	24 hours
Baden-Württemberg parliament	160 hours
Total	364 hours

Table 2: *German acoustic modeling data.*

evaluation test set, automatic segmentation was required.

3. Feature Extraction

Our systems are built using several different front-ends as previously described in [3] including 40-dimensional log scale mel filterbank (lMEL), 20-dimensional mel frequency ceptral coefficient (MFCC), 20-dimensional minimum variance distortionless response (MVDR) and 14-dimensional tonal (T) features. These features can be augmented with i-vectors (Section 3.2) or bottleneck speaker vectors (Section 3.3) to be directly used for acoustic modeling or fed into deep bottleneck networks (Section 3.1) for extracting bottleneck features. The extracted bottleneck features are then transformed using feature-space maximum likelihood linear regression (fMLLR) and augmented with i-vectors to build speaker-adaptive features (Section 3.4). Our detailed feature extraction pipeline is explained in [9].

3.1. Bottleneck Features

We employed the deep bottleneck architecture described by [10], which consists of a stacked denoising auto-encoder of 4-5 layers each containing 1600-2000 units, followed by a 42 unit bottleneck, a hidden layer and the classification layer. The stacked auto-encoder is first pre-trained layer-wise [11], then the whole network is fine-tuned to discriminate target phoneme states. For the extraction of bottleneck features (BN), the layers after the bottleneck were removed and the output activations of the bottleneck layer were used as BN.

3.2. I-vectors

To extract i-vectors, a full universal background model (UBM) with 2048 mixtures was trained on the training dataset using 20 Mel-frequency cepstral coefficients with delta and delta-delta features appended. The total variability matrices were estimated for extracting 100 dimensional i-vectors. We tuned the size of the i-vectors in a series of preliminary experiments for optimal recognition performance. The UBM model training and i-vector extraction was performed by using the sre08 module from the Kaldi toolkit [6]. I-vectors as well as tonal features were always used in combination with other features.

3.3. Bottleneck Speaker Vectors

In addition to i-vectors, we also used Bottleneck Speaker Vectors (BSVs) [12]. While they serve the same purpose, they are entirely neural network based. We used the same setup as for our hybrid systems, but trained the network to recognize different speakers instead of phonemes using a one-hot encoding of the speaker identities. To extract the BSVs, we used a bottleneck layer as second last layer of the speaker classification network and discarded all layers after this layer after training. For obtaining the final speaker vector, we averaged the output activation of this hidden layer on a per speaker basis.

3.4. Speaker Adaptive Features

To build speaker-adaptive features (SAF) for GMM systems, we first train deep bottleneck network from 11 stacked frames of regular features and i-vectors. The extracted BN features are then spliced for 11 consecutive frames and transformed using Linear Discriminate Analysis (LDA) which are known to make inputs more accurately modeled by GMMs.

The speaker-adaptive features for DNN systems are obtained after transforming BN features using fMLLR transformation and then augmented with i-vectors. The process of fMLLR estimation was performed as traditional approach. During the training, we used the adaptation data of the same speaker and the reference transcriptions to do the alignment, while the same GMMs were used as first-pass systems to generate transcriptions in the testing.

4. Phoneme and Dictionary

For English, we used the CMU dictionary[1]. This is the same phoneme set as the one used in last year's systems. It consists of 45 phonemes and allophones. We used 7 noise tags and one silence tag. Missing pronunciations were created using the FESTIVAL [13] Text-to-Speech Engine.

Our German system uses an initial dictionary based on the Verbmobil Phoneset [14]. Missing pronunciations are generated using both MaryTTS [15] and FESTIVAL [13].

5. Acoustic Modeling

5.1. HMM CD-Phone

All GMM and hybrid models classify context-dependent quinphones with three states per phoneme and a left-to-right HMM topology without skip states. The English acoustic models use 8,156 distributions and codebooks derived from decision-tree based clustering of the states of all possible quinphones. The German acoustic models use either 10k or 18k context-dependent states.

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

5.2. GMM Models

The GMM models are trained by using incremental splitting of Gaussians training (MAS) [16], followed by optimal feature space training (OFS) which is a variant of *semi-tied covariance* (STC) [17] training using a single global transformation matrix. The model is then refined by one iteration of Viterbi training.

For the evaluation, we trained one GMM system using SAF features with MFCC front-ends for the English lecture task.

5.3. Hybrid Models

All the DNN models also share the same architecture which has 5-6 hidden layers with 2000 units per layer. The input of the DNNs are 11 stacked frames of 42-dimensional transformed bottleneck features or 40-dimensional lMEL, with or without combining i-vectors and tonal features. We used the sigmoid activation function for the hidden layers and softmax for the output layer. DNN systems were trained using the cross-entropy loss function to predict context-dependent states. The same training method is applied for all DNNs which includes pre-training with denoising auto-encoders and followed by fine-tuning with back-propagation. We used an exponential schedule to update the learning rate during the neural network training.

This year, we built two DNNs using SA features with different front-ends for the English TED task.

The German setup for the lectures task consists of 4 DNN systems based on different combinations of input features as shown in Table 6. SAF features were used as well.

6. Language Models

6.1. Vocabulary and Kneser-Ney Models

For language model training and vocabulary selection, we used the subtitles of TED talks, or translations thereof, and text data from various sources (see Tables 3 and 4). Text cleaning included tokenization, lowercasing, number normalization, and removal of punctuation. Language model training was performed by building separate language models for all (sub-)corpora using the SRILM toolkit [18] with modified Kneser-Ney smoothing. These were then linearly interpolated, with interpolation weights tuned using held-out data from the TED corpus. For German, we split compounds similarly as in [19].

For the vocabulary selection, we followed an approach proposed by Venkataraman et al. [20]. We built unigram language models using Witten-Bell smoothing from all text sources, and determined unigram probabilities that maximized the likelihood of a held-out TED data set. As our vocabulary, we then used the top 150k words for English, and 300k words for German.

Text corpus	**# Words**
TED	3.6m
Fisher	10.4m
Switchboard	1.4m
TEDLIUM dataselection	155m
News + News-commentary + -crawl	4,478m
Commoncrawl	185m
GIGA	2323m

Table 3: *English language modeling data.*

Text corpus	**# Words**
TED	2,685k
News+Newscrawl	1,500M
Euro Language Newspaper	95,783k
Common Crawl	51,156k
Europarl	49,008k
ECI	14,582k
MultiUN	6,964k
German Political Speeches	5,695k
Callhome	159k
HUB5	20k

Table 4: *German language modeling data after cleaning and compound splitting*.

6.2. Feed-forward Neural Language Model

During decoding the probabilities of a feedforward neural network language model were linearly interpolated with the baseline language model. Due to performance considerations, the most recent 40k queries for this language model were cached and we constrained the output vocabulary to the 20k most frequent words which appeared in the text corpora. We used 200 dimensional word embeddings trained with the Skip-gram model [21]. Three words were considered as the context, while the rest of the network consisted of three hidden layers followed by a softmax output layer. The training text consisted of 30M words and was selected based on the text sources listed in Table 4.

7. Automatic Segmentation

In this evaluation, the test set for the ASR track was provided without manual sentence segmentation, thus automatic segmentation of the target data was mandatory. We utilized an approach to automatic segmentation of audio data that is SVM based. This kind of segmentation is using speech and non-speech models, using the framework introduced in [22]. The pre-processing makes use of an LDA transformation on DBNF feature vectors after frame stacking to effectively incorporate temporal information. The SVM classifier is trained with the help of LIBSVM [23]. A 2-phased post-processing is applied for final segment generation.

We generated the segmentations for both English and German using this SVM based segmentation. The parameters for the SVM segmenter were chosen on a per language basis after preliminary experiments.

8. Systems and Combination

Table 5 shows our systems built for the English submission. In the first-pass, we used a GMM and two DNN systems with the acoustic models and 4-gram language model described in Section 5 and Section 6. Their decoded lattices are sent to a consensus decoding system (CNC) to produce combined hypotheses and confidence scores for the adaptation in the second-pass. The GMM system is fully adapted as transitional approach using both feature space adaptation (fMLLR) and model adaptation (MLLR). The DNN systems are adapted by training the DNN acoustic models one more epoch on the adaptation data of each speaker. The adaptation data is obtained by performing alignment of the CNC decoded results with the speaker audio and filtering out the frames with the confidence scores higher than 0.7. All these systems were built using Janus Recognition Toolkit (JRTK) [14].

Beside that we also used the Kaldi toolkit [6] to build a new system with similar feature adaptation techniques. The same train database is used for acoustic modeling and we use the trained 4-gram language model for rescoring.

Our final submission for the English lecture task consists of a ROVER of the Kaldi based system and the adapted systems. The results of the single and adapted systems as well the combined system are presented in Table 5.

9. Results

For the English task, we gained significant improvements over building speaker adaptive features, DNN model adaptation and CNC combination. On the test set "tst2015", we archived 8.3% WERs.

System	tst2015
GMM(SAF-MFCC)	11.6
DNN(SAF-lMEL)	10.2
DNN(SAF-MFCC)	11.2
CNC	9.4
GMM(SAF-MFCC) adapted	9.3
DNN(SAF-lMEL) adapted	8.8
DNN(SAF-MFCC) adapted	9.3
Kaldi 4-gram LM rescored	9.3
ROVER	8.3

Table 5: *Results for English talk task on 'tst2015' development set.*

In addition to our experiments on these two English tracks, we also participated in the German lecture task. The results on the "dev2017" test set are shown in Table 6.

System	dev2017
18k DNN(BSV BN-lMEL+T) NNLM	26.7
18k DNN(Mod-M2+lMEL+T)	27.1
10k DNN(SAF-BN-M2+T) NNLM	25.2
10k DNN(SAF-BN-lMEL+T) NNLM	25.7
CNC	25.7

Table 6: *Results for German lecture task on 'dev2017' development set.*

10. Conclusion

In this paper we presented our English and German LVCSR systems, with which we participated in the 2017 IWSLT evaluation. All systems make use of neural network based front-ends, HMM/GMM and HMM/DNN based acoustics models. The decoding set-up of all languages makes extensive use of system combination of single systems obtained by combining different feature extraction front-ends and acoustic models.

11. References

[1] 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, Dec. 1997, pp. 347–354.

[2] 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.

[3] Markus Müller, Thai-Son Nguyen, Matthias Sperber, Kevin Kilgour, Sebastian Stüker, and Alex Waibel, "The 2015 KIT IWSLT Speech-to-Text Systems for English and German," in *International Workshop on Spoken Language Translation (IWSLT)*, Dec. 2015.

[4] M. Woszczyna, N. Aoki-Waibel, F. D. Buø, N. Coccaro, K. Horiguchi, T. Kemp, A. Lavie, A. McNair, T. Polzin, I. Rogina, C. Rose, T. Schultz, B. Suhm, M. Tomita, and A. Waibel, "Janus 93: Towards spontaneous speech translation," in *International Conference on Acoustics, Speech, and Signal Processing 1994*, Adelaide, Australia, 1994.

[5] 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*. IEEE, 2001, pp. 214–217.

[6] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[7] D. Graff, "The 1996 broadcast news speech and language-model corpus," in *Proceedings of the 1997 DARPA Speech Recognition Workshop*, 1996.

[8] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks," in *Proc. of LREC*, 2014, pp. 3935–3939.

[9] Thai Son Nguyen, Kevin Kilgour, Matthias Sperber, and Alex Waibel, "Improved speaker adaption by combining i-vector and fmllr with deep bottleneck networks," SPECOM 2017. Springer, Cham, 2017.

[10] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE*, 2013.

[11] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *The 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[12] H. Huang and K. C. Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for dnn-based speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4610–4613.

[13] A. Black, P. Taylor, R. Caley, and R. Clark, "The festival speech synthesis system," 1998.

[14] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, and M. Westphal, "The karlsruhe-verbmobil speech recognition engine," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 83–86.

[15] M. Schröder and J. Trouvain, "The german text-to-speech synthesis system mary: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, no. 4, pp. 365–377, 2003.

[16] 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.

[17] 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.

[18] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh International Conference on Spoken Language Processing*, 2002.

[19] Kevin Kilgour, Christian Mohr, Michael Heck, Quoc Bao Nguyen, Van Huy Nguyen, Evgeniy Shin, Igor Tseyzer, Jonas Gehring, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel, "The 2013 KIT IWSLT Speech-to-Text Systems for German and English," in *International Workshop on Spoken Language Translation (IWSLT)*, Dec. 2013.

[20] A. Venkataraman and W. Wang, "Techniques for effective vocabulary selection," in *Proceedings of the 8th European Conference on Speech Communication and Technology*, 2003, pp. 245–248.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[22] M. Heck, C. Mohr, S. Stüker, M. Müller, K. Kilgour, J. Gehring, Q. Nguyen, V. Nguyen, and A. Waibel, "Segmentation of telephone speech based on speech and non-speech models," in *Speech and Computer*, ser. Lecture Notes in Computer Science, M. Železný, I. Habernal, and A. Ronzhin, Eds. Springer International Publishing, 2013, vol. 8113, pp. 286–293.

[23] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

TECHNICAL PAPERS

Neural Machine Translation Training in a Multi-Domain Scenario

*Hassan Sajjad, Nadir Durrani, Fahim Dalvi, *Yonatan Belinkov, Stephan Vogel*

Qatar Computing Research Institute – HBKU
*MIT Computer Science and Artificial Intelligence Laboratory
{hsajjad,ndurrani,faimaduddin,svogel}@qf.org.qa, *belinkov@mit.edu

Abstract

In this paper, we explore alternative ways to train a neural machine translation system in a multi-domain scenario. We investigate data concatenation (with fine tuning), model stacking (multi-level fine tuning), data selection and multi-model ensemble. Our findings show that the best translation quality can be achieved by building an initial system on a concatenation of available out-of-domain data and then fine-tuning it on in-domain data. Model stacking works best when training begins with the furthest out-of-domain data and the model is incrementally fine-tuned with the next furthest domain and so on. Data selection did not give the best results, but can be considered as a decent compromise between training time and translation quality. A weighted ensemble of different individual models performed better than data selection. It is beneficial in a scenario when there is no time for fine-tuning an already trained model.

1. Introduction

Neural machine translation (NMT) systems are sensitive to the data they are trained on. The available parallel corpora come from various genres and have different stylistic variations and semantic ambiguities. While such data is often beneficial for a general purpose machine translation system, a problem arises when building systems for specific domains such as lectures [1, 2], patents [3] or medical text [4], where either the in-domain bilingual text does not exist or is available in small quantities.

Domain adaptation aims to preserve the identity of the in-domain data while exploiting the out-of-domain data in favor of the in-domain data and avoiding possible drift towards out-of-domain jargon and style. The most commonly used approach to train a domain-specific neural MT system is to fine-tune an existing model (trained on generic data) with the new domain [5, 6, 7, 8] or to add domain-aware tags in building a concatenated system [9]. [10] proposed a gradual fine-tuning method that starts training with complete in- and out-of-domain data and gradually reduces the out-of-domain data for next epochs. Other approaches that have been recently proposed for domain adaptation of neural machine translation are instance weighting [11, 12] and data selection [13].

In this paper we explore NMT in a multi-domain scenario. Considering a small in-domain corpus and a number of out-of-domain corpora, we target questions like:

- What are the different ways to combine multiple domains during a training process?
- What is the best strategy to build an optimal in-domain system?
- Which training strategy results in a robust system?
- Which strategy should be used to build a decent in-domain system given limited time?

To answer these, we try the following approaches: **i) data concatenation:** train a system by concatenating all the available in-domain and out-of-domain data; **ii) model stacking:** build NMT in an online fashion starting from the most distant domain, fine-tune on the closer domain and finish by fine-tuning the model on the in-domain data; **iii) data selection:** select a certain percentage of the available out-of-domain corpora that is closest to the in-domain data and use it for training the system; **iv) multi-model ensemble:** separately train models for each available domain and combine them during decoding using balanced or weighted averaging. We experiment with Arabic-English and German-English language pairs. Our results demonstrate the following findings:

- A concatenated system fine-tuned on the in-domain data achieves the most optimal in-domain system.
- Model stacking works best when starting from the furthest domain, fine-tuning on closer domains and then finally fine-tuning on the in-domain data.
- A concatenated system on all available data results in the most robust system.
- Data selection gives a decent trade-off between translation quality and training time.
- Weighted ensemble is helpful when several individual models have been already trained and there is no time for retraining/fine-tuning.

The paper is organized as follows: Section 2 describes the adaptation approaches explored in this work. We present experimental design in Section 3. Section 4 summarizes the results and Section 5 concludes.

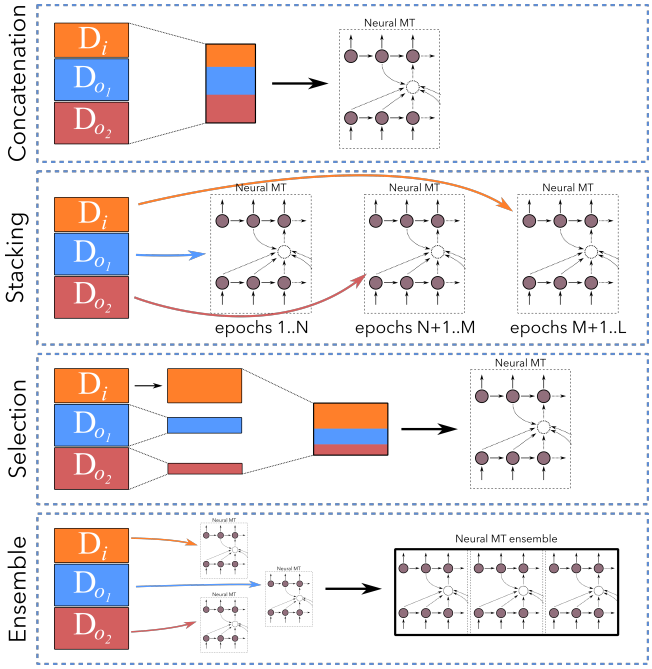


Figure 1: Multi-domain training approaches

2. Approaches

Consider an in-domain data D_i and a set of out-of-domain data $D_o = D_{o_1}, D_{o_2}, ..D_{o_n}$. We explore several methods to benefit from the available data with an aim to optimize translation quality on the in-domain data. Specifically, we try data concatenation, model stacking, data selection and ensemble. Figure 1 presents them graphically. In the following, we describe each approach briefly.

2.1. Concatenation

A naïve yet commonly used method when training both statistical [14][1] and neural machine translation systems [15] is to simply concatenate all the bilingual parallel data before training the system. During training an in-domain validation set is used to guide the training loss. The resulting system has an advantage of seeing a mix of all available data at every time interval, and is thus robust to handle heterogeneous test data.

2.2. Fine Tuning and Model Stacking

Neural machine translation follows an online training strategy. It sees only a small portion of the data in every training step and estimates the value of network parameters based on that portion. Previous work has exploited this strategy in the context of domain adaptation. [5] trained an initial model on an out-of-domain data and later extended the training on in-domain data. In this way the final model parameters are tuned towards the in-domain data. The approach is referred as *fine-tuning* later on.

Since in this work we deal with several domains, we propose a stacking method that uses multi-level fine-tuning to train a system. Figure 1 (second row) shows the complete procedure: first, the model is trained on the out-of-domain data D_{o_1} for N epochs; training is resumed from $N+1$-th epoch to the M-th epoch but using the next available out-of-domain data D_{o_2}; repeat the process till all of the available out-of-domain corpora have been used; in the last step, resume training on the in-domain data D_i for a few epochs. The resulting model has seen all of the available data as in the case of the data concatenation approach. However, here the system learns from the data domain by domain. We call this technique *model stacking*.

The model stacking and fine-tuning approaches have the advantage of seeing the in-domain data in the end of training, thus making the system parameters more optimized for the in-domain data. They also provide flexibility in extending an existing model to any new domain without having to retrain the complete system again on the available corpora.

2.3. Data Selection

Building a model, whether concatenated or stacked, on all the available data is computationally expensive. An alternative approach is *data selection*, where we select a part of the out-of-domain data which is close to the in-domain data for training. The intuition here is two fold: i) the out-of-domain data is huge and takes a lot of time to train on, and ii) not all parts of the out-of-domain data are beneficial for the in-domain data. Training only on a selected part of the out-of-domain data reduces the training time significantly while at the same time creating a model closer to the in-domain.

In this work, we use the modified Moore-Lewis [16] for data selection. It trains in- and out-of-domain n-gram models and then ranks sequences in the out-of-domain data based on cross-entropy difference. The out-of-domain sentences below a certain threshold are selected for training. Since we are dealing with several out-of-domain corpora, we apply data selection separately on each of them and build a concatenated system using in-domain plus selected out-of-domain data as shown in Figure 1. Data selection significantly reduces data size thus improving training time for NMT. However, finding the optimal threshold to filter data is a cumbersome process. Data selection using joint neural networks has been explored in [17]. We explore data selection as an alternative to the above mentioned techniques.

2.4. Multi-domain Ensemble

Out-of-domain data is generally available in larger quantity. Training a concatenated system whenever a new in-domain becomes available is expensive in terms of both time and computation. An alternative to fine-tuning the system with new in-domain is to do ensemble of the new model with the

[1] State-of-the-art baselines are trained on plain concatenation of the data with MT feature functions (such as Language Model) skewed towards in-domain data, through interpolation.

existing model. The ensemble approach brings the flexibility to use them during decoding without a need of retraining and fine-tuning.

Consider N models that we would like to use to generate translations. For each decoding step, we use the scores over the vocabulary from each of these N models and combine them by averaging. We then use these averaged scores to choose the output word(s) for each hypothesis in our beam. The intuition is to combine the knowledge of the N models to generate a translation. We refer to this approach as *balanced ensemble* later on. Since here we deal with several different domains, averaging scores of all the models equally may not result in optimum performance. We explore a variation of balanced ensemble called *weighted ensemble* that performs a weighted average of these scores, where the weights can be pre-defined or learned on a development set.

Balanced ensemble using several models of a single training run saved at different iterations has shown to improve performance by 1-2 BLEU points [15]. Here our goal is not to improve the best system but to benefit from individual models built using several domains during a single decoding process. We experiment with both balanced and weighted ensemble under the multi-domain condition only.[2]

3. Experimental Design

3.1. Data

We experiment with Arabic-English and German-English language pairs using the WIT[3] TED corpus [20] made available for IWSLT 2016 as our in-domain data. For Arabic-English, we take the UN corpus [21] and the OPUS corpus [22] as out-of-domain corpora. For German-English, we use the Europarl (EP), and the Common Crawl (CC) corpora made available for the 1^{st} Conference on Statistical Machine Translation[3] as out-of-domain corpus. We tokenize Arabic, German and English using the default *Moses* tokenizer. We did not do morphological segmentation of Arabic. Instead we apply sub-word based segmentation [23] that implicitly segment as part of the compression process.[4] Table 1 shows the data statistics after running the Moses tokenizer.

We use a concatenation of dev2010 and tst2010 sets for validation during training. Test sets tst2011 and tst2012 served as development sets to find the best model for fine-tuning and tst2013 and tst2014 are used for evaluation. We use BLEU [26] to measure performance.

3.2. System Settings

We use the Nematus tool [27] to train a 2-layered LSTM encoder-decoder with attention [28]. We use the default settings: embedding layer size: 512, hidden layer size: 1000. We limit the vocabulary to 50k words using BPE [23] with 50,000 operations.

Arabic-English			
Corpus	Sentences	Tok$_{ar}$	Tok$_{en}$
TED	229k	3.7M	4.7M
UN	18.3M	433M	494M
OPUS	22.4M	139M	195M
German-English			
Corpus	Sentences	Tok$_{de}$	Tok$_{en}$
TED	209K	4M	4.2M
EP	1.9M	51M	53M
CC	2.3M	55M	59M

Table 1: Statistics of the Arabic-English and German-English training corpora in terms of Sentences and Tokens. EP = Europarl, CC = Common Crawl, UN = United Nations.

4. Results

In this section, we empirically compare several approaches to combine in- and out-of-domain data to train an NMT system. Figure 2 and Figure 3 show the learning curve on development sets using various approaches mentioned in this work. We will go through them individually later in this section.

4.1. Individual Systems

We trained systems on each domain individually (for 10 epochs)[5] and chose the best model using the development set. We tested every model on the in-domain testsets. Table 2 shows the results. On Arabic-English, the system trained on the out-of-domain data OPUS performed the best. This is due to the large size of the corpus and its spoken nature which makes it close to TED in style and genre. However, despite the large size of UN, the system trained using UN performed poorly. The reason is the difference in genre of UN from the TED corpus where the former consists of United Nations proceedings and the latter is based on talks.

For German-English, the systems built using out-of-domain corpora performed better than the in-domain corpus. The CC corpus appeared to be very close to the TED domain. The system trained on it performed even better than the in-domain system by an average of 2 BLEU points.

4.2. Concatenation and Fine-tuning

Next we evaluated how the models performed when trained on concatenated data. We mainly tried two variations: i) concatenating all the available data (*ALL*) ii) combine only the available out-of-domain data (*OD*) and later fine-tune the

[2]Weighted fusion of Neural Networks trained on different domains has been explored in [18] for phrase-based SMT. Weighted training for Neural Network Models has been proposed in [19].

[3]http://www.statmt.org/wmt16/translation-task.html

[4][24] showed that using BPE performs comparable to morphological tokenization [25] in Arabic-English machine translation.

[5]For German-English, we ran the models until they converged because the training data is much smaller compared to Arabic-English direction

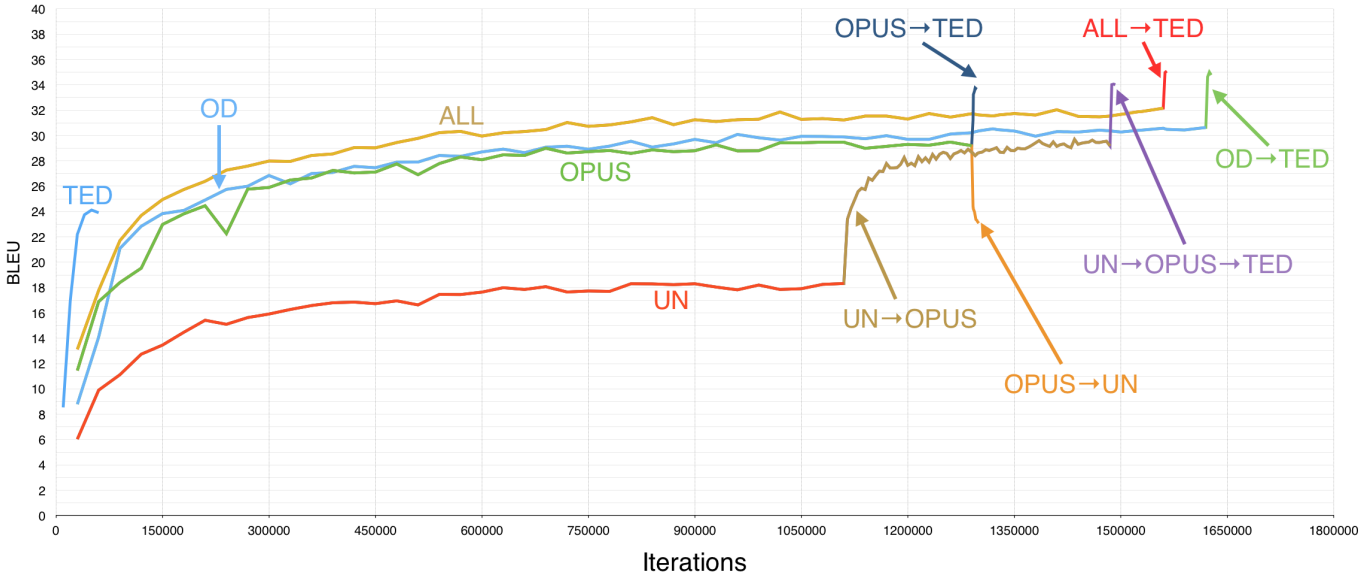


Figure 2: Arabic-English system development life line evaluated on development set tst-11 and tst-12. Here, `ALL` refers to `UN+OPUS+TED`, and `OD` refers to `UN+OPUS`

Arabic-English			
	TED	UN	OPUS
tst13	23.6	22.4	**32.2**
tst14	20.5	17.8	**27.3**
avg.	22.1	20.1	**29.7**
German-English			
	TED	CC	EP
tst13	29.5	**29.8**	29.1
tst14	23.3	**25.7**	25.1
avg.	26.4	**27.7**	27.1

Table 2: Individual domain models evaluated on TED testsets

Arabic-English				
	TED	ALL	OD→TED	ALL→TED
tst13	23.6	36.1	37.9	38.0
tst14	20.5	30.2	32.1	32.2
avg.	22.1	33.2	35.0	35.1
German-English				
	TED	ALL	OD→TED	ALL→TED
tst13	29.5	35.7	38.1	38.1
tst14	23.3	30.8	32.8	32.9
avg.	28.0	33.3	35.4	35.5

Table 3: Comparing results of systems built on a concatenation of the data. OD represents a concatenation of the out-of-domain corpora and ALL represents a concatenation of OD and the in-domain data. → sign means fine-tuning

model on the in-domain data. Table 3 shows the results. The fine-tuned system outperformed a full concatenated system by 1.8 and 2.1 average BLEU points in Arabic-English and German-English systems respectively.

Looking at the development life line of these systems (Figures 2, 3), since *ALL* has seen all of the data, it is better than *OD* till the point *OD* is fine-tuned on the in-domain corpus. Interestingly, at that point *ALL* and *OD*→TED have seen the same amount of data but the parameters of the latter model are fine-tuned towards the in-domain data. This gives it average improvements of up to 2 BLEU points over *ALL*.

The *ALL* system does not give any explicit weight to any domain [6] during training. In order to revive the in-domain data, we fine-tuned it on the in-domain data. We achieved comparable results to that of the OD→TED model which means that one can adapt an already trained model on all the available data to a specific domain by fine tuning it on the domain of interest. This can be helpful in cases where in-domain data is not known beforehand.

4.3. Model Stacking

Previously we concatenated all out-of-domain data and fine-tuned it with the in-domain TED corpus. In this approach, we picked one out-of-domain corpus at a time, trained a model and fine-tuned it with the other available domain. We repeated this process till all out-of-domain data had been used. In the last step, we fine-tuned the model on the in-domain data. Since we have a number of out-of-domain corpora available, we experimented with using them in different per-

[6] other than the data size itself

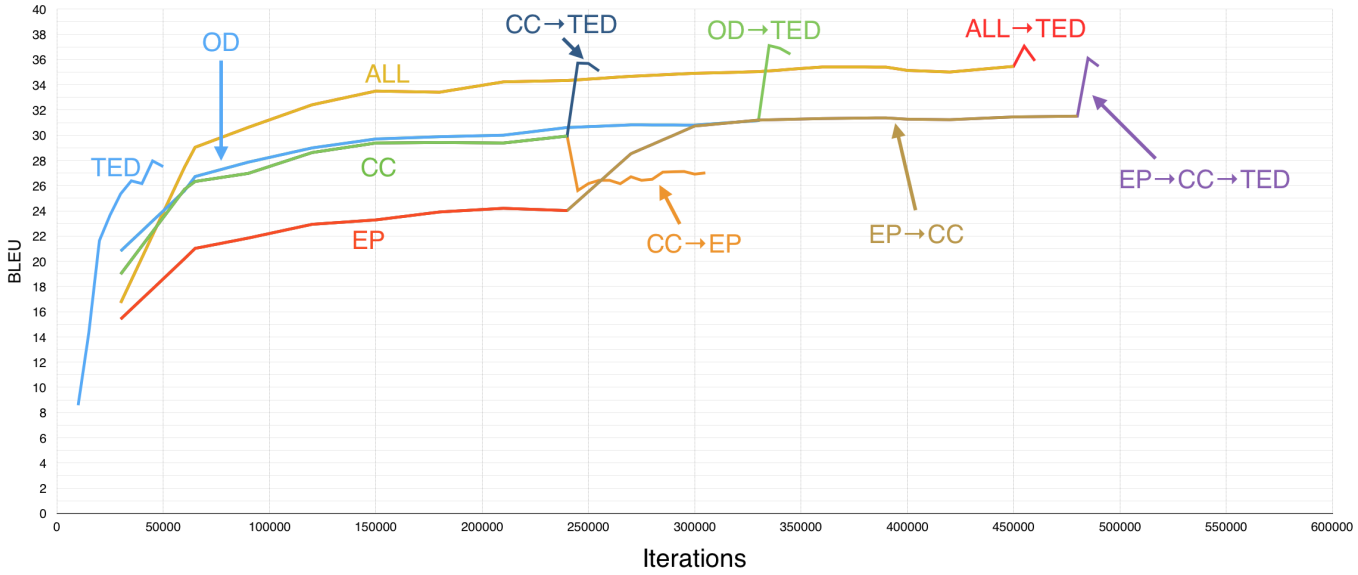


Figure 3: German-English system development life line evaluated on development set tst-11 and tst-12. Here, `ALL` refers to `EP+CC+TED`, and `OD` refers to `EP+CC`

mutations for training and analyzed their effect on the development sets. Figure 2 and Figure 3 show the results. It is interesting to see that the order of stacking has a significant effect on achieving a high quality system. The best combination for the Arabic-English language pair started with the UN data, fine-tuned on OPUS and then fine-tuned on TED. When we started with OPUS and fine-tuned the model on UN, the results dropped drastically as shown in Figure 2 (see OPUS→UN). The model started forgetting the previously used data and focused on the newly provided data which is very distant from the in-domain data. We saw similar trends in the case of German-English language pair where CC→EP dropped the performance drastically. We did not fine-tune CC→EP and OPUS→UN on TED since there was no better model to fine-tune than to completely ignore the second corpus i.e. UN and EP for Arabic and German respectively and fine-tune OPUS and CC on TED. The results of OPUS→TED and CC→TED are shown in Figures.

Comparing the OPUS→TED system with the UN→OPUS→TED system, the result of OPUS→TED are lowered by 0.62 BLEU points from the UN→OPUS→TED system. Similarly, we saw a drop of 0.4 BLEU points for German-English language pair when we did not use EP and directly fine-tuned CC on TED. There are two ways to look at these results, considering quality vs. time: i) by using UN and EP in model stacking, the model learned to remember only those parts of the data that are beneficial for achieving better translation quality on the in-domain development sets. Thus using them as part of the training pipeline is helpful for building a better system. ii) training on UN and EP is expensive. Dropping them from the pipeline significantly reduced the training time and resulted in a loss of 0.62 and 0.4 BLEU points only.

To summarize, model stacking performs best when it starts from the domain furthest from the in-domain data. In the following, we compare it with the data concatenation approach.

4.4. Stacking versus Concatenation

We compared model stacking with different forms of concatenation. In terms of data usage, all models are exposed to identical data. Table 4 shows the results. The best systems are achieved using a concatenation of all of the out-of-domain data for initial model training and then fine-tuning the trained model on the in-domain data. The concatenated system *ALL* performed the lowest among all.

ALL learned a generic model from all the available data without giving explicit weight to any particular domain whereas model stacking resulted in a specialized system for the in-domain data. In order to confirm the generalization ability of *ALL* vs. model stacking, we tested them on a new domain, News. *ALL* performed 4 BLEU points better than model stacking in translating the news NIST MT04 testset. This concludes that a concatenation system is not an optimum solution for one particular domain but is robust enough to perform well in new testing conditions.

4.5. Data Selection

Since training on large out-of-domain data is time inefficient, we selected a small portion of out-of-domain data that is closer to the in-domain data. For Arabic-English, we selected 3% and 5% from the UN and OPUS data respectively which constitutes roughly 2M sentences. For German-English, we

	Arabic-English		
	ALL	OD→TED	UN→OPUS→TED
tst13	36.1	37.9	36.8
tst14	30.2	32.1	31.2
avg.	33.2	35.0	34.0
	German-English		
	ALL	OD→TED	EP→CC→TED
tst13	35.7	38.1	36.8
tst14	30.8	32.8	31.7
avg.	33.3	35.4	34.3

Table 4: Stacking versus concatenation

	Arabic-English		German-English	
	ALL	Selected	ALL	Selected
tst13	36.1	32.7	35.7	34.1
tst14	30.2	27.8	30.8	29.9
avg.	33.2	30.3	33.3	32.0

Table 5: Results of systems trained on a concatenation of selected data and on a concatenation of all available data

selected 20% from a concatenation of EP and CC, which roughly constitutes 1M training sentences.[7]

We concatenated the selected data and the in-domain data to train an NMT system. Table 5 presents the results. The selected system is worse than the *ALL* system. This is in contrary to the results mentioned in the literature on phrase-based machine translation where data selection on UN improves translation quality [29]. This shows that NMT is not as sensitive as phrase-based to the presence of the out-of-domain data.

Data selection comes with a cost of reduced translation quality. However, the selected system is better than all individual systems shown in Table 2. Each of these out-of-domain systems take more time to train than a selected system. For example, compared to individual UN system, the selected system took approximately 1/10th of the time to train. One can look at data selected system as a decent trade-off between training time and translation quality.

4.6. Multi-domain Ensemble

We took the best model for every domain according to the average BLEU on the development sets and ensembled them during decoding. For weighted ensemble, we did a grid search and selected the weights using the development set. Table 6 presents the results of an ensemble on the Arabic-English language pair and compares them with the individual best model, OPUS, and a model built on *ALL*. As expected,

[7]These data-selection percentages have been previously found to be optimal when training phrase-based systems using the same data. For example see [29].

	Arabic-English			
	OPUS	ALL	ENS_b	ENS_w
tst13	32.2	36.1	31.9	34.3
tst14	27.3	30.2	25.8	28.6
avg.	29.7	33.2	28.9	31.5

Table 6: Comparing results of balanced ensemble (ENS_b) and weighted ensemble (ENS_w) with the best individual model and the concatenated model

balanced ensemble (*ENS*$_b$) dropped results compared to the best individual model. Since the domains are very distant, giving equal weights to them hurts the overall performance. The weighted ensemble (*ENS*$_w$) improved from the best individual model by 1.8 BLEU points but is still lower than the concatenated system by 1.7 BLEU points. The weighted ensemble approach is beneficial when individual domain specific models are already available for testing. Decoding with multiple models is more efficient compared to training a system from scratch on a concatenation of the entire data.

4.7. Discussion

The concatenation system showed robust behavior in translating new domains. [9] proposed a domain aware concatenated system by introducing domain tags for every domain. We trained a system using their approach and compared the results with simple concatenated system. The domain aware system performed slightly better than the concatenated system (up to 0.3 BLEU points) when tested on the in-domain TED development sets. However, domain tags bring a limitation to the model since it can only be tested on the domains it is trained on. Testing on an unknown domain would first require to find its closest domain from the set of domains the model is trained on. The system can then use that tag to translate unknown domain sentences.

5. Conclusion

We explored several approaches to train a neural machine translation system under multi-domain conditions and evaluated them based on three metrics: translation quality, training time and robustness. Our results showed that an optimum in-domain system can be built using a concatenation of the out-of-domain data and then fine-tuning it on the in-domain data. A system built on the concatenated data resulted in a generic system that is robust to new domains. Model stacking is sensitive to the order of domains it is trained on. Data selection and weighted ensemble resulted in a less optimal solution. The former is efficient to train in a short time and the latter is useful when different individual models are available for testing. It provides a mix of all domains without retraining or fine-tuning the system.

6. Acknowledgments

The research presented in this paper is partially conducted as part of the the European Unions Horizon 2020 research and innovation programme under grant agreement 644333 (SUMMA).

7. References

[1] F. Guzmán, H. Sajjad, S. Vogel, and A. Abdelali, "The AMARA corpus: Building resources for translating the web's educational content," in *Proceedings of the 10th International Workshop on Spoken Language Technology (IWSLT-13)*, December 2013.

[2] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 11th IWSLT Evaluation Campaign," *Proceedings of the International Workshop on Spoken Language Translation, Lake Tahoe, US*, 2014.

[3] A. Fujii, M. Utiyama, M. Yamamoto, and T. Utsuro, "Overview of the patent translation task at the ntcir-8 workshop," in *In Proceedings of the 8th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access*, 2010, pp. 293–302.

[4] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna, "Findings of the 2014 workshop on statistical machine translation," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA, June 2014.

[5] M.-T. Luong and C. D. Manning, "Stanford Neural Machine Translation Systems for Spoken Language Domains," in *Proceedings of the International Workshop on Spoken Language Translation*, Da Nang, Vietnam, December 2015.

[6] M. Freitag and Y. Al-Onaizan, "Fast domain adaptation for neural machine translation," *CoRR*, vol. abs/1612.06897, 2016. [Online]. Available: http://arxiv.org/abs/1612.06897

[7] C. Servan, J. M. Crego, and J. Senellart, "Domain specialization: a post-training domain adaptation for neural machine translation," *CoRR*, vol. abs/1612.06141, 2016. [Online]. Available: http://arxiv.org/abs/1612.06141

[8] C. Chu, R. Dabre, and S. Kurohashi, "An empirical comparison of simple domain adaptation methods for neural machine translation," *CoRR*, vol. abs/1701.03214, 2017. [Online]. Available: http://arxiv.org/abs/1701.03214

[9] C. Kobus, J. M. Crego, and J. Senellart, "Domain control for neural machine translation," *CoRR*, vol. abs/1612.06140, 2016. [Online]. Available: http://arxiv.org/abs/1612.06140

[10] M. van der Wees, A. Bisazza, and C. Monz, "Dynamic data selection for neural machine translation," in *Proceedings of the the Conference on Empirical Methods in Natural Language Processing*, September 2017.

[11] R. Wang, M. Utiyama, L. Liu, K. Chen, and E. Sumita, "Instance weighting for neural machine translation domain adaptation," in *Proceedings of the the Conference on Empirical Methods in Natural Language Processing,*, September 2017.

[12] B. Chen, C. Cherry, G. Foster, and S. Larkin, "Cost weighting for neural machine translation domain adaptation," in *Proceedings of the First Workshop on Neural Machine Translation*, September 2017.

[13] R. Wang, A. Finch, M. Utiyama, and E. Sumita, "Sentence embedding for neural machine translation domain adaptation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, August 2017.

[14] P. Williams, R. Sennrich, M. Nadejde, M. Huck, B. Haddow, and O. Bojar, "Edinburgh's statistical machine translation systems for wmt16," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 399–410. [Online]. Available: http://www.aclweb.org/anthology/W16-2327

[15] R. Sennrich, B. Haddow, and A. Birch, "Edinburgh neural machine translation systems for wmt 16," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 371–376. [Online]. Available: http://www.aclweb.org/anthology/W16-2323

[16] 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, Edinburgh, United Kingdom, 2011.

[17] N. Durrani, H. Sajjad, S. Joty, A. Abdelali, and S. Vogel, "Using Joint Models for Domain Adaptation in Statistical Machine Translation," in *Proceedings of the Fifteenth Machine Translation Summit (MT Summit XV)*. Florida, USA: AMTA, To Appear 2015.

[18] N. Durrani, H. Sajjad, S. Joty, and A. Abdelali, "A deep fusion model for domain adaptation in phrase-based mt," in *Proceedings of COLING*

2016, the 26th International Conference on Computational Linguistics: Technical Papers. Osaka, Japan: The COLING 2016 Organizing Committee, December 2016, pp. 3177–3187. [Online]. Available: http://aclweb.org/anthology/C16-1299

[19] S. Joty, H. Sajjad, N. Durrani, K. Al-Mannai, A. Abdelali, and S. Vogel, "How to Avoid Unwanted Pregnancies: Domain Adaptation using Neural Network Models," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, September 2015.

[20] M. Cettolo, "An Arabic-Hebrew parallel corpus of TED talks," in *Proceedings of the AMTA Workshop on Semitic Machine Translation (SeMaT)*, Austin, US-TX, November 2016.

[21] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen, "The united nations parallel corpus v1.0," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*, 2016.

[22] P. Lison and J. Tiedemann, "Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), may 2016.

[23] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 1715–1725. [Online]. Available: http://www.aclweb.org/anthology/P16-1162

[24] H. Sajjad, F. Dalvi, N. Durrani, A. Abdelali, Y. Belinkov, and S. Vogel, "Challenging Language-Dependent Segmentation for Arabic: An Application to Machine Translation and Part-of-Speech Tagging," in *Proceedings of the Association for Computational Linguistics*, 2017.

[25] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A fast and furious segmenter for arabic," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 11–16. [Online]. Available: http://www.aclweb.org/anthology/N16-3003

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the Association for Computational Linguistics (ACL'02)*, Philadelphia, PA, USA, 2002.

[27] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. L"aubli, A. V. Miceli Barone, J. Mokry, and M. Nadejde, "Nematus: a toolkit for neural machine translation," in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain: Association for Computational Linguistics, April 2017, pp. 65–68. [Online]. Available: http://aclweb.org/anthology/E17-3017

[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015. [Online]. Available: http://arxiv.org/pdf/1409.0473v6.pdf

[29] H. Sajjad, F. Guzmn, P. Nakov, A. Abdelali, K. Murray, F. A. Obaidli, and S. Vogel, "QCRI at IWSLT 2013: Experiments in Arabic-English and English-Arabic spoken language translation," in *Proceedings of the 10th International Workshop on Spoken Language Technology (IWSLT-13)*, December 2013.

Domain-independent Punctuation and Segmentation Insertion

Eunah Cho[*]*, Jan Niehues, Alex Waibel*

Institute for Anthropomatics and Robotics,
Karlsruhe Institute of Technology,
Karlsruhe, Germany
firstname.lastname@kit.edu

Abstract

Punctuation and segmentation is crucial in spoken language translation, as it has a strong impact to translation performance. However, the impact of rare or unknown words in the performance of punctuation and segmentation insertion has not been thoroughly studied. In this work, we simulate various degrees of domain-match in testing scenario and investigate their impact to the punctuation insertion task.

We explore three rare word generalizing schemes using part-of-speech (POS) tokens. Experiments show that generalizing rare and unknown words greatly improves the punctuation insertion performance, reaching up to 8.8 points of improvement in F-score when applied to the out-of-domain test scenario. We show that this improvement in punctuation quality has a positive impact on a following machine translation (MT) performance, improving it by 2 BLEU points.

1. Introduction

Punctuation and segmentation for automatic speech recognition (ASR) output is crucial in order to provide a better readability of the transcript as well as for a better performance in a subsequent application, such as machine translation (MT). Current state-of-the-art ASR systems often do not generate any or reliable punctuation marks. Thus, there has been an extensive amount of study on this issue.

A widely used method for punctuation and segmentation insertion utilizes language model with prosody features [1] due to its low latency. On the other hand, translation model-inspired systems [2, 3, 4] show an outstanding performance, both in accuracy of punctuation marks and improving the following MT performance. Using such monolingual translation systems, a non-punctuated source language is *translated* into a punctuated source language. Recently a neural machine translation (NMT)-based model [5] is shown to have a better performance, maintaining low latency in a real-time application.

While the monolingual translation system for punctuation insertion has been thoroughly investigated for its performance in subsequent applications, such as MT, and for real-time scenario constraints, such as latency and compactness of the model, domain-dependency of the model and its potential impact have been left under-explored.

In this paper, we investigate the domain-dependency of punctuation and segmentation insertion task and suggest that a generalization scheme over domain-specific words can greatly improve the performance. In this scheme, rare and unknown words are represented in their part-of-speech (POS) tokens for generalization.

In order to analyze the problem, we consider three scenarios where a different amount of matching in-domain data is available for training. In the first scenario, test data and training data are from the same resource. Therefore they share a same genre. In the second scenario, we consider a case where only a small amount of in-domain data is available for training the punctuation insertion model. In the last scenario, no matching in-domain data is available for training. Detailed data description for each scenario will be given in Section 5.

We then design three different schemes for modeling rare-words in punctuation insertion, whose details of the schemes will be given in Section 4.

For the punctuation systems, we use an attentional encoder-decoder model [6], so that a non-punctuated text is punctuated and true-cased using a translation framework. The punctuation insertion system is built for two source languages, English and German. We also translate punctuated test data into another language and measured the translation performance, in order to evaluate the impact of punctuation insertion in a further down text processing.

Our experiments show that generalizing rare and unknown words for punctuation and segmentation insertion task brings up to 8.8 points of improvements in F-score. Experiments on both manual and ASR transcripts show that generalizing rare and unknown words using POS tokens improves punctuation accuracy and also enhances the performance of following MT.

This paper is organized as follows. In Section 2, we overview past research in related fields. The problem statement and motivation as well as a detailed description of the task are given in Section 3. In Section 4 we will describe how

[*]Now in Amazon: eunahch@amazon.com

rare words can be generalized for better performances and the scenarios we consider in this work. Section 5 describes three different domain-match scenarios in testing scenario. Section 6 contains a detailed description of experimental setups, data preparation and evaluation settings, followed by Section 7 where the results and analyses are given. The paper is concluded in Section 8.

2. Related Work

Many previous research has been devoted to insertion of punctuation and segmentation into ASR transcripts. In [1], authors investigated using language model for the task, incorporated with prosody information such as pause duration. Authors in [7] explored maximum entropy model for the task using lexical as well as prosodic features. Modeling punctuation marks was also viewed as a sequential tagging problem in [8].

Punctuation insertion task is considered as a part of translation task in [2], where the authors build an implicit translation model, which translates non-punctuated source language into punctuated one. Later this work is extended by the authors in [3]. In this work, authors compared three approaches to view the punctuation insertion task as a machine translation problem. Among them, the explicit model where punctuation marks are inserted on the source side, prior to the translation, showed the best performance. This approach, however, can only be used under the assumption that the sentence boundary is already defined. Therefore, this approach would punctuation marks within pre-defined sentence boundaries only.

In [4], authors solved this problem by preparing the training data differently. They altered the training data so that sentence breaks are inserted in random locations. Therefore, sentence breaks can be observed anywhere throughout the data, not necessarily after a sentence-finalizing punctuation marks. On the other hand, they used a sliding window for testing.

Punctuation insertion task using neural networks has been studied using various architectures. The authors in [9] used a classifier based on a recurrent neural network (RNN). It is shown that a bidirectional recurrent network with attention mechanism can be effective for the task as well [10].

Recent development of attention-based NMT [6] has improved the performance machine translation greatly. An attentional NMT system consists of an encoder representing a source sentence and an attention-aware decoder that produces the translated sentence. In [5], a neural machine translation model is used as a method to insert punctuation marks into a non-punctuated source language. Authors investigated into the trade-off between network size and performance. By applying compact representation on the target side, they show that the NMT-based model outperforms PBMT-based model, maintaining low latency in an end-to-end translation scenario.

Domain adaptation and topic-matching problem for machine translation has been studied from various perspectives. In [11], authors gave a thorough analysis on different approaches to adapt a statistical machine translation system towards a target domain, using a small in-domain data. Techniques for domain adaptation in NMT has been explored and evaluated in an evaluation campaign in [12]. Rare word problem of NMT and potential solutions for machine translation scenario have been discussed in various literatures [13, 14]. Also, in [15] authors investigated how to adapt existing NMT systems to into a spoken language domain.

3. Domain-dependency of Punctuation and Segmentation Insertion Task

Domain adaptation for machine translation has received a great deal of attention [16], since applying an MT system into a test data of a different domain significantly affects translation quality.

In this paper, we study the impact of domain mismatch in the punctuation insertion task. Table 1 shows three separate excerpts extracted from a test data, which is punctuated using an NMT-based punctuation and segmentation system [5]. The system is trained on generic data and the test data contains domain-specific terminologies.

Table 1: *Three excerpts from test data, punctuated using a segmenter trained on generic data. Company and product names are anonymized.*

1	...use your existing Git and Gerrit Implementations. As well.
2	...server level should ever reference. The schema itself this.
3	...that might be an existing *#Company #Product1*. *#Product1-cont* system an, *#Product2* System it. Could be a replicated...

We can observe that the system provides rather poor quality of punctuation especially around rare words. Especially in the third excerpt, the product name (marked as *#Product1*), which originally consists of two tokens, is even separated by the inserted full stop.

Building separate domain-matching systems and obtaining a substantial amount of training data for each domain is costly. Therefore, we aim to build a punctuation insertion system which can be used relatively independent from the domain of test data. In order to generalize rare words, we explore methods using POS tags. The details are described in Section 4.

4. Modeling of Rare Words

In this work, punctuation and segmentation insertion task is considered as translation problem. Lower-cased text without any punctuation is translated into true-cased text with proper punctuation and segmentation. While the NMT-based punc-

tuation and segmentation insertion system shows a good result [5], the performance can be affected by rare words, as discussed in Section 3.

4.1. Definition

In this work, we define *rare word* as a word occurring less than 10 times throughout the training corpus. Additionally, we also need to model unknown words during training, in order to account them during the test case. Thus, we define *unknown word* as a word occurred only once in the training data.

4.2. Model

For generalization of rare words, we utilize POS information in order to consider syntactic information of them. In this work, we compare three different methods to represent rare and unknown words in a generalized form using POS.

- *unknown-NN*: Only unknown words are generalized. In order to generalize unknown words, we replace all words that occurred only once in the training data, into a POS-tag for noun (NN). We choose NN as it is the most frequently occurring POS throughout the corpus.

- *rare-NN*: Rare words, including unknown words, are generalized into the POS-tag for noun (NN).

- *rare-MF*: Unknown words are mapped into a POS-tag for noun (NN), while rare words are mapped into each word's most frequently (MF) used POS tag. Thus, we build a MF map from the training corpus. The MF map stores the most frequently used POS for each unique word in the training data. We obtain the POS-tag for each word of the training corpus using TreeTagger [17].

Test data is prepared in a similar manner for each criteria. Unknown word, that was not observed during training, is replaced into *NN*.

An excerpt from the training data is shown in Table 2 to depict *rare-MF* operation. In the first example, we can see that a rare word *thrives* is replaced into its most frequent tag, *VVZ* for the source side. In the same way, words like *beryllium*, *Adit*, or *tungsten* in the second example are replaced into POS tags.

Once rare words are generalized using different methods, further preprocessings are applied in order to form a parallel data for MT training. Details are discussed in Section 6.2.

5. Scenarios

While an extensive amount of previous research investigate the punctuation insertion task [3, 5], the impact of non-matching domain in test case is under-explored. In order to establish the importance of domain-match for this task, we model three scenarios of in-domain data availability by utilizing test data and training data from different sources. We utilize in-house English and German data for different scenarios.

5.1. Matching Data

The first scenario simulates the case where we have enough genre-matching training data. We take the training data and test data from the same source and model and evaluate the punctuation prediction system on the English TED data[1].

The training data comprises of ∼200K sentences of TED corpus, while the in-domain test data is around 1K sentences of TED. The audio reaches around 2 hours and 16 minutes.

By modeling this scenario, we aim to evaluate the impact of generalizing rare words into POS tokens in the punctuation prediction system, even when it is applied to a perfectly genre-fitting input.

5.2. Small In-domain Data

In the next scenario, we consider the case where only a limited amount of in-domain data is available. The model is trained using around ∼200K sentences of German TED data concatenated with 10K sentences of lecture corpus [18]. While the lecture corpus may share a similar style with the TED corpus (monologue, lecture), the lecture corpus contains a variety of domain-specific terms. The punctuation insertion system is then tested on a lecture data. Its manual transcript has 3K sentences, and its audio reaches around 6 hours and 32 minutes.

Detailed analysis on the data statistics is shown in Table 3. In the top two rows, we show the word count information in the original corpus, before we replace rare words into POS tokens. In the third line, we show how many words in the training/test data (among all occurrences) are considered as rare words according to the definition given in Section 4.1. We can see that around 4.5% of words of training data are rare words. About 2.0% of words in training data has occurred only once throughout the corpus. In the lecture test data, around 3.1% of words are unknown words, which were not observed during the training. As the university lecture corpus contains domain-specific terminologies, we can see that overall 4.9% of words in the test data are rare words. When using *rare-MF* method to generalize the rare and unknown words, the training data has 14.7K unique words. The number for test data is also decreased to 3.4K.

5.3. No in-domain data

In this scenario, we evaluate the English punctuation insertion built on the TED data, described in Section 5.1, on an online lecture corpus obtained from an internal project. The manual transcript reaches around 700 sentences, with the audio of a length of 1 hour and 55 minutes. The english online

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

Table 2: *POS replacement for rare and unknown word generalization*

Original	... a type of bacteria that thrives at 180 degrees. I think that's ...
rare-MF	... a type of bacteria that VVZ at 180 degrees. I think that's ...
Original	it doesn't have any beryllium in it. it's called the Pole Adit. and it does have tungsten, ...
rare-MF	it doesn't have any NN in it. it's called the Pole NP. and it does have NN, ...

Table 3: *Data statistics: German*

	Train	Test
All word	3,866.2K	53.6K
Unique word	137.2K	6.2K
Rare word	4.51%	4.87%
Unknown word	1.95%	3.07%

lecture mostly covers its most recent technologies. Consequently the test data contains a relatively higher proportion of rare and unknown words. By applying the system on this out-of-domain test data, we aim to show the effectiveness of our system handling rare words.

Table 4: *Data statistics: English*

	Train	Test:in	Test:out
All word	3,801.5K	20.3K	17.8K
Uniq word	63.5K	3.1K	1.8K
Rare word	2.63%	2.73%	4.54%
Unknown word	0.68%	1.07%	4.67%

Data statistics for English training and two test data are summarized in Table 4. First two rows, same as before, are showing general statistics of training and test data before the POS-replacement operation was applied. We can see that the ratio of rare words to all words in the corpus for both training and in-domain test data is around 2.7%. However, this ratio for out-of-domain test data rises to 4.5%. More importantly, the out-of-domain data has a significantly higher ratio of unknown word, 4.7%, compared to training as well as the in-domain test data. The statistics shows that the out-of-domain test data indeed includes a great proportion of unknown and rare-words, which is replaced into POS tokens during the replacement operation. Using *rare-MF* system, the training data has 11.9K unique words, in-domain test data 2.5K and out-of-domain test data 1.3K unique words respectively.

6. Experimental Setups

Since this process already decreases the vocabulary size effectively, we did not use any sub-word units. The detailed data statistics changed by this process will be discussed in Section 5.

In this section, we discuss the architecture of NMT-based punctuation insertion system as well as machine translation systems used to translate the punctuated test data.

6.1. Punctuation Insertion by NMT-based System

Inspired by [5], all punctuation insertion systems are built using the NMT framework `lamtram` [19], with an attention-based encoder-decoder model.

The models were all trained with Adam, where the algorithm is restarted twice and early stopping is applied. In [5], authors investigated the tradeoff between network size and performance. Following this work, we also configured that the encoder uses word embeddings of size 128 and a bidirectional LSTM [20] with 64 hidden layers for each direction. We use a multi-layer perceptron with 128 hidden units for the attention. For the decoder, we use conditional GRU units with 128 hidden units. Both networks are applied with dropout at every layer with the probability of 0.5.

6.2. Data Preparation

General data preparation follows the work in [4]. Except for tokenization and true-casing, no other preprocessing is applied for both input languages. The training data is randomly cut so that sentence boundaries can be observed in any location throughout the segment. Source side of the training data consists of lower-cased words and/or POS tokens. All punctuation marks are removed.

In this work, we build three systems to measure the impact of generalizing rare words. The details of the generalization scheme is given in Section 4.2. Since generalization of rare words largely decreases the number of unique words in the training data, we did not apply any sub-word operations on the training data for the three systems.

As a *baseline* system, we build a system following the work in [5], where no POS-replacement operation is applied. Source words are instead applied with byte-pair encoding of an operation size 40K. As another comparative system, in addition, we build a system *all-MF* where all words are replaced into their most frequent tag. In this system, thus, source side text consists of POS tags only. For English *all-MF* system, we introduce an additional POS tag for the word *I*. Since this word is always uppercased in English, we believe that it is fair to introduce a separate token for it. Vocabulary size for the *all-MF* system is therefore same as the number of possible POSs in each language.

For all systems with different source side representation, the target side follows the compact representation shown in [5]. Therefore, the target side corpus consists of *U* (meaning to be uppercased token), *L* (to be lowercased), and punctuation marks. As punctuation marks, we only consider sentence

boundary marks (.?!) and commas.

Test data is prepared in the same manner of training data, where random line breaks are inserted, in order to simulate ASR output.

6.3. Evaluation

As discussed in Section 5, English system, trained only on TED corpus, is evaluated on two different test sets, in-domain and out-of-domain test data. German system, whose training data includes a small lecture corpus, is tested on a lecture data.

The performance is measured intrinsically as well as extrinsically. The accuracy of punctuation marks inserted into manual transcripts is measured in F-score. Later this set is translated into another language in order to measure the impact of punctuation marks in translation performance. German lecture data is translated into English, and English TED data is translated into German. English lecture data is translated into Spanish following reference translation availability. The detailed description of each machine translation system used is given in Section 6.4.

The general system description for ASR is given in [21]. The training data for the English ASR system includes 450 hours of TED data. In addition, it also includes 30 hours of lecture courses obtained from the same project.

6.4. Machine Translation Systems

Punctuated German test data is translated into English for performance evaluation. The detailed description to German to English machine translation system can be found in [22, 23]. It is a phrase-based machine translation system that is trained on European Parliament and News Commentary corpora and adapted into TED and lecture domain.

The in-domain English test data, once it is punctuated, is translated into German. We use a phrase-based MT system described in [24]. The out-of-domain English test data has a Spanish reference translation. In order to evaluate the impact of punctuation prediction in this data, we use a neural machine translation system.

The English to Spanish NMT system is trained using the toolkit OpenNMT [25][2], with its default architecture. The training data includes EPPS, NC, and TED corpora, where a BPE of operation size 40K is applied. Additionally, we also use a data from Wikipedia[3] in order to support translation of domain-specific words [26].

The impact of inserted punctuation marks on each test data are measured in translation performance, in case-sensitive BLEU [27].

[2] https://github.com/OpenNMT/OpenNMT-py

[3] https://wikipedia.org

7. Results and Analysis

In this section, we show the results of experiments, followed by detailed analysis.

7.1. German Punctuation Insertion

In order to simulate scenarios with small in-domain training data, we build a system on German in-house data. Table 5 shows the results for German manual transcript.

Table 5: *Punctuation insertion performance: German lecture manual transcript*

System	F-score	De→En (BLEU)
Baseline	50.18	22.01
(1) unknown-NN	55.55	22.22
(2) rare-NN	55.23	22.30
(3) rare-MF	**56.79**	**22.61**
all-MF	47.21	21.25

When using *rare-MF* system to insert punctuation and segmentation into manual transcript, we can see that we achieve 6.6 points of F-score improvement over the *baseline*. This improvement also led to better translation, yielding 0.6 points of BLEU improvement by simply using different punctuation and segmentation into the same manual transcript prior to translation. As comparison, we also show the number of *all-MF*, where we can see the negative impact of over-generalization of this system.

Table 6: *Punctuation insertion performance: German lecture ASR transcript*

System	De→En (BLEU)
Baseline	18.71
(1) unknown-NN	19.12
(2) rare-NN	19.16
(3) rare-MF	**19.23**
all-MF	18.53

Table 6 also shows how much we can improve the translation of ASR transcript when we use the punctuation insertion system which generalizes rare words. Compared to the *baseline* where no generalization is applied, we improve the translation performance by 0.5 BLEU points. Thus, we can observe that the rare words in the lecture test data can be handled better when we use the *rare-MF* system.

When comparing the performance on manual transcripts and ASR, we see that in both cases rare-MF leads to the best performance. Also, in both cases, we improve the translation performance by 0.5 BLEU points.

7.2. English Punctuation Insertion

Table 7 shows the performance for English manual transcripts, tested on both in-domain test data and out-of-domain

test data. As mentioned, we show the intrinsic performance of punctuation insertion in F-score for each test data and extrinsic performance in BLEU. In-domain test data is translated into German, while out-of-domain test data, lecture test set, is translated into Spanish.

Table 7: *Punctuation insertion performance: English in-domain and out-of-domain manual transcript*

System	in-domainTest		out-domainTest	
	F-score	En→De	F-score	En→Es
Baseline	53.95	18.46	51.21	22.73
(1) unknown-NN	57.40	18.77	59.87	24.45
(2) rare-NN	59.23	**19.16**	57.93	24.45
(3) rare-MF	**59.63**	18.93	**59.99**	**24.68**
all-MF	38.10	16.87	42.82	20.89

We can observe that the *rare-MF* system yields a big improvement in F-score, 5.7 for in-domain test data and 8.8 for out-of-domain test data. This improvement in the F-score is also continued in the translation performance. The improvement in translation performance is bigger for out-of-domain test data, reaching around 1 BLEU point. The results show that generalizing rare and unknown words does not only improve the punctuation insertion but also the sequential applications' performance. The over-generalization of the additional system *all-MF* shows a worse performance for both test sets.

Table 8: *Punctuation insertion performance: English in-domain and out-of-domain ASR transcripts (BLEU)*

System	in-domainTest	out-domainTest
	En→De	En→Es
Baseline	13.74	19.21
(1) unknown-NN	**13.92**	20.22
(2) rare-NN	13.74	20.13
(3) rare-MF	13.77	**20.23**
all-MF	12.44	17.85

We apply the same set of experiments for English ASR transcripts. The results are shown in Table 8. For ASR transcripts, we translate the punctuated output into different languages and measure the performance in BLEU. As shown in the table, we can see that punctuation inserted from the *rare-MF* system into the in-domain test data did not yield a big performance improvement over the Baseline. For the out-of-domain test data, however, the *rare-MF* improves the translation performance around 0.8 BLEU points, by inserting the different punctuation marks and sentence segmentation only. The results show the importance of rare word generalization in the punctuation insertion system.

Another constant observation is that the more we lack of in-domain training data, the bigger improvement we may expect from using the *rare-MF* system.

7.3. Analysis

In addition, we measure the impact of using POS tokens over rare words in overall speed.

Table 9: *Running Time*

	English	German
Baseline	0m56.219s	1m49.818s
rare-MF	0m47.242s	1m45.493s
all-MF	0m42.341s	1m36.889s

The results are shown in Table 9. We measure the time used for decoding English in-domain test data and German test data, both manual transcripts. It is worth to note that the test sets are decoded in a CPU. We can see that decreasing vocabulary on the source side by replacing rare words into their POS tags, while keeping the target side vocabulary the same, overall testing time is decreased by 85∼90% of the baseline system. Faster runtime is therefore another advantage of generalization of rare words, which is often crucial in real-time applications.

Table 10 shows excerpts from the test data of the scenario where we have only a little amount of in-domain training data. We can observe that while the baseline system often misplaces a punctuation mark, *rare-MF* offers a better performance.

8. Conclusion

In this work, we showed that the performance of punctuation and segmentation can be greatly improved by generalizing rare and unknown words. In order to evaluate the impact of this system, we set three different scenarios on in-domain data availability. Our experiments show that we can improve the F-score by 5.7 points even for the scenario where we have a perfectly genre-matching training data. In the setting where in-domain data is not available at all and therefore rare/unknown words occur very frequently, F-score was improved by 8.8 points and subsequently 1 BLEU point in the following translation task of the punctuated test data.

In a detailed data analysis, we show that using this generalization also decreases source vocabulary dramatically. Compared to the baseline where we use sub-word units, the vocabulary size is decreased to 30∼37%. This also boosts faster running time during the testing.

Future work includes combining this model with other post-processing tasks for ASR, i.e. disfluency removal.

9. Acknowledgements

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 645452. This work was supported by the Carl-Zeiss-Stiftung.

Table 10: *Excerpts from output using different segmentation and punctuation system*

Excerpt 1	Baseline	wir sind nur zehn Kilometer voneinander. entfernt mit einem Auto fünfzehn Minuten.
	En. gloss	we are only ten kilometres from each other. away with a car fifteen minutes.
	rare-MF	wir sind nur zehn Kilometer voneinander entfernt mit einem Auto, fünfzehn Minuten.
	En. gloss	we are only ten kilometres from each other away with a car, fifteen minutes.
Excerpt 2	Baseline	Universitäten sind bottom-up. strukturiert Ideen entstehen in kleinen Ecken ...
	En. gloss	Universtities are bottom-up. structured ideas grow in small corners...
	rare-MF	Universitäten sind Bottom-up strukturiert. Ideen entstehen in kleinen Ecken...
	En. gloss	Universities are bottem-up structured. Ideas grow in small corners ...

10. References

[1] S. Rao, I. Lane, and T. Schultz, "Optimizing Sentence Segmentation for Spoken Language Translation," in *Proceedings of the eighth Annual Conference of the International Speech Communication Association (INTERSPEECH 2007)*, Antwerp, Belgium, 2007.

[2] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence Segmentation and Punctuation Recovery for Spoken Language Translation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008)*, Las Vegas, Nevada, USA, April 2008.

[3] 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 2011)*, San Francisco, California, USA, 2011.

[4] E. Cho, J. Niehues, and A. Waibel, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system," in *Proceedings of the International Workshop for Spoken Language Translation (IWSLT 2012)*, Hong Kong, China, 2012, pp. 252–259.

[5] ——, "Nmt-based segmentation and punctuation insertion for real-time spoken language translation," in *Interspeech*, 2017.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2015.

[7] J. Huang and G. Zweig, "Maximum Entropy Model for Punctuation Annotation from Speech," in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, Denver, Colorado, USA, 2002.

[8] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, 2010, pp. 177–186.

[9] M. Kazi, B. Thompson, E. Salesky, T. Anderson, G. Erdmann, E. Hansen, B. Ore, K. Young, J. Gwinnup, M. Hutt, and C. May, "The MITLL-AFRL IWSLT 2015 systems," in *Proceedings of the Eleventh International Workshop on Spoken Language Translation (IWSLT 2015)*, Da Nang, Vietnam, 2015.

[10] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration," *Interspeech 2016*, pp. 3047–3051, 2016.

[11] J. Niehues and A. Waibel, "Detailed analysis of different strategies for phrase table adaptation in smt," in *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*, 2012.

[12] E. Cho, J. Niehues, T.-L. Ha, M. Sperber, M. Mediani, and A. Waibel, "Adaptation and combination of nmt systems: The kit translation systems for iwslt 2016," in *IWSLT*, Seattle, WA, USA, 2016.

[13] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2015.

[14] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2015.

[15] M.-T. Luong and C. D. Manning, "Stanford neural machine translation systems for spoken language domains," in *Proceedings of the International Workshop on Spoken Language Translation*, 2015.

[16] P. Koehn and J. Schroeder, "Experiments in domain adaptation for statistical machine translation," in *Proceedings of the second workshop on statistical machine translation*. Association for Computational Linguistics, 2007, pp. 224–227.

[17] H. Schmid and F. Laws, "Estimation of Conditional Probabilities with Decision Trees and an Application to Fine-Grained POS Tagging," in *Proceedings of the 22nd International Conference on Computational Linguistics, Proceedings of the Conference (COLING 2008)*, Manchester, UK, 2008.

[18] 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 2012)*, Istanbul, Turkey, 2012, pp. 3409–3414.

[19] G. Neubig, "lamtram: A toolkit for language and translation modeling using neural networks," http://www.github.com/neubig/lamtram, 2015.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] T.-S. Nguyen, M. Mueller, M. Sperber, T. Zenkel, K. Kilgour, S. Stueker, and A. Waibel, "The 2016 kit iwslt speech-to-text systems for english and german," in *IWSLT*, Seattle, WA, USA, 2016.

[22] E. Cho, C. Fügen, T. Herrmann, K. Kilgour, M. Mediani, C. Mohr, J. Niehues, K. Rottmann, C. Saam, S. Stüker, and A. Waibel, "A real-world system for simultaneous translation of german lectures," in *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH 2013)*, Lyon, France, 2013.

[23] M. Müller, T. S. Nguyen, J. Niehues, E. Cho, B. Krüger, T.-L. Ha, K. Kilgour, M. Sperber, M. Mediani, S. Stüker, *et al.*, "Lecture translator speech translation framework for simultaneous lecture translation," *NAACL HLT 2016*, p. 82, 2016.

[24] I. Slawik, M. Mediani, J. Niehues, Y. Zhang, E. Cho, T. Herrmann, T.-L. Ha, and A. Waibel, "The KIT Translation Systems for IWSLT 2014," in *Proceedings of the eleventh International Workshop for Spoken Language Translation (IWSLT 2014)*, Lake Tahoe, California, USA, 2014.

[25] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "OpenNMT: Open-Source Toolkit for Neural Machine Translation," *ArXiv e-prints*, 2017.

[26] J. Niehues and A. Waibel, "Using wikipedia to translate domain-specific terms in smt," in *Proceedings of the eighth International Workshop on Spoken Language Translation (IWSLT 2011)*, San Francisco, California, USA, 2011.

[27] 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 on Association for Computational Linguistics (ACL 2002)*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, 2002, pp. 311–318.

Synthetic Data for Neural Machine Translation of Spoken-Dialects

Hany Hassan, Mostafa Elaraby, Ahmed Y. Tawfik

Microsoft AI & Research

hanyh,a-moelar,atawfik@microsoft.com

Abstract

In this paper, we introduce a novel approach to generate synthetic data for training Neural Machine Translation systems. The proposed approach supports language variants and dialects with very limited parallel training data. This is achieved using a seed data to project words from a closely-related resource-rich language to an under-resourced language variant via word embedding representations. The proposed approach is based on localized embedding projection of distributed representations which utilizes monolingual embeddings and approximate nearest neighbors queries to transform parallel data across language variants.

Our approach is language independent and can be used to generate data for any variant of the source language such as slang or spoken dialect or even for a different language that is related to the source language. We report experimental results on Levantine to English translation using Neural Machine Translation. We show that the synthetic data can provide significant improvements over a very large scale system by more than 2.8 Bleu points and it can be used to provide a reliable translation system for a spoken dialect which does not have sufficient parallel data.

1. Introduction

Neural Machine Translation (NMT) [1] has achieved state-of-the-art translation quality in various research evaluations campaigns [29] and online large scale production systems [3] and [4]. With such large systems, NMT showed that it can scale up to huge amounts of parallel data. However, such large parallel data is not widely available for all domains and language styles. Usually parallel training data is widely available in written formal languages such as UN and Europarl data.

Real-time speech translation systems support spontaneous, open-domain conversations between speakers of different languages. Speech Translation Systems are becoming a practical tool that can help in eliminating language barriers for spoken languages. Those machine translation systems are usually trained using NMT with large amount of parallel data adapted from written data to the spoken style [5]. This is a valid approach when the spoken and written languages are similar and mainly differ in style. For many languages, the written and spoken forms are quite different §2. While the written form usually has an abundance of parallel data available to train a reliable NMT system; the spoken form may not have any parallel data or even, in some cases, a standardized written form.

In this paper, we propose a novel approach to generate synthetic data for NMT. The proposed approach transforms a given parallel corpus between a written language and a target language to a parallel corpus between the spoken dialect variant and the target language. Our approach is language independent and can be used to generate data for any variant of the source language such as slang, spoken dialect or social media style or even for a different language that is closely-related to such source language.

The synthetic data generation approach is based on two simple principles: first, distributional word representation (word embeddings) can preserve similarity relations across languages [6]. Secondly, a localized projection can be learned to transform between various representations [7]. We assume that we are trying to learn a translation system between F' and E, where F' is a variant of F, i.e. a spoken dialect. We start from parallel corpus between the two standard languages F and E, then we transform it into a three-way corpus between F, E and F'. The proposed approach assumes the existence of a seed bi-lingual lexicon or a small seed parallel data between F' and either F or E.

The proposed approach is motivated by the assumption that both Language F and its variant Language F' share some vocabulary, have similar word orders and share similar bi-lingual characteristics with Language E. We start by constructing a continuous word representation (i.e. word2vec [6]) for each one of the three languages. Using the seed bi-lexicon between either E and F' or F and F', we train a local projection to transform the words across the different representation spaces.

We used the proposed approach to generate spoken Levantine-English data from Arabic-English data then we experimented with utilizing the generated data in various settings to improve translation of the spoken dialect. The rest of this paper is organized as follows, Section §2 presents an overview of spoken dialects since it is the focus application of this work. Section §3 discusses related work. Section §4 presents a brief overview of Neural machine translation. Section §5 discussed in detail the proposed approach for generating data. Section §6 presents the experimental setup. Finally,

we discuss the results and conclude in section §7.

2. Spoken Language Variants

Some languages present an additional challenge to Spoken Language Translation (SLT) when the spoken variant differs significantly from the written one. Moreover, sometimes the spoken language used in the daily life is quite different than the standard form used in the education system as well as in formal communication such as news papers and broadcast news. For example, Singapore English (Singlish) is an English-based creole with a mix of English, Mandarin, Malay, and Tamil [8]. Similarly, the standard form of written Arabic is Modern Standard Arabic (MSA); however, it is not the spoken mother tongue by Arabic speakers. The Arabic spoken dialects vary by geo-graphical region with at least five dialects: Egyptian, Levantine, Iraqi, gulf, and North African. While all dialects are stemmed from MSA, they are quite different phonologically, lexically, morphologically and syntactically. For example, spoken colloquial Levantine Arabic conversations share between 61.7% and 77.4% of their vocabulary with a written news corpus from the same region [9]. This results in spoken dialects that are quite different and not even well interpreted between Arabic speakers of different dialects.

Most of the spoken language variants stem from a more formal written language such as Singlish from English and Levantine from MSA. While the spoken dialects do not usually have parallel data, they enjoy a wide adoption on social media which results in large monolingual corpora for such spoken variants. In this work, we are proposing a novel approach to overcome such limitation for spoken languages through generating parallel data leveraging the spoken dialects monolingual data and the written form parallel data.

In this paper we focus on Levantine-English translation as the pressing need for such translation systems due to the refugee crisis that dictates the need for a reliable open-domain translation from Levantine to English.

3. Related Work

There have been a number of proposed approaches to learn synthesized translation units for statistical machine translation systems such as [14], [15] and [7]. Such approaches focused on learning translation rules that would fit into a statistical phrase-based system. Those approaches do not fit into Neural Machine Translation (NMT) systems which require full context to learn to encode the sentences.

A number of approaches have been proposed utilizing monolingual target data into NMT training. Most notably, [16] used monolingual sentences by generating pseudo parallel data through back-translating the monolingual data and using it in the reverse direction to improve NMT systems. Back-Translation showed significant improvement especially in domain adaption setups. The back-translation approach is not directly comparable to ours, since ours does not require a pre-trained system while back-translation does require one. However, we are using a seed parallel data as a source of our lexicon and it would be fairly comparable to use such data in both settings as we report in our experiments.

Dialectal Arabic translation has been a well-known problem; [11] tried to solve this problem by crowd-sourcing translation for dialect data. They translated around (160K sentences) of Levantine and Egyptian data. The main limitation of this approach is that it is quite limited and not scalable. The vocabulary of the collected data is not sufficient to provide open-domain translation system. On the other hand, [12] and [13] tried to solve the problem by applying rule-based transformation between Levantine or Egyptian to MSA. The main limitation of such approaches is that they require extensive linguistics knowledge to design the conversion rules which are not flexible to new vocabulary and styles that are constantly being introduced to the spoken languages.

4. Neural Machine Translation

Neural Machine Translation is based on Sequence-to-Sequence encoder-decoder model as proposed in [31] along with an attention mechanism to handle longer sentences [1] and [25].

In this work, we use an in-house implementation [4] for attention-based encoder-decoder NMT which is similar to [1]. NMT is modeling the log conditional probability of the target sequence given the source as shown in eqn1:

$$\log p(y|x) = \sum_{k=1}^{n} \log p(y_k|y_{<k}, x) \tag{1}$$

NMT follows encoder-decoder architecture; the encoder is a bidirectional recurrent neural network (LSTM) that calculates the hidden encoder state at each word $h_1 h_2 ... h_m$. The decoder is another recurrent neural network (LSTM) as well that calculates the hidden state at each decoded output state $s_1 s_2 .... s_n$. Then a softmax is applied to get a distribution over target words.

$$y_k = softmax(g(y_{k-1}, s_k, c_k)) \tag{2}$$

where c_k is calculated by the attention mechanism which is a weighted sum of the encoder's hidden states that determines the importance of each encoder hidden state to the predicted output. The attention mechanism represents the variable length input sequence as a weighted fixed-dimension context vector c_k

$$c_k = \sum_{i=1}^{m} \alpha_{ki} h_i \tag{3}$$

where $\alpha_k i$ is calculated as a normalized weight of the association between the previous decoder state s_{k-1} and the current encoder state h_i which is calculated as a dot product as described in [25].

During training, all model's parameters are optimized jointly using stochastic gradient methods to maximize the conditional probability of all sentence pairs in the training data. At decoding time, one word is predicted at each step, a beam search is used to score the best translation path.

5. Synthesized Data Generation

Our data generation approach is motivated by two observations: firstly, distributional representations of words have been found to capture syntactic and semantic regularities in languages. In such continuous representation space, the relative positions between words are preserved across languages [6]. Secondly, the representation spaces have localized sub-clusters of neighboring data points that form smooth manifolds [18] which can be leveraged to learn a localized transformation between the sub-clusters in different spaces across languages [7]. Since the sub-clusters are formed by similar words, a mapping can be learned between sub-clusters across representations. We exploit those characteristics to design our synthetic data generation approach.

The proposed approach assumes the availability of three resources: (1) parallel data between Language F and Language E, (2) a seed lexicon or seed parallel data between either E and F' or F and F'. (3) Monolingual corpora for E, F and F' to train word vectors. The resulting synthesized data is a three-way data (F-F'-E). In this paper, we use a seed parallel data to acquire the lexicon between E and F' through word alignment. However, a pre-existing lexicon can be used exactly the same way.

Figure 1 illustrates the data generation process. For illustration purposes, let's assume that E is English, F is Spanish (ESN) and we would like to generate F' which is Catalan (CAT) to English parallel data [1] . Furthermore, we assume that we have a seed lexicon between Catalan F' and English E which we call *BiLexicon*.

We build three distributional representations (i.e. word2vec) using monolingual corpora: the first is a target representation for E, English in our example . The second is mixed source representation F-F' (Spanish-Catalan in our example). And the third is a Catalan (F') only embedding.

The data generation proceeds as follows:

- For each English word e in a sentence from F-E parallel data, we query its k-nearest neighbors (k-NN)
- k-NN query on the E embedding results in a sub-cluster of k English words around e.
- If the k queried neighbors do not contain at least m words in *BiLexicon*, we repeat the query with $2k$.
- If no m neighbors words can be retrieved, the process terminates for this word and move to the next word.
- We use m to query *BiLexicon* for equivalent words in the F' space.
- As shown in Figure 1, we use the two localized sub-clusters in E (English) and F' (Catalan) spaces to learn a localized projection between the two spaces. This is done using Local Embedding Projection (LEP) §5.3.
- The locally trained LEP is used to project the current E word e to its equivalent vector in the F' space.
- We perform k-NN query around the projected vector in the F' (Catalan) space to get n candidates words.
- We then rank the n candidates words according to their similarity with the F (Spanish) words f aligned to the current English word e based on word alignment of the F-E parallel data.
- The similarity is calculating cosine Similarity (SIM) in the Spanish-Catalan space between the candidate Catalan words and the Spanish word(f).
- The top ranked Catalan word f' is selected and substituted in palace of f
- Alternatively, we can obtain the alignment information between E (English) and F (Spanish) words either by conventional word alignment techniques or by using Bi-Lingual embeddings as described in Section §5.1.

It is worth noting that for one-to-many mappings, we construct a composed vector for the multiple words by performing addition of their corresponding vectors. There are a few other approaches to compose multi-words vectors. However, it has been shown empirically that simple additive method achieves good performance [27].

Later on, we discuss the main components we utilize in the generation process: Word Representation §5.1, efficient Nearest Neighbors Search §5.2 and Local Embedding Projection §5.3.

5.1. Word Representation

Continuous representations of words have been found to capture syntactic and semantic regularities in languages [6]. The induced representations tend to cluster similar words together. We directly use continuous representations learned from monolingual corpora such as Continuous Bag-Of-Words (CBOW) representation. In such continuous representation spaces, the relative positions between words are preserved across languages. As shown in Figure 1, we learn three independent representations for spoken source, target and mixed sources. Those can be learned from monolingual corpora using off-the-shelf tools such as word2vec [6].

We require a mapping between the words in the original parallel corpus which can be obtained by performing word alignment on the parallel sentences. Alternatively, this requirement can be relaxed by using a bi-lingual embedding trained on any parallel corpus such as Bivec [26]. Instead of using word alignment to map the source word to target

[1] The languages in the example are for illustration purposes only

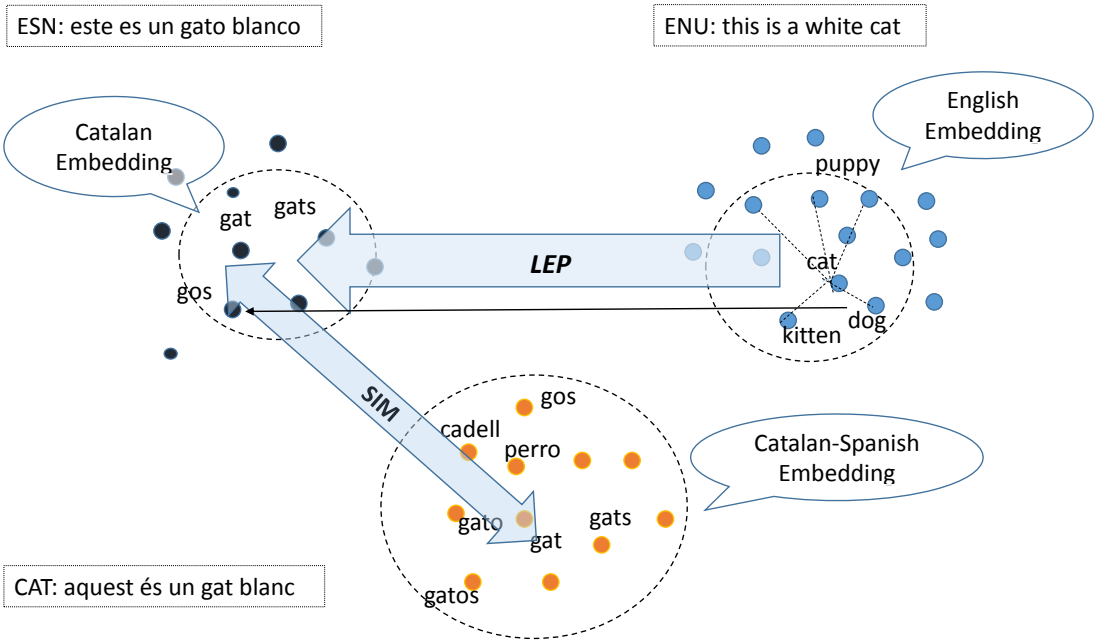


Figure 1: Synthetic Data Generation Using LEP

word(s), we initiate a query to bilingual representation to retrieve the most likely target word mapped to a given source word. This can be handy in the case of using comparable corpus rather than parallel corpus. We evaluate the merit of this approach in §6.4

5.2. Nearest Neighbors Search

The algorithm discussed above, requires an extensive number of k-NN queries per word, which are the most time-consuming part of the procedure. A brute force k-NN query requires a linear search over the whole source or target vocabulary which is usually in the order of millions requiring $O(n)$ search. This dictates the need for a fast approximate k-NN query technique. While such techniques are widely used in various machine learning areas especially in vision application, they are not well explored for text applications.

Approximated k-NN query usually involves two steps, an offline index construction step and an online query step. While the offline step does not affect the run-time, it can be memory consuming. A good approximation sacrifices the query accuracy a little bit, but speeds up the query by orders of magnitude. Locality Sensitive Hashing (LSH) [19] is a popular technique, but its performance decreases as the number of dimensions grows, therefore it is not a good match for high dimensional spaces like ours. In this paper, we use Multiple Random Projection Trees (MRPT) [20] for approximated k-NN queries.

MRPT [20] uses multiple random projection trees to get a more randomized space-partitioning trees. The random projection trees result in splitting hyperplanes that are aligned with random directions sampled from the space hypersphere instead of the coordinate axes. Moreover, it utilizes voting search among the random projection trees to provide more randomization that leads to fast query times and accurate results. At run-time, a query $\mathbf{p}$ is routed down in several trees, and then a linear search, similar to RBV, is performed in the union of the points of all the leaves the query point fells into, the result is the approximated k-nearest neighbors to $\mathbf{p}$.

5.3. Localized Embedding Projection (LEP)

The k-NN queries result in two local clusters as shown in Figure 1. Given a word in one of the sub-clusters we want to find similar word(s) in the corresponding target sub-cluster. We use Localized Embedding Projection (LEP) to achieve this task.

LEP is based on simple intuition: the two sub-clusters represent smooth manifolds where each data point in a sub-cluster can be mapped to a corresponding data point in the other sub-cluster using local linear transformation. LEP has been successfully used in [7] to transform between various representations based on the *locally linear embedding* method which was originally proposed in [18] for dimensionality reduction.

LEP utilizes a localized projection matrix for each word,

this is unlike global linear projection, as proposed in [6], which uses a single projection matrix for the all words in the space. As shown in [7], it can be brittle to small non-linearity in the representation vector space and therefore it is not a good choice for all possible words. Unlike global projection, local projection requires an additional k-NN query to find the neighbors of each word.

In LEP, a linear projection $W_{\mathbf{f}}$ is learned for each word $\mathbf{f}$ to map between its neighbors to the neighbors of the projected points in the projected/translation space. $(\mathbf{f}_1, \mathbf{e}_1), (\mathbf{f}_2, \mathbf{e}_2), \ldots, (\mathbf{f}_m, \mathbf{e}_m)$, $\mathbf{f}_i \in N(\mathbf{f})$.

Let's denote f and e as source side and target side words respectively, and $\mathbf{f}$ and $\mathbf{e}$ as the corresponding words vectors. Following [6], we learn the linear projection W from the translations of the n most frequent labeled source side phrases: $(\mathbf{f}_1, \mathbf{e}_1), (\mathbf{f}_2, \mathbf{e}_2), \ldots, (\mathbf{f}_n, \mathbf{e}_n)$. Denote $F = [\mathbf{f}_1^T, \mathbf{f}_2^T, \ldots, \mathbf{f}_n^T]^T$, $E = [\mathbf{e}_1^T, \mathbf{e}_2^T, \ldots, \mathbf{e}_n^T]^T$. W is calculated by solving the following linear system:

$$FW = E,$$

whose solution is:

$$W \approx (F^T F)^{-1} F^T E.$$

Once the linear transform W is known, for each word $\mathbf{f}$, $\mathbf{f}W = \bar{\mathbf{e}}$ is the location in the target side that should be close to the target words representing similar meaning. A k-NN query can fetch all the target word vectors near point $\bar{\mathbf{e}}$.

6. Experimental Setup

We used the proposed approach to generate spoken Levantine-English data from Arabic-English data then we experimented with utilizing the generated data in various settings to improve translation for the spoken dialect.

6.1. Datasets

The only publicly available Dialectal Arabic to English parallel corpus is LDC2012T09[2] [11]. It consists of about 160K sentences of web data of mixed Levantine and Egyptian manually translated to English. We use this data set as our baseline and as a source for the seed lexicon between English and Levantine.

Our main focus is to develop an open-domain conversational translation system for Levantine-English. In recent translation evaluations, OpenSubtitles data [32] has been found to yield good translation quality for conversational domains compared to other data sources [5]. Therefore, we opt for using OpenSubtitles-2013[3] which consists of 3M sentences as our Arabic(MSA)-to-English parallel corpus, to generate Levantine-English Parallel corpus.

We have created a three-way test set to evaluate this work (LEV-ENG-Test), where the source is transcription of spontaneous Levantine audio conversations translated into both English and MSA Arabic. The test set is composed of 6K sentences and has been used to report all results in this paper.

[2]https://catalog.ldc.upenn.edu/LDC2012T09

[3]http://opus.lingfil.uu.se/

Corpus	English	Arabic MSA	Levantine
# of Tokens	2B	1.1B	106M
# of Word Vectors	5.1M	6.8M	1.5M

Table 1: Monolingual corpora used in experiments.

We used monolingual corpora to train three distributional representations of English, Levantine and Mixed (MSA with Levantine. The data mostly consist of Gigaword corpora, UN data, Subtitles and web crawled data. The information of these corpora is listed in Table 1.

After that we use the off-the-shelf Word2Vec [6] to generate the word embeddings for each language using the Continuous Bag-Of-Words scheme, where the number of dimensions $d = 250$, $window = 5$, $mincount = 5$.

6.2. Data filtering

Our proposed approach depends on the quality of the parallel data, we have noticed that OpenSubtitles data has a lot of misaligned or badly translated sentences. Therefore, we have trained a decision tree classifier to identify whether the sentence pair is noisy or not. We reject the sentence pairs that are noisy. The decision tree classifier utilizes features from the meta-data of the aligned sentence pairs, namely: number of source words, number of target words, unaligned percentage, length-normalized alignment confidence score and percentage of one-to-one alignments. We used 150 sentences manually annotated to train the classifier with Gini impurity with minimum samples split of 2 and minimum samples leaf of 1.

On the word level, we have applied a named entity tagger to detect named entities on either source or target sides to avoid mangling them. We also used a stop-word list to avoid mapping them.

6.3. NMT model and Pre-Processing

Our NMT system is described in §4, we use a bidirectional encoder with 1024-units LSTM and 2 layers decoder with attention. We use embedding size of 512 and dropout of 0.2.

For pre-processing, we use Byte Pair Encoding PBE [29] with 32000 merging operations separately on the source and target. This results in 35K source and 34K target vocabularies. We limit the length of the sentences to 50 words. The training is done using Stochastic Gradient Descent (SGD) with Adam[21]. We use mini-batch size of 64 and train for 1M steps. The translation quality is measured with lowercased BLEU.

Across all experiments we use those hyper parameters for the data generation process described in §5: $k = 200$, $n = 3$ and $m = 5$.

System	Data Size	B LEV-ENG-Test
Baseline	160K	16.15
Gen-BiVec	210K	16.43
Gen-Align	210K	16.98

Table 2: Translation performances using BLEU on LEV-ENG-Test for using Bivec vs. word alignment

6.4. Bivec vs Word Alignment

In the first set of experiments, we have evaluated whether we should use word-alignment information or Bivec §5.1 to connect the source and target words in the given parallel data. As shown in Table 2, our Baseline is trained on LDC2012T09 (160K) of mostly Levantine-English data. We then generate 50K sentences from Arabic-English Subtitles data with bilingual embedding (Gen-Bivec) and without it (Gen-Align). When we are not using Bivec, we just use the word alignment information on the Arabic-English parallel corpus to get the mapping between the words. The result shows that using alignment information is better than using Bivec in this case. It worth noting that using Bivec may be handy if the data is comparable data. In the rest of this work we used word alignment information since it yields better performance.

6.5. Data Generation Experiments

In this set of experiments, we added more generated data from the subtitles data applying the filtering described above §6.2. We end up with 1.1M sentences candidates for generation which we use for generating LEV-ENG data. In this setup, we also compared our approach with back-translation [16] which is commonly used with NMT. The back-translation is not directly comparable to ours, since ours does not require a pre-trained system while back-translation does require one. However, we are using a seed parallel data as a source of our lexicon and it would be fairly comparable to use such data in both settings.

Furthermore, we investigated two different models to utilize the synthetic data. The first just used the LEV-ENG data while the second leveraged the 3-way characteristic of the generated corpus LEV-MSA-ENG.

We train the following systems:

- Baseline: This is trained on LDC Levantine-English corpus of 160K. Which is also part of all other systems reported below.
- Baseline-PBMT: this is the same as above but trained as a phrase-based system, following standard practice.
- Baseline-MSA: This is trained on LDC data in addition to 1.1M sentence pairs of filtered subtitles data which is MSA-English.
- BT: We trained an English-Levantine system similar to the Baseline though in the reverse direction; we used it to back-translate the 1.1M subtitles data from English into Levantine.
- Gen-Mono-1M: This is the system using the generated LEV-ENG data.
- LEV-MSA–MSA-ENG: This a pipeline system where we train a system to convert LEV to MSA using the 3-way generated data, followed by MSA to ENG translation.

System	BLEU on LEV-ENG-Test
Baseline	16.15
Baseline-PBMT	16.42
Baseline-MSA	15.37
BT	16.59
Gen-Mono	17.33
LEV-MSA–MSA-ENG	12.87

Table 3: Translation performances in BLEU for NMT with Generated data

Table 3 shows that adding the MSA subtitles data (Baseline-MSA) hurt the performance, this is quite expected since the data is mainly MSA but it add a fair comparison in terms of the size of the training data. The phrase-based baseline is slightly better than NMT baseline as expected in such low resource case.

Back-translation helped a little bit (0.3 BLEU), we think the system trained on LDC parallel data is quite small to provide good lexical coverage to generate variates of the translated data that can help in back-translation.

Adding the synthetic data (Gen-Mono) is quite useful and improves the performance by more than 2 BLEU points. Compared to back-translation, the synthesized data utilized monolingual representation which can lead to lexical varieties that help in having better translation examples.

Since the generated data is a 3-way corpus LEV-MSA-ENG, we can leverage this by training a system that translates from LEV to MSA. At run-time, we use a pipeline of two systems: LEV-MSA followed by MSA-ENG. We experimented with two variants of LEV-to-MSA system, subwords-based and character-based. We found out that the system is not producing reasonable results since it produces MSA words not related to the LEV words in input. We think one reason is that MSA and LEV shares a lot of their vocabulary together; in our monolingual data sets listed in Table 1, they share 58% of their vocabulary. The system tends to replace MSA words (in LEV input) to other MSA words. The resulted outcome is very noisy MSA sentences that not closely related to the LEV input.

6.6. Open-domain NMT System Experiments

Our main objective in this work is to enable large scale NMT systems to support spoken dialects. Therefore, we experimented with a very large scale Arabic-English open-domain

System	LEV-ENG-Test	LEV-MSA-ENG	NIST08
Large-Sys	25.03	28.20	53.45
Large+GenData	27.91	28.32	53.42
Large+Adapted	27.37	27.45	52.97

Table 4: Translation performances in BLEU for Large Scale NMT with Generated data

system trying to adapt it to Levantine using the synthetic data. The large scale system uses UN data, subtitles data and various web crawled data with a total of 42M parallel sentences. The system is an ensemble of two identical systems that only differ by initialization, each ensemble is trained for 10 epochs on the data. We tried two approaches to utilize the synthetic data: adding it to the training data as usual and adapting one of the two ensembles by continuing to train it on the synthetic data for 2 more epochs, similar to the approach proposed in [23].

For this set of experiments, we have added 2M synthetic Levantine-English sentences. We also report results on NIST-08 Arabic-English which is a 4-references test-set [4]. Furthermore, we report results on the human converted LEV-MSA-ENG which is the same as LEV-ENG-Test testset but translated into MSA as well by human annotators. Since LEV data is converted to MSA by annotators, translating the human-converted test set can represent the oracle score that we can get using an MSA trained system on this test set. This would help us understand how good the system using the generated data compared to MSA systems.

As shown in Table 4, we get a very good improvement when adding the synthetic data as additional training data (Large+GenData) with 2.8 BLEU points. The performance of the system with the synthetic data is just 0.3 BLEU less than the oracle score on the human translated MSA (27.91 vs 28.20). Moreover, the addition of the synthetic data did not negatively affect the MSA NIST08 test sets as well; this simply enables us to have a single system to serve both written and spoken variants. This is a nice characteristic of NMT systems where encoders can successfully handle varieties of source data as has been utilized in multi-lingual systems [22].

Adapting the system did help as well but not as good as re-training from scratch, however it may be a good option to avoid retraining the large system again.

Figure 6.6 shows some cherry picked examples that show the improvement of the proposed approach compared to GNMT [3] online neural system. It is quite clear that our system is doing much better compared to a large scale neural system.

7. Discussion and Conclusion

In this paper we presented a novel approach for generating synthetic parallel data for spoken dialects to overcome the limitations of the training data availability for such language variants. We show that we need to start from a corresponding parallel data and a seed lexicon or small parallel data. The results show that this approach is quite efficient and useful to improve general purpose NMT systems to the spoken variants.

As for the future work, we would like to investigate the utilization of this approach for more languages as well as different variants such as social media text translation. As a further step, we are investigating the possibility of training the transformation process end-to-end within the neural machine translation system using a single neural network through learning the transformation from the sample seeds while making use of the monolingual corpus to learn the embeddings.

8. Acknowledgments

We would like to thank Christian Federman and Will Lewis for fruitful discussions and helping in creating the test set.

9. References

[1] Bahdanau, D., Cho, K., Bengio, Y., "Neural machine translation by jointly learning to align and translate", ICLR 2015

[2] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units.", In ACL 2016.

[3] Wu, Y., Schuster, M., Chen, Z., Le, Q., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X. , Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang,W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J., "Google's neural machine translation system: Bridging the gap between human and machine translation.", https://arxiv.org/abs/1609.08144

[4] Devlin, J., "Sharp Models on Dull Hardware: Fast and Accurate Neural Machine Translation Decoding on the CPU", EMNLP 2017: 2810-2815

[5] Niu, X. and Carpuat, M., "The umd machine translation systems at iwslt 2016: English-to-french translation of speech transcripts.", In IWSLT, 2016.

[6] Mikolov, T.,Sutskever, I., Chen, k., Corrado, G., Dean, J., "Distributed representations of words and phrases and their compositionality.", In NIPS 2013.

[7] Zhao, K., Hassan, H., Auli, M., "Learning translation models from monolingual continuous representations.", In NAACL HLT 2015

[8] Lisa Lim, L., Deterding, D., "Singapore English", Journal of the International Phonetic Association, 2010.

[4] https://catalog.ldc.upenn.edu/LDC2010T01

Source	النفسية بتختلف كثير مثل لما تكون حاطه بصحن كبير بعدين بتحطه بكاسة
GNMT	Psychological differ many as if it was surrounded by a large dish after two
Our System	The psychological differs a lot like when it's put in a big dish then it is in a cup
Reference	The psychological situation differs a lot when you're put in a huge plate then suddenly you're placed in a glass.
Source	مثلا بيفضلوا أنه أنا أكون مجوزة حدا من ذات البيئة تبعي.
GNMT	For example, they would prefer that I be consumed by the same environment
Our System	For example they prefer that I am married to one of the same environment
Reference	For example they prefer I had married someone from my environment.
Source	مو أكثر شيء تأثر على الفيلم و على نجاحه
GNMT	Mo most influenced by the film and its success
Our System	Not the most thing affecting the movie and its success
Reference	Not the thing that effect the movie and its success the most.
Source	ليك هلق هون السؤال الثاني الغبي شوي
GNMT	The second dumb question is Shui
Our System	Now here is the second stupid question
Reference	Look now here is the second question that's a little stupid.

Figure 2: Examples of system output and comparison

[9] Al-Shareef, S. and Hain, T., "An investigation in speech recognition for colloquial Arabic", INTERSPEECH 2011

[11] Zbib, R., Malchiodi, E., Devlin, J., Stallard, D., Matsoukas. S., Schwartz, R., Makhoul, J., Zaidan, O., Callison-Burch, C., "Machine translation of arabic dialects." , In HLT/NAACL 201.

[12] Durrani, N., Al-Onaizan, Y., Ittycheriah, A., "Improving Egyptian-to-English SMT by Mapping Egyptian into MSA", Springer Berlin Heidelberg, Berlin, Heidelberg, pages 271–282.

[13] Sajjad, H., Durrani, N., Guzman, F., Nakov, P., Abdelali, A., Vogel, S., Salloum, W., ElKholy, A., Habash, N. "Egyptian arabic to english statistical machine translation system for NIST openmt'2015." In *CoRR* abs/1606.05759.

[14] Klementiev, A., Irvine,A., , Callison-Burch, C., Yarowsky, D., "Toward statistical machine translation without parallel corpora.", In Proceedings of EACL '12.

[15] Saluja, A., Hassan, H., Toutanova, K., Quirk, C., "Graph-based semi-supervised learning of translation models from monolingual data.", In ACL 2014

[16] Rico Sennrich, R. , Haddow, B., Birch, A., " Improving neural machine translation models with monolingual data.", In ACL 2016.

[18] Sam T Roweis and Lawrence K Saul. "Nonlinear dimensionality reduction by locally linear embedding.", In Science, 2000

[19] Indyk, P., and Motwani ,R., "Approximate nearest neighbors: towards removing the curse of dimensionality." In *Proceedings of the thirtieth annual ACM symposium on Theory, of computing*.

[20] Hyvönen, V., Pitkänen, T., Tasoulis, T., Jaasaari, E., Tuomainen, R., Wang, L., Corander, J. Roos, T., "Fast nearest neighbor search through sparse random projections and voting", In 2016 IEEE International Conference on Big Data, BigData.

[21] Kingma, D., and Ba, J., "Adam: A method for stochastic optimization", http://arxiv.org/abs/1412.6980

[22] Firat, O., Cho, K., Bengio, Y., "Multi-way, multilingual neural machine translation with a shared attention mechanism" ,HLT-NAACL 2016.

[23] Freitag, M. and Al-Onaizan, Y., "Fast domain adaptation for neural machine translation", http://arxiv.org/abs/1612.06897

[25] Luong, M., Pham, H., Manning, D., "Effective approaches to attention-based neural machine translation", in EMNLP 2015.

[26] Luong, M., Pham, H., Manning, D., "Bilingual word representations with monolingual quality in mind", In Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing.

[27] Jeff Mitchell and Mirella Lapata. 2010. "Composition in distributional models of semantics.", In Cognitive science, 2010.

[29] Rico Sennrich, R., Haddow, B. Birch, A. "Neural machine translation of rare words with subword units.", In ACL 2016.

[31] Ilya Sutskever, I., Vinyals, O., and Le, Q., " Sequence to sequence learning with neural networks.", In Advances in Neural Information Processing Systems 2014

[32] Tiedemann, J., "Parallel data, tools and interfaces in opus.", In Proceedings of (LREC'12)

Toward Robust Neural Machine Translation for Noisy Input Sequences

Matthias Sperber, Jan Niehues, Alex Waibel

Interactive Systems Labs
Karlsruhe Institute of Technology, Germany
matthias.sperber@kit.edu

Abstract

Translating noisy inputs, such as the output of a speech recognizer, is a difficult but important challenge for neural machine translation. One way to increase robustness of neural models is by introducing artificial noise to the training data. In this paper, we experiment with appropriate forms of such noise, exploring a middle ground between general-purpose regularizers and highly task-specific forms of noise induction. We show that with a simple generative noise model, moderate gains can be achieved in translating erroneous speech transcripts, provided that type and amount of noise are properly calibrated. The optimal amount of noise at training time is much smaller than the amount of noise in our test data, indicating limitations due to trainability issues. We note that unlike our baseline model, models trained on noisy data are able to generate outputs of proper length even for noisy inputs, while gradually reducing output length for higher amount of noise, as might also be expected from a human translator. We discuss these findings in details and give suggestions for future work.

1. Introduction

Many natural language processing tasks require applying sequence models on corrupted or noisy input sequences. A typical example is machine translation of erroneous outputs from an automatic speech recognizer (ASR). Ideally, we would like the translation process to ignore or even correct the corrupted inputs. Translation models are usually trained on wellformed parallel sentences that do not exhibit such noise. This results in a harmful mismatch between training and test data, and further aggravates the difficulty of having to transform malformed inputs in the first place. The now prevalent neural sequence-to-sequence models [1, 2, 3] have been identified to be especially sensitive to noisy data [4, 5, 6], and more specifically to corrupted inputs due to erroneous ASR [7].

Robustness at test-time may be improved by inducing suitable forms of noise during the training process. The spectrum of suitable approaches ranges from general-purpose regularizers[1] such as dropout [9] to task-specific approaches that alter the training data to resemble the corrupted inputs at test-time. Task-specific approaches can make stronger assumptions about the data distribution and are potentially more effective or provide additive gains when combined with general-purpose methods. As a disadvantage, they are also more complex and may require task-specific knowledge or resources. Another tradeoff to consider concerns trainability. Neural sequence-to-sequence models are known to suffer from explaining-away effects, where models may learn to generate outputs by relying on the target-side context while ignoring the source-side context [10, 11], especially when the source side provides only a weak or noisy signal. As a result, careful calibration of type and amount of induced noise may be necessary.

Prior work on speech translation attempted inducing task-specific noise by training on actual ASR outputs paired with their correct translations. Unfortunately, such data is scarce, and exploiting it may not be straightforward (see [12] and §4.1; but [13]). Alternatively, it has been proposed to synthesize realistic ASR error patterns and suitable translations thereof, and augment the training data accordingly [14, 15]. However, this approach has not yet been shown to transfer to neural machine translation, and is relatively complex, requiring availability of resources such as pronunciation dictionaries and suitable language models.

In this paper, we seek to improve robustness of a neural machine translation model applied to speech recognition input by exploring tradeoffs between general-purpose and task-specific methods. For this purpose, we introduce a simple noise model that is inspired by the word error rate (WER), which categorizes the common ASR error types into substitutions, insertions, and deletions. Accordingly, our noise model artificially corrupts the source side of a parallel training corpus by randomly introducing substitutions, insertions, or deletions. Our noise model is simpler than the prior approaches [14, 15], but nonetheless effective, and provides a flexible test bed that allows exploring the middle ground between task specificity and generality in the context of neural sequence-to-sequence models. In addition, we discuss preliminary efforts toward refining the noise model to capture more task-specific intuitions similar to these prior ap-

[1] In this paper, we use the notions of good generalization (avoiding overfitting, e.g. via regularization) and robustness (stability w.r.t. noisy data) loosely interchangeably. In fact, both are strongly linked in the sense that in general, good generalization implies robustness [8].

proaches.

We conduct experiments on the Fisher and Callhome Spanish–English speech translation corpus [12] and observe minor improvements in robustness when applying our noise model. We find that increasing the amount of noise during training up to a certain point helps translation of noisy inputs but hurts translation of clean inputs. Strikingly, the optimal amount of noise is much smaller than the amount of noise in our test data, indicating trainability issues. Increasing the amount of noise further leads to a drop in recall but slight increase in precision, leading to the question of to what extent it is desirable from a usability perspective to drop uncertain source-side content as opposed to guessing a translation for it. We conclude with discussing shortcomings of our approach and give suggestions for future work.

2. Related Work

Inducing noise in the training inputs can be seen as a form of data augmentation, which has been used in several applications such as acoustic modeling [16], computer vision [17], language modeling [18], and statistical machine translation where data can be augmented by paraphrases [19]. It has been described as more powerful than general-purpose regularization in the context of deep learning [17]. Note that these approaches aim at inducing *label-preserving* noise, in contrast to our noise model which may alter or destroy the meaning of an input despite keeping targets unchanged. Data augmentation has also been used specifically to improve robustness to noisy inputs as in our work, such as research on speech recognition under noisy conditions [20] and translating spelling mistakes [5, 6]. The latter work demonstrates the importance of using natural (as opposed to synthetic) noise to make models robust to realistic noisy test-time conditions.

Several works have identified noisy or mismatched text inputs as a challenge for neural models: [21] mention domain mismatch as a challenge for neural machine translation, [4] show that NMT suffers from noisy training data, [22] show that recurrent neural networks can be sensitive to corrupted input sequences.

Our approach is methodologically inspired by reward-augmented maximum likelihood (RAML) [23]. We use a similar sampling procedure on the source side, instead of the target side as in RAML. However, RAML is very differently motivated, aiming at fixing exposure bias whereas we are concerned with noise from upstream components. In addition, sampling according to [23]'s approach is biased toward producing less deletions than substitutions and insertions, which our noise model purposefully avoids.

Finally, prior work has dealt with uncertain inputs from upstream components through explicit representation of the uncertainty, for example by directly translating word lattices produced by the speech recognizer [24, 25, 26, 27].

3. Noise Model

This section introduces a noise model that will be applied to every input sentence of the training data. The general idea follows the intuitions behind the WER, according to which ASR errors can be categorized into substitutions, insertions, and deletions. Design goals are flexibility to capture various levels of refinement, and convenient control of the amount of noise and other properties. We first describe the vanilla model, and then present several refinements.

3.1. Vanilla Noise Model

The vanilla noise model, outlined in Algorithm 1, can be summarized as follows. For each sentence, we first decide on the number of edits, while considering the desired amount of overall noise. The edits are then randomly divided into substitutions, insertions and deletions. Finally, for each edit a position is randomly chosen along with a new word for substitutions and deletions.

More formally, let hyperparameter $\tau \in [0, 1]$ denote the amount of noise to be induced, let V be a sampling vocabulary, and assume a sentence of length n as $\langle w_0{=}\text{sos}, w_1, \cdots, w_n, w_{n+1}{=}\text{eos}\rangle$. We first draw the number of edits e (line 1). The Poisson distribution is a suitable choice because it is defined over non-negative integers and has probability mass centered around its mean. For simplicity, we allow a maximum of n edits for a sentence of length n. Thus, we sample according to a n-truncated Poisson distribution [28], defined as $P_\lambda(k) \propto \exp(-\lambda)\frac{\lambda^k}{k!}$ with support $k \in \{0, \cdots, n\}$, where we set $\lambda := \tau \cdot n$. The mean of this distribution is approximately λ. Because of the finite support, this distribution reduces to a categorical distribution and is thus trivial to sample from.

Next, we draw the number of substitutions n_s, number of insertions n_i, and number of deletions n_d such that $n_\text{s}{+}n_\text{i}{+}n_\text{d} = e$ and $n_\text{s}, n_\text{i}, n_\text{d} \in \mathbb{N}^0$ (line 2). This defines a space over $\langle n_\text{s}, n_\text{i}, n_\text{d}\rangle$, known as the discrete 3-simplex [29]. We sample from a uniform distribution over this space (§3.1.1).

We then draw without replacement a position for each substitution, insertion, and deletion (lines $3, 4, 5$). Finally, we corrupt the original sentence accordingly (lines 6 through 16), sampling new words for substitutions and insertions uniformly from the sampling vocabulary (lines 7 and 14).

3.1.1. Sampling from the Discrete Simplex

In order to determine the number of edit operations $n_1, \cdots, n_d$ for each operation type (here: $n_\text{s}, n_\text{i}, n_\text{d}$, corresponding to substitutions, insertions, and deletions), we uniformly sample $\langle n_1, \cdots, n_d\rangle \sim \text{DiscrSimplex}\,(d, e)$ such that $\sum_{i=1}^{d} n_d = e$ and $n_i \in \mathbb{N}^0$. This can be accomplished by slightly adjusting the sampling approach for the continuous simplex [30] to the discrete simplex as follows. Sample auxiliary random variables $x_1, \cdots, x_{d-1}$ uniformly without replacement from $\{1, 2, \cdots, e{+}d{-}1\}$. Let $x_0{=}0, x_d{=}e{+}d$.

Algorithm 1 Vanilla Noise Model.
– given magnitude of noise: $\tau \in [0, 1]$
– given sentence $\langle w_0{=}\text{sos}, w_1, \cdots, w_n, w_{n+1}{=}\text{eos}\rangle$
– given vocabulary V

1: sample distance $e \sim \text{TruncPoisson}(\tau \cdot n, n)$
2: sample $\langle n_\text{s}, n_\text{i}, n_\text{d}\rangle \sim \text{DiscrSimplex}(3, e)$
3: sample substitution positions $s_1, \cdots, s_{n_\text{s}}$ uniformly without replacement from $\{1, \cdots, n\}$
4: sample insertion positions $i_1, \cdots, i_{n_\text{i}}$ uniformly without replacement from $\{0, \cdots, n\}$
5: sample deletion positions $d_1, \cdots, d_{n_\text{d}}$ uniformly without replacement from $\{1, \cdots, n\} \setminus \{s_1, \cdots, s_{n_s}\}$
6: **for** $i \leftarrow 1 \cdots n_\text{s}$ **do**
7: uniformly sample $\tilde{w} \sim V$
8: replace $w_i \leftarrow \tilde{w}$ ▷ substitution
9: **end for**
10: **for** $i \leftarrow 1 \cdots n_\text{d}$ **do**
11: replace $w_i \leftarrow \epsilon$ ▷ deletion
12: **end for**
13: **for** $i \leftarrow n_\text{i} \cdots 1$ **do**
14: uniformly sample $\tilde{w} \sim V$
15: insert $\tilde{w}$ between w_i and w_{i+1} ▷ insertion
16: **end for**

Finally, let $n_i = x_i - x_{i-1} - 1, \forall i \in \{1, 2, \ldots, d\}$. Proof of correctness directly follows argumentation in [30].

3.2. Refinements

The following discusses several simple steps, all aiming at making the sampled noise more similar to the ASR outputs. For more elaborate refinements, we refer to prior work [14, 15].

3.2.1. Sampling Vocabulary: Linguistic Conditioning

The vanilla model draws substitutions and insertions uniformly from the vocabulary (lines 7 and 14), causing a large portion of induced noise to be drawn from the long tail of rarely occurring words. As a more linguistically informed strategy, we can draw from a unigram instead of a uniform distribution over the vocabular, replacing lines 7 and 14 accordingly.

3.2.2. Sampling Vocabulary: Acoustic Conditioning

Preferably, substitutions would be chosen based on acoustic similarity to the original input. Here, we use negative character edit distance as an approximation for acoustic similarity, and sample according to exponentiated distances $p(\tilde{w}|w) \propto \exp(-\text{dist}(w, \tilde{w}))$, replacing lines 7 and 14.

3.2.3. Sampling Positions

ASR tends to err more often for certain types of words than others. For example, shorter tend to be confused more often because these words can suffer from linguistic and acoustic ambiguity. We can model this by substituting or deleting short words more often, again working with an exponentiated distribution $p(\text{pos} = j) \propto \exp(-|w_j|)$ (lines 3 and 5).

3.2.4. Proportion of Error Types

ASR usually produces more substitutions than insertions and deletions. We may wish to reflect this in our noise distribution, for example by drawing edit operations from a 7-simplex and assigning 1 bucket to insertions, 1 bucket to deletions, and 5 buckets to substitutions[2] (lines 1 and 2).

4. Experiments

We conduct experiments on the Fisher and Callhome Spanish–English speech translation corpus [12], a corpus of Spanish telephone conversations that includes ASR transcripts. The Fisher portion consists of telephone conversations between strangers, while the Callhome portion contains telephone conversations between relatives or friends. The training data size of Fisher/Train is 138,819 sentences, we do not make use of the much smaller Callhome/Train part of the corpus. We use Fisher/Dev as held-out testing data for most of our experiments, which has a WER of 41.3%. The relatively high WER is due to the spontaneous speaking style and challenging acoustics. It should also be noted that the ASR model used by [12] is slightly outdated by now and better WER are achieved with recent advancements [31, 32]. Here, our main concern is handling of noisy inputs, not achieving the most competitive end-to-end BLEU scores.

For preprocessing, we tokenized and lowercased source and target sides. We removed punctuation from the reference transcripts on the source side for consistency with the automatic transcripts which also do not contain punctuation. Although punctuation is removed, we use the manual segmentation as given in the corpus, and leave dealing with noisy segmentation boundaries to future work. Our source-side vocabulary contains all words from the automatic transcripts for Fisher/Train, replacing singletons by an unknown word token, totaling 14,648 words. Similarly, on the target side we used all words from the reference translations of Fisher/Train, replacing singletons by the unknown word, yielding 10,800 words in total.

Our implementation uses the eXtensible Neural Machine Translation (XNMT) toolkit,[3] which is based on DyNet [33]. We use a standard attentional encoder-decoder architecture with one encoder and decoder layer. The encoder is a bidirectional LSTM with 256 hidden units per direction, the decoder is an LSTM with 512 hidden units. We used 128-dimensional word embeddings. We use variational dropout [34] in encoder and decoder LSTMs (p=0.5). To obtain a more noise-

[2] This particular choice of distribution is motivated by our experimental data containing about 5 times as many substitutions as insertions or deletions.

[3] github.com/neulab/xnmt

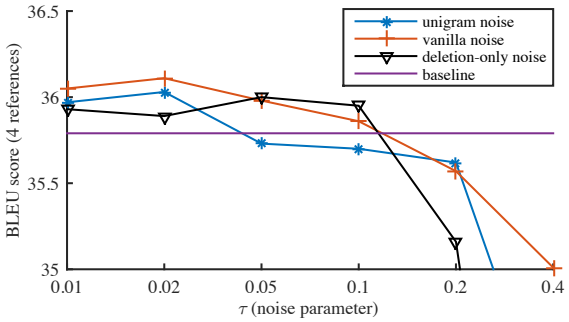


Figure 1: BLEU scores (4 references) on Fisher/Dev, using ASR transcripts as inputs, varying the amount of induced noise.

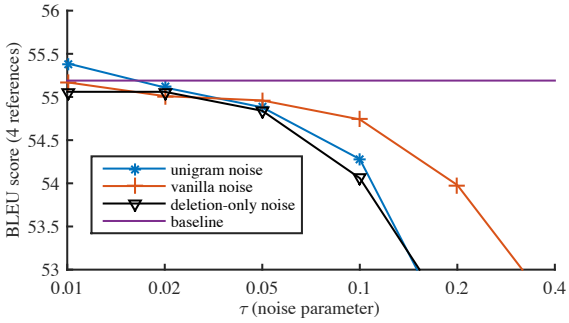


Figure 2: BLEU scores (4 references) on Fisher/Dev, using clean reference transcripts as inputs, varying the amount of induced noise.

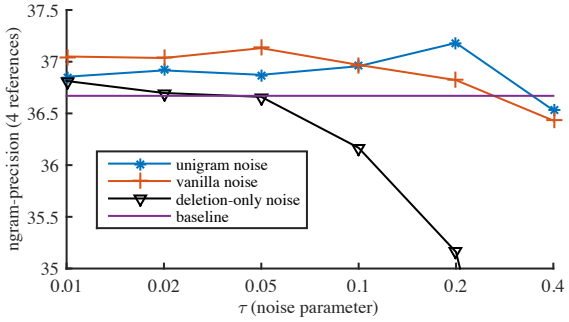


Figure 3: ngram precision (BLEU without brevity penalty) on Fisher/Dev, using ASR transcripts as inputs, varying the amount of induced noise.

robust baseline, we also apply word type dropout [34] to the source word embeddings (p=0.1).

Training was performed with Adam [35]. For all experiments, we first pretrained a model using reference transcripts only, starting from an initial learning rate of 0.0003, restarting Adam and halving learning rates when perplexities did not improve for 2 consecutive epochs [36]. We then fine-tuned the model weights by training on noisy data according to the proposed noise model. Fine-tuning used an initial learning rate of 0.00001 and the same learning rate decay and restarting strategy as during pre-training. The pretraining-finetuning scheme was used in part to make experimental effort manageable, and in part because we observed better BLEU scores in preliminary experiments.

4.1. Main Results

Figure 1 compares our baseline model against several models trained using our noise model. VANILLA NOISE induces varying amounts of noise using the basic model and yields substantial improvements over the BASELINE, which is trained only on clean data. UNIGRAM NOISE replaces the uniform sampling distribution with a unigram distribution and yields similar gains. Perhaps surprisingly, DELETION-ONLY NOISE, a simplified model that induces only deletions, produces strong results as well. We present a possible explanation later. Note that improvements are achieved only for small to moderate amounts of noise. For $\tau = 0.4$, which is close to the WER of the test data, results are rather poor. This indicates that we are facing a trade-off between better trainability for small values for τ, and better distributional similarity with the test data for higher values for τ. We also trained a model by fine-tuning on actual 1-best transcripts rather than using the proposed noise model. Results are rather poor at 32.55 BLEU points, which may be explained by the amount of noise being so high that trainability is compromised, and possibly by some proneness to overfitting because the same noise is used in every epoch.

Figure 2 shows performance of the same models when using clean reference transcripts as inputs. Translation of clean inputs is improved for one configuration of inducing noise, in which case the induced noise can be understood to act as a general-purpose regularizer.[4] However, note that performance drops quickly when increasing the noise parameter τ, again highlighting both the importance of distributional similarity between training and test data, and potential trainability issues.

Figure 3 evaluates models in terms of n-gram precision, which we compute identically to the BLEU score but drop the brevity penalty. Comparing results to Figure 1, we can clearly observe some interactions that lead to trading off precision for recall. Most notably, DELETION-ONLY NOISE performs substantially worse than VANILLA NOISE and UNIGRAM NOISE when measuring only precision. Closer analysis showed that models generally tend to produce shorter outputs the more noise is contained in the inputs. The BLEU metric's brevity penalty is known to punish such short outputs quite severely. DELETION-ONLY NOISE, on the other hand, is trained on inputs where words are deleted. In other words the training-time inputs are shorter than the test-time inputs, counteracting the tendency to produce shorter outputs and thereby avoiding a severe brevity penalty. While this helps BLEU score, arguably producing shorter outputs for

[4]This explanation is supported by prior work relating data noising to traditional smoothing methods [18].

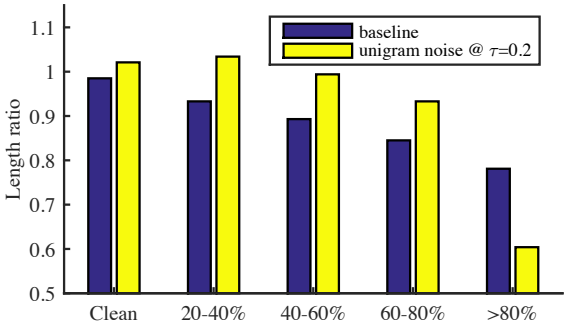


Figure 4: Length ratio of translations when binning test inputs according to their WER.

noisier inputs is a desirable behavior that we would expect also from a human translator, and BLEU may thus not be sufficient as ground for model selection in our task.

4.2. Impact of ASR Quality

For this experiment, we combined all available test data (Fisher/Dev, Fisher/Dev2, Fisher/Test, Callhome/Devtest, Callhome/Evltest), and divided it into bins according to ASR WER. Figure 4 shows the length ratio of translations produced for these inputs for two different models. It can be seen that both BASELINE and UNIGRAM NOISE produce length ratios close to 1.0 for clean inputs. However, when inputs contain even moderate amounts of noise, uncertainty in BASELINE seems to become problematic and outputs quickly become rather short. UNIGRAM NOISE on the other hand appears to handle noisier inputs much more gracefully, while also exhibiting a tendency for shorter outputs when inputs are noisy.

While this demonstrates greater robustness of the noise-induced model, it also raises the question as to what extent shorter outputs for noisy inputs are desirable. Arguably, a human translator may exhibit the same tendency, but further research is required to answer the question of what behavior is desired by a user: dropping uncertain inputs and thus erring on the side of better precision, or trying to guess translations for those inputs anyways and erring on the side of better recall.[5]

4.3. Negative Results for Model Refinements

Our analysis so far only considered the unigram-sampling refinement of the vanilla noise model. We also tested acoustic conditioning (§3.2.2), better sampling positions (§3.2.3), and more realistic proportion of error types (§3.2.4), but did not observe noticeable improvements and do not present details here. Future work may attempt using even more realistic error patterns along the lines of prior work [14, 15]. However, a possible difficulty when trying this may be that, unlike phrase-based machine translation, neural machine translation has been known to be ineffective at learning from rare training examples [11]. Permutations of error patterns potentially consist of mainly such hard-to-learn rarely occurring patterns. Counteracting this by increasing the amount of noise may lead to trainability issues as observed in our experiments as well. Instead, it may be necessary to represent knowledge about confusability more explicitly and efficiently in the model.

[5]Consider a typical example we found in an English ASR transcript, *Boesch as ever his son decides to have a feast*. While the first 3 or 4 words are clearly recognition mistakes (caused by a rare name in the audio), the rest makes sense and a human might choose to only translate the latter part. Another example is *buildings and boundaries around the location very part*, where the last 2 words are easily recognizable as mistakes and could be dropped before translating. However, an experienced translator might guess correctly that *very part* should be replaced by *where to park*. We suggest investigation of desirable translation strategies from a usability perspective for future work.

5. Conclusion

We identified robustness to noisy inputs as a challenge for neural sequence-to-sequence models, and proposed to introduce randomized noise into the training using a simple generative noise model. We found that this improves robustness when properly calibrating type and amount of noise, and that type and amount of noise at training and test time affect the length of the outputs. We highlighted the trade-off between trainability and distributional data similarity, and found that the amount of induced noise must be much smaller than the expected noise at test time for good results. Future work may investigate appropriate trade-offs between precision and recall when translating noisy inputs from a user perspective, use our method for different tasks such as translating user-generated content, and experiment with more refined types of noise or other ways of modeling acoustic similarity in the context of neural machine translation of ASR outputs.

6. Acknowledgments

We thank Sakriani Sakti and the anonymous reviewers for their valuable feedback that helped improve this work.

7. References

[1] N. Kalchbrenner and P. Blunsom, "Recurrent Continuous Translation Models," in *Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, Washington, USA, 2013, pp. 1700–1709.

[2] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems (NIPS)*, Montréal, Canada, 2014, pp. 3104–3112.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *International Conference on Representation Learning (ICLR)*, San Diego, USA, 2015.

[4] B. Chen, R. Kuhn, G. Foster, C. Cherry, and F. Huang,

"Bilingual Methods for Adaptive Training Data Selection for Machine Translation," in *Association for the Machine Translation in Americas (AMTA)*, 2016.

[5] G. Heigold, G. Neumann, and J. van Genabith, "How Robust Are Character-Based Word Embeddings in Tagging and MT Against Wrod Scramlbing or Randdm Nouse?" *arXiv:1704.04441*, 2017.

[6] Y. Belinkov and Y. Bisk, "Synthetic and Natural Noise Both Break Neural Machine Translation," *arXiv:1711.02173*, 2017.

[7] N. Ruiz, M. A. Di Gangi, N. Bertoldi, and M. Federico, "Assessing the Tolerance of Neural Machine Translation Systems Against Speech Recognition Errors," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Stockholm, Sweden, 2017, pp. 2635–2639.

[8] C. Caramanis, S. Mannor, and H. Xu, "Robust Optimization in Machine Learning," in *Optimization for Machine Learning*. The MIT Press, 2011.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[10] L. Yu, P. Blunsom, C. Dyer, E. Grefenstette, and T. Kocisky, "The Neural Noisy Channel," in *International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[11] P. Koehn, "Neural Machine Translation," in *Statistical Machine Translation*, 2017, ch. 13.

[12] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, "Improved Speech-to-Text Translation with the Fisher and Callhome Spanish–English Speech Translation Corpus," in *International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[13] P.-J. Chen, I.-H. Hsu, Y.-Y. Huang, and H.-Y. Lee, "Mitigating the Impact of Speech Recognition Errors on Chatbot using Sequence-to-sequence Model," *arXiv:1709.07862*, 2017.

[14] Y. Tsvetkov, F. Metze, and C. Dyer, "Augmenting translation models with simulated acoustic confusions for improved spoken language translation," in *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Gothenburg, Sweden, 2014, pp. 616–625.

[15] N. Ruiz, Q. Gao, W. Lewis, and M. Federico, "Adapting Machine Translation Models toward Misrecognized Speech with Text-to-Speech Pronunciation Rules and Acoustic Confusability," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Dresden, Germany, 2015, pp. 2247–2251.

[16] N. Jaitly and G. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in *International Conference on Machine Learning (ICML) Workshop on Deep Learning for Audio, Speech, and Language Processing*, Atlanta, USA, 2013.

[17] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[18] Z. Xie, S. I. Wang, J. Li, D. Lévy, A. Nie, D. Jurafsky, and A. Y. Ng, "Data Noising as Smoothing in Neural Network Language Models," in *International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[19] C. Callison-Burch, P. Koehn, and M. Osborne, "Improved statistical machine translation using paraphrases," in *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, New York City, USA, 2006, pp. 17–24.

[20] T. Ishii, H. Komiyama, T. Shinozaki, Y. Horiuchi, and S. Kuroiwa, "Reverberant speech recognition based on denoising autoencoder," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Lyon, France, 2013, pp. 3512–3516.

[21] P. Koehn and R. Knowles, "Six Challenges for Neural Machine Translation," *arXiv:1706.03872*, 2017.

[22] J. Li, W. Monroe, and D. Jurafsky, "Understanding Neural Networks through Representation Erasure," *arXiv:1612.08220*, 2017.

[23] M. Norouzi, S. Bengio, Z. Chen, N. Jaitly, M. Schuster, Y. Wu, and D. Schuurmans, "Reward Augmented Maximum Likelihood for Neural Structured Prediction," in *Neural Information Processing Systems Conference (NIPS)*, Barcelona, Spain, 2016, pp. 1723–1731.

[24] H. Ney, "Speech Translation: Coupling of Recognition and Translation," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Phoenix, USA, 1999, pp. 517–520.

[25] F. Casacuberta, H. Ney, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. Garcia-Varea, D. Llorens, C. Martinez, S. Molau, F. Nevado, M. Pastor, D. Picco, A. Sanchis, and C. Tillmann, "Some approaches to statistical and finite-state speech-to-speech translation," *Com-*

puter Speech and Language, vol. 18, no. 1, pp. 25–47, 2004.

[26] E. Matusov, B. Hoffmeister, and H. Ney, "ASR word lattice translation with exhaustive reordering is possible," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Brisbane, Australia, 2008, pp. 2342–2345.

[27] M. Sperber, G. Neubig, J. Niehues, and A. Waibel, "Neural Lattice-to-Sequence Models for Uncertain Inputs," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[28] P. G. Moore, "The Estimation of the Poisson Parameter from a Truncated Distribution," *Biometrika*, vol. 39, no. 3/4, pp. 247–251, 1952.

[29] J. Costello, "On the number of points in regular discrete simplex," *IEEE Transactions on Information Theory*, vol. 17, no. 2, pp. 211–212, 1971.

[30] N. Smith and R. Tromble, "Sampling uniformly from the unit simplex," Johns Hopkins University, Tech. Rep., 2004.

[31] G. Kumar, G. Blackwood, J. Trmal, D. Povey, and S. Khudanpur, "A Coarse-Grained Model for Optimal Coupling of ASR and SMT Systems for Speech Translation," in *Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, 2015, pp. 1902–1907.

[32] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-Sequence Models Can Directly Transcribe Foreign Speech," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Stockholm, Sweden, 2017.

[33] G. Neubig, C. Dyer, Y. Goldberg, A. Matthews, W. Ammar, A. Anastasopoulos, M. Ballesteros, D. Chiang, D. Clothiaux, T. Cohn, K. Duh, M. Faruqui, C. Gan, D. Garrette, Y. Ji, L. Kong, A. Kuncoro, G. Kumar, C. Malaviya, P. Michel, Y. Oda, M. Richardson, N. Saphra, S. Swayamdipta, and P. Yin, "DyNet: The Dynamic Neural Network Toolkit," *arXiv preprint arXiv:1701.03980*, 2017.

[34] Y. Gal and Z. Ghahramani, "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks," in *Neural Information Processing Systems Conference (NIPS)*, Barcelona, Spain, 2016.

[35] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014.

[36] M. Denkowski and G. Neubig, "Stronger Baselines for Trustable Results in Neural Machine Translation," in *The First Workshop on Neural Machine Translation*, Vancouver, Canada, 2017.

Monolingual Embeddings for Low Resourced Neural Machine Translation

Mattia Antonino Di Gangi[1,2], *Marcello Federico*[1]

[1]Fondazione Bruno Kessler, Trento, Italy
[2]ICT Doctoral School - University of Trento, Italy
{digangi,federico}@fbk.eu

Abstract

Neural machine translation (NMT) is the state of the art for machine translation, and it shows the best performance when there is a considerable amount of data available. When only little data exist for a language pair, the model cannot produce good representations for words, particularly for rare words. One common solution consists in reducing data sparsity by segmenting words into sub-words, in order to allow rare words to have shared representations with other words. Taking a different approach, in this paper we present a method to feed an NMT network with word embeddings trained on monolingual data, which are combined with the task-specific embeddings learned at training time. This method can leverage an embedding matrix with a huge number of words, which can therefore extend the word-level vocabulary. Our experiments on two language pairs show good results for the typical low-resourced data scenario (IWSLT in-domain dataset). Our consistent improvements over the baselines represent a positive proof about the possibility to leverage models pre-trained on monolingual data in NMT.

1. Introduction

Neural machine translation [1, 2] has shown to be highly effective in conditions where there is a good quantity of data available, but struggles to provide good results in a low-resource condition. In general, publicly-available parallel data are small in size, containing at most only few millions of parallel sentences. Therefore, it becomes important to increase the quantity of data by using monolingual data, which are always available in a larger quantity.

Improving MT with monolingual data is a long-standing technique from statistical machine translation (SMT) [3]. In that case, target-side monolingual data are used to train a better language model for producing more fluent translations [4], or even to perform domain adaptation [5]. By contrast, there are no effective usages of source-side monolingual data.

In NMT, there is only one model trained end to end instead of several different statistical models that are combined by means of a log-linear function. The end-to-end approach is considered to be the strength point of NMT [6], but it also means that there is no obvious way to use monolingual data. In fact, the most used approach so far consists in augmenting the training set with synthetic parallel data. They are usually back-translations of target monolingual sentences [7], but also forward-translations of the source side [8] or even copies of the target language in the source side [9]. In all the cases, as the synthetic data are mixed with the real data, the number of synthetic sentence pairs should be kept under control to prevent a degradation of performance. This strongly limits the size of usable monolingual data. Other approaches explore different machine learning frameworks for using monolingual data, such as multi-task learning [10] to improve the encoder with source-side monolingual data [11], or reinforcement learning to jointly learn two systems and exploit monolingual data from both sides [12].

In other NLP tasks, unsupervised learning on large data has been extensively used for training continuous representation of words [13, 14] that are used to initialize the embeddings for the task-specific model, or as an input to it. In NMT, there are word embeddings for both source and target side, and they are generally jointly learnt with the rest of the network. As far as we know, for NMT there are no works reporting improvements by initializing the embeddings with embeddings trained on monolingual data. One of the reasons can be that pre-training the embeddings together with the RNNs that combine them [15, 16] was considered a more promising option. A second reason can be found in the tokens granularity in NMT, which is usually at a sub-word level in state-of-the-art systems. By using sub-words, the embeddings should be recomputed every time a different training set is used. Thus, while effective in terms of performance, the subword-level translation precludes the access to additional existing word-level resources. Moreover, the sub-word tokens are more ambiguous than their word-level counterparts, and this can lead to wrong translations that are harder to catch automatically if compared with "unknown" tokens.

In this work, we propose to modify the NMT architecture to take as additional input the embeddings computed on monolingual data, which we call *external*. The external embeddings are merged with the internal embeddings learned during the NMT training in order to achieve an improved word representation. A previous work [17] shows that using external embeddings in a high resource setting harms the performance. Thus, we set the experiments in a low-resource scenario, simulated by taking only in-domain IWSLT [18] data for TED talks. We experiment our method on En↔Fr

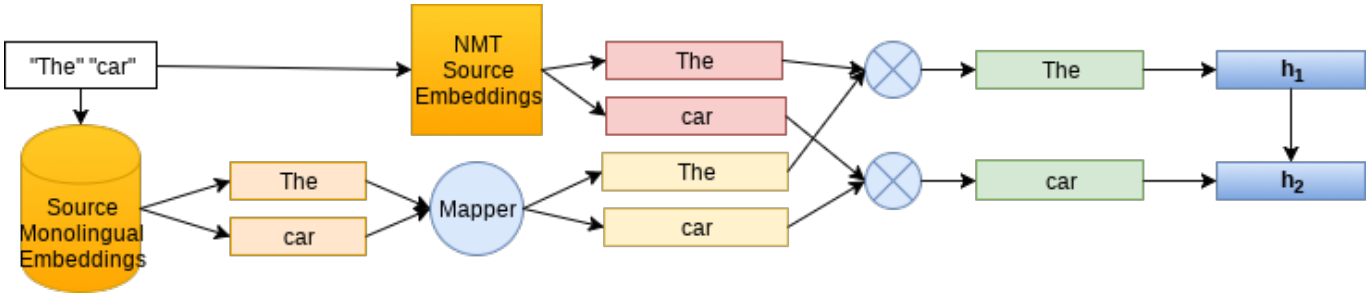


Figure 1: Merging external embeddings with the normal NMT embeddings in the encoder side. The tokens "The" and "car" are used to extract the two kinds of embeddings that are merged before being used as input for the encoder RNN.

and En→De. Our results in all the language directions show significant improvements over the word-level baseline while using only out-domain monolingual data, and comparable results with the BPE baseline that is not limited by the vocabulary size.

The codebase we have used, based on Nematus[1] [19] is available on Github[2].

2. Background

Neural machine translation is based on the attention-based encoder-decoder architecture [2] which jointly learns the translation and alignment models with a sequence-to-sequence process. A sequence of source words $f_1, f_2, \ldots, f_m$ is mapped to sequence of embedding vectors $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_m}$, via a look-up table $X \in R^{|V| \times d}$, where $|V|$ is the vocabulary size and d is the dimensionality of the embedding vectors. Hence, the memory occupied by the vocabulary is linear in both the vocabulary size and the embeddings size.

The embedding sequence is then processed by a bi-directional RNN [20]:

$$\overrightarrow{\mathbf{h}}_j = g(\mathbf{x}_j, \overrightarrow{\mathbf{h}}_{j-1}), \quad j = 1, ..m$$

$$\overleftarrow{\mathbf{h}}_j = g(\mathbf{x}_j, \overleftarrow{\mathbf{h}}_{j+1}), \quad j = m, .., 1$$

where g is the LSTM [21] or the GRU [22] function, and the outputs from the two directions are then concatenated. The sequence of vectors produced by the bidirectional RNN is the encoded representation of the source sentence.

The decoder takes as input the encoder outputs (or states) and produces a sequence of target words $e_1, e_2, \ldots, e_l$. The decoder works by progressively predicting the probability of the next target word e_i given the previously generated target words and the source context vector $\mathbf{c_i}$. At each step, the decoder extracts the word embedding $\mathbf{y}_{i-1}$ of the previous target word, applies one recurrent layer to it, and the output from this layer is used to compute the attention over the source tokens. Finally, the hidden state from the recurrent layers, from the attention output and the word embeddings are combined and then used for computing the normalized probabilities over the target words with a softmax. The recurrent layer produces an hidden state $\mathbf{s}_i$

$$\mathbf{s}_i = g(\mathbf{y}_{i-1}, \mathbf{s}_{i-1}, \mathbf{c}_i)$$

where, g can be computed with one or more LSTM or GRU layers. The output of the RNN is then used by the attention model to weigh the source vectors according to their similarity with it, which is computed as:

$$\alpha_{ij} = \frac{\exp(score(\tilde{\mathbf{s}}_i, \mathbf{h}_j))}{\sum_{k=1}^{m} \exp(score(\tilde{\mathbf{s}}_i, \mathbf{h}_k))}$$

Where $\tilde{\mathbf{s}}_i = GRU(\mathbf{y}_{i-1}, \mathbf{s}_{i-1})$ is a partial computation of the hidden state whose aim is to compute the attention. After this step, the weights are used to compute a weighted average of the encoder outputs, which represents the source context:

$$\mathbf{c}_i = \sum_{j=1}^{m} \alpha_{ij} \mathbf{h}_j$$

The source context vector is then combined with the output of the last RNN layer in a new vector $\mathbf{o}_i$ that is passed as input to the softmax layer to compute the probability for each word in the vocabulary to be the next word, such that:

$$p(e_i = k \mid e_{i-1}, \mathbf{c}_i) \propto \exp(\mathbf{o}_i^\top \mathbf{V}^k)$$

where $\mathbf{V}_k$ is the $k-th$ column of the matrix $\mathbf{V}$, which holds the same size of the target-side embedding matrix, and $\mathbf{o}_i$ is a function of $\mathbf{s}_i$ and $\mathbf{c}_i$. Let Θ be the set of all the network parameters, then the objective of the training is to find parameter values maximizing the likelihood of the training set S, i.e.:

$$\mathbf{\Theta}^* = \text{argmin}_{\mathbf{\Theta}} \sum_{(\mathbf{f},\mathbf{e}) \in S} \sum_{i=1}^{|e|} \log p(e_i | e_{<i}, \mathbf{x}; \mathbf{\Theta})$$

Hence, the network adapts all the parameters together to optimize the loss function.

[1] https://github.com/EdinburghNLP/nematus

[2] https://github.com/mattiadg/NMT-external-embeddings

	EnFr		FrEn		EnDe	
Model	tst2013	tst2014	tst2013	tst2014	tst2013	tst2014
Baseline word-level	31.41	28.26	31.30	29.09	16.51	13.33
Baseline tgt-BPE	/	/	/	/	21.72	18.11
Mix Sum	33.00	29.96	32.50	30.40	**22.40**	18.55
Mix Gate	32.23	29.44	32.76	29.86	21.64	18.54
Mix Ctrl	32.77	30.10	32.98	30.77	22.33	**19.13**
BPE	33.37	**31.01**	**34.09**	**30.81**	22.28	18.72
Mix Ctrl Bi	**33.75**	30.38	33.27	30.65	18.17	15.36
Mix Ctrl Bi BPE	33.58	30.98	32.66	30.72	21.79	18.42

Table 1: Results in terms of BLEU scores for all the language directions. In half of the cases, using subwords is still the best approach. Adding external embeddings in the target side is usually not helpful. The improvements over the word-level baseline are always clear.

3. Related works

The most widespread approach for improving NMT with monolingual data is the use of back-translations for augmenting the training set [7]. Although being used in the state of the art, this approach has limitations in a low-resource scenario for two reasons. The first reason is the need for a good system in the opposite translation direction, which is also low-resources, and the translations quality affects performance of the method [23]. The second reason is the sensitiveness to data of this approach, which makes impossible the use of large quantities of monolingual data.

Zoph et al. [15] investigated the transfer learning from a high-resource language pair (parent) to low-resource language pairs for MT (target), leading to consistent improvements on the target language pairs. This approach, though computationally expensive if the parent system is not already available, is simple but it also does not have any effect outside a low-resource scenario.

Gulcehre et al. [24] were the first who tried to use monolingual data in NMT, by integrating a language model (LM) into the MT model. The model uses only the LM output for the integration, thus monolingual data have no effect in improving the word representations.

Domhan and Hieber [25] proposed to add another recurrent layer without dependencies on the source sentence to the decoder, in order to use target-side monolingual data via multi-task training. Again, the multi-task learning does not affect all the parameters of the network, thus the improvements are limited. In fact, the authors show that back-translations still perform better than their method. Ramachandran et al. [16] propose to pre-train encoder and decoder as two separate language models, hence using monolingual data from both sides. They show that with monolingual data it is possible to improve representations beyond the embeddings, and to improve over back-translations. Our work differs from theirs as we are focusing only on the contribution given by the embeddings, and we use them as an additional input to the network, instead of pre-training it.

4. Using external word embeddings

The method we propose uses word embeddings trained on monolingual data to enrich the representation of words in the case of a low-resource scenario.

Each word in a sentence is used to index a word vector in the NMT word embedding matrices and a word vector from an external matrix trained on monolingual data. From now on, we will refer to the first kind of embeddings as *internal* and to the second as *external*. The internal and external vectors for each word are then merged into a final vector that will be used as input for the following layer. As this method can be applied to both source and target side, the following layer is the GRU both in the encoder and in the decoder. Our method changes the word representations before any other computation on words is performed, thus it could also be used in principle with different sequence-to-sequence architectures.

The external embeddings are learned for a task that is not machine translation, hence we introduce a fully-connected nonlinear layer that allows the network to learn how to map the embeddings into a new space, hopefully more useful for the translation task:

$$\tilde{\mathbf{x}_j} = \tanh(\bar{\mathbf{x}}_j^\top \mathbf{W} + \mathbf{b}) \text{ for j} = 1, \ldots, m$$

The data flow from words to RNN is illustrated in Figure 1.

In this work we investigated three different merge functions with an increasing number of parameters: (1) *mix sum*, (2) *mix controller*, (3) and *mix gate*, which can be used either only in the source side or also in the target side.

In the rest of this section we describe the merge functions we have investigated for combining internal and external embeddings.

4.1. Mix sum

The *mix sum* follows the assumption that the internal and external embeddings have the same importance in the word representation, and the network can learn to obtain complementary information from the two. Consequently, we add a simple element-wise sum between the internal and the exter-

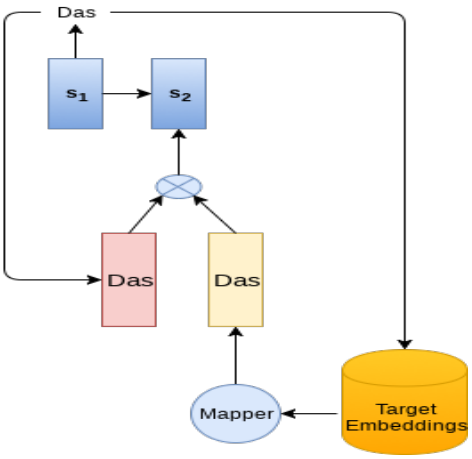


Figure 2: External embeddings in the decoder. As for the internal embeddings, during training the ground-truth word is used, while at translation time it uses the previously translated word. This limits the possibility to use the extended vocabulary of the external embeddings.

nal mapped embedding:

$$\hat{\mathbf{x}}_j = \mathbf{x}_j + \tilde{\mathbf{x}}_j$$

Despite its simplicity, our experiments show that in several cases this function performs comparably to the best function.

4.2. Mix controller

The *mix controller* relaxes the assumption of the same importance for the two embeddings. It is inspired by the *controller function* introduced in [24], and allows us to give a scalar weight to the external embeddings while giving always a weight of 1 to the internal ones. In fact, in our preliminary experiments we obtained some negative results using the external embeddings with large training data, suggesting that in that case the embeddings are better learned from the translation task only.
The first step consists in computing the weight for the external embedding in the range $[0, 1]$, as a function of the embedding itself:

$$w_{ext} = \sigma(\tilde{\mathbf{x}}^\top \mathbf{w_{ctrl}} + b_{ctrl})$$

after the weight has been computed, the two vectors are simply summed:

$$\hat{\mathbf{x}}_j = \mathbf{x}_j + w_{ext}\tilde{\mathbf{x}}_j$$

The controller function is jointly learned with the rest of the network.

4.3. Mix gate

With *mix gate* we want to give the network a finer-grained control over the merging function with respect to the controller. A gate is a vector that modifies the flow of the data by giving weights to each vector component. The gate is computed as a function of a branch of the data flow, which may or may not coincide with the vector to which it is finally applied. All the elements of the gate are in the range $[0, 1]$, and it is applied by element-wise multiplication. Some widely used gated functions are LSTM [21] and GRU [22], but in this work we are inspired by the context gate [26]. The context gate is computed as a function of two inputs and then it is applied to both of them for computing an element-wise weighted average of the two vectors. We apply the gate to the internal and external embeddings:

$$\mathbf{z}_j = \sigma([\mathbf{x}_j; \tilde{\mathbf{x}}_j]^\top \mathbf{W_z} + \mathbf{b_z})$$

where $\mathbf{z}_j$ is the output of the gate and σ is the sigmoid function. The new vector is produced by combining linear transformations of the inputs with the gate $\mathbf{z}_j$:

$$\hat{\mathbf{x}}_j = \tanh(\mathbf{z}_j \odot ff_1(\mathbf{x}_j) + (1 - \mathbf{z}_j) \odot ff_2(\tilde{\mathbf{x}}_j))$$

Where ff is a fully-connected layer. In this setting the network has more parameters to learn for combining the internal and external embeddings in an effective way.

4.4. External embeddings in the target side

In the target side, we investigate the effectiveness of a straightforward extension of the method. At each time step, we merge the external and internal embeddings for the previous word with the same function used in the encoder. But, the softmax can generate only words that are in the internal vocabulary. We have chosen not to use the external vocabulary both for speed reason, as a softmax over a big vocabulary is really expensive, but also to give a priority to the internal embeddings that we consider more relevant for the translation task. But, the main limitation of this approach resides in the difference between training and generation. In fact, during training we know all the target words in advance, and the OOV words that are present in a sentence can still use their "external" representation, if it exists. Hence, during training it is similar to what happens in the source side. By contrast, during the generation phase, when the system produces an unknown token, this will be passed to the next time step and the embeddings for "unknown" will be retrieved from both internal and external matrices.
We are interested in verifying whether the additional information during training, which can modify the unknown token representation in a meaningful way, results in a less frequent generation of unknown tokens, and better sentences in general, during the translation phase [27].

5. Experiments

We have evaluated our method on the IWSLT 2016 [28] datasets for English↔French and English→German. For all the experiments we used an attention-based encoder-decoder with Nematus [19] as a codebase. The encoder is a single-layer bidirectional GRU [20] while the decoder is the con-

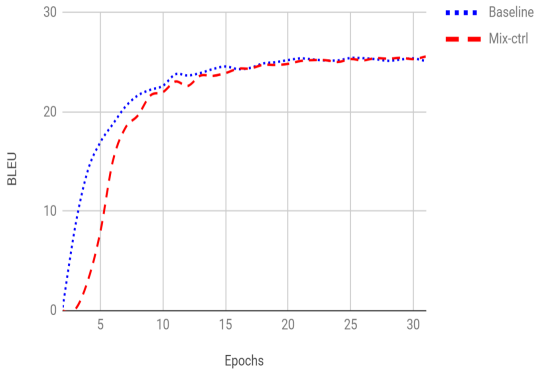


Figure 3: Learning curves on En→De. Without external embeddings the improvement is faster at the beginning, but then it reaches a lower plateau. For readability reasons we inserted only the word-level baseline and the best performing systems with external embeddings.

ditional GRU. We have used embeddings of dimension 500, RNNs with 500 units and GRU [22] activation. As an optimizer we have used Adam [29] with learning rate 0.0003. A dropout of 0.1 is applied to the word indexes and 0.2 to the embedding and hidden vectors.

For the monolingual embeddings, we have used in English the Gigacrawl embeddings available in the GloVe website[3] which has a vocabulary of $1.9M$ words and has been trained on $42B$ tokens. The French and German monolingual embeddings have been computed using fastText [30] on monolingual data, training for 5 epochs with context windows of size 10 and hierarchical softmax as a loss function. For French, we used the publicly-available Gigaword dataset that consists of $2.5B$ tokens, and a vocabulary of $900K$ words. For German, we used the monolingual newscrawl from 2007 to 2017, for a total of $5B$ tokens and a vocabulary of about $4.7M$ words.

The experiments run on En↔Fr and En→De are different among them and are aimed at showing different properties of this method. In fact, with En↔Fr we want to investigate mainly the effectiveness of our approach at a word level, while with En→De we move to a combined approach with BPEs because of the higher inflection of German.

The experimental results are listed in Table 1. For all the language directions we have run a word-level baseline (Baseline word-level) and a BPE baseline (BPE). Then, we have experimental runs using the three merging methods only in the source side. As the mix-ctrl shows better results in general, we have run experiments using this merging method both in source and target sides (Mix Ctrl Bi), and also adding BPE embeddings in German and French (Mix Ctrl Bi BPE). We do not report results by initializing the NMT word embeddings with the external embeddings because we do not observe any significant variation with respect to the word-level baseline.

[3]https://nlp.stanford.edu/projects/glove/

All the translations are evaluated on the de-tokenized and cased output, using the multi-bleu.perl script available in the Moses toolkit [31].

5.1. IWSLT En↔Fr

Our first group of experiments was run on the En↔Fr language pair and is aimed at verifying the improvement given by the external embeddings in a word-level setting.

For both language directions we have trained a word-level baseline using $80K$ words in source and $40K$ in target for En→Fr and $40K$ per language in the opposite direction. In this task we have about $210K$ in-domain (TED talks) parallel sentences.

We compare our systems with a word-level baseline and a BPE baseline. In the En→Fr direction, listed in the first two columns of Table 1 the mix controller and sum are quite comparable, while the gate is clearly worse. Comparing with the word-level baseline we get improvements up to $+1.8$ BLEU points with *mix ctrl* in tst2014. Adding the external embeddings to the decoder improves by another BLEU point for test2013 but the improvement is negligible for 2014, while by using target external embeddings and BPEs in French the improvement is of 0.8 BLEU points in both test sets. This last method produces results comparable with the BPE baseline.

In the Fr→En direction, listed in the two following columns of Table 1, the improvement obtained by the source-side external embeddings is up to $+1.7$ BLEU scores with *mix ctrl*, but adding them in the target side does not provide any significant improvement. For this direction, the BPE system is always the best performing, but in tst2014 is comparable with all the versions of mix-ctrl.

	EnDe		EnFr		FrEn	
Type	2013	2014	2013	2014	2013	2014
Internal	290	391	254	430	538	583
External	147	206	163	275	2715	3300
Both	34	34	57	131	194	200

Table 2: Unknown words in the source side. The external embeddings helps to reduce the unknown words but their number is low from the beginning.

5.2. IWSLT En→De

In En→De the training set consists of about $190K$ parallel sentences. For the increased difficulty of the target language, we introduce the BPE segmentation in the target side. We run a word-level baseline with a vocabulary of $40K$ words per side. The first comparison is with another baseline which uses BPE-segmented words on the target side. Then, we run the three experiments with the external embeddings, and finally a stronger system that uses subwords in both sides. We consider $16K$ merge BPE rules. The baseline using target-side BPEs is from $+4.5$ to $+6$ BLEU points stronger than

Table 3: Unknown words generated by different systems.

	EnDe		EnFr		FrEn	
Type	2013	2014	2013	2014	2013	2014
Word bl	3402	4885	281	301	395	393
Mix ctrl	0	0	449	514	431	463
Mix ctrl bi	2522	3484	466	537	422	445

the word-level baseline (last two columns of Table 1), and adding the external embeddings in the source improves by further $+0.5$ to $+1$ BLEU points. These results are comparable with the ones obtained using BPE segmentation in both source and target side, and our mix-ctrl system obtains the best result in tst2014.
Using external embeddings in the target side together with BPEs produces a deterioration of performance with respect to the mix-ctrl system. If we combine this result with the low number of BPE merging rules (16K), we may suppose that is not possible to learn good embeddings for small sub-words from large monolingual data because of the high ambiguity of each token. But, this hypothesis needs further investigation.

6. Analysis

In this section, we show some phenomena occurring during training and translation with our methods, in order to better understand their impact. In fact, the experiments provided us with results that are definitely stronger than the word-level baseline, but the comparison with BPE needs further investigation.

6.1. Learning curves

A comparison of the learning curves (Fig. 3) shows a big initial advantage for the baseline. The mix controller arrives to similar validation scores only at epoch 10. The mix sum and mix gate systems (not shown) are even slower than mix controller.
Despite the better starting of the word-level baseline, it reaches a plateau faster than the other systems and to a lower score. The curves in Fig. 3 have been computed for En→De, but a similar trend is observed for the other directions, too.
It is interesting to notice how the small score difference in the validation (Fig. 3) becomes much larger in test (Table 1), although the test is performed in a slightly different setting.
After 5 epochs, the mix-ctrl system is not able to produce an intelligible translation, while the word-level system reached the 70% of its quality after the same number of samples. This suggests that the external embeddings are difficult to work with, and we suppose that our merging method, consisting of one single mapper for all the vectors, contributes to slow down the training process.

6.2. Impact of unknown words

In Table 2 we have summarized the number of out-of-vocabulary (OOV) words in the source side. For each test set, we show their number for the internal and external vocabularies, and the number of words that are unknown to both. Although the external vocabulary size is much bigger, as the external embeddings are trained on out-domain data the number of unknown words is higher than in the internal vocabulary. As expected, when the source language is English the number of OOVs is quite small in both vocabularies, but when we use French, it becomes really high in the external vocabulary. This can explain the reduced improvement obtained by the system using our method in Fr→En, where the improvement over the baseline is always less than $+2$ BLEU points.
Now, we focus on the unknown words generated during translations, for which we expected a reduction due to the improved representation. Surprisingly, as it is listed in Table 3, we get more unknown words with our method when we use external embeddings in the source than with the word-level baseline. On the other hand, the contribution of adding them in the target side seems to be language dependent. In fact, it slightly increases in En→Fr and slightly decreases in Fr→En.
In EN→DE, the number of generated UNK tokens is extremely high, and this is the main reason why adding BPEs in the target side greatly increases the BLEU score.
The reported results are computed with the output files containing the "UNK" tokens, but by removing them we get a negligible BLEU score variation. By looking at translation examples (Table 4) we can notice that our approach generates "UNK" when the word-level generates words that are similar to the target, but wrong. This can be combined with the clearer alignment produced by using words instead of sub-words in order to effectively replace these tokens with an effective translation.

6.3. Example Translations

In Table 4 we present some examples of translations to understand what actually happens in our model. In most of the sentences we have read, the translations were basically one the rephrasing of the others, thus the BLEU scores often depend on the number of reference words chosen by the systems, even if the paraphrasing would produce a good translation. Sometimes there are significant differences between the systems, as we can see in the examples.
In the first example, the word-level baseline did not translate "are hearing", which is translated instead by all the other systems, but with a different tense with respect to the reference. Going from mix-ctrl to mix-ctrl-bi, we notice that "de la vingtaine" disappears, thus there is no reference to the "twentysomethings". In this case the BPE system performs worse, maybe because of a wrong segmentation that makes it translate something that is not in the source.

Table 4: Examples of translations

src	but this isn ’t what twentysomethings are hearing .
ref	mais ce n’ est pas ce que les jeunes adultes entendent .
word-level	mais ce n’ est pas ce que les jeunes de la vingtaine .
mix-ctrl	mais ce n’ est pas ce que les jeunes de la vingtaine sont en train d’ entendre .
mix-ctrl Bi	mais ce n’ est pas ce que les jeunes sont en train d’ entendre .
BPE	mais ce n’ est pas ce que les gens de twitymer sont en train d’ entendre .
src	[...] but if we remove this boundary , the only boundary left is our imagination .
ref	[...] mais si on supprime cette limite , la seule qu’ il nous reste est notre imagination .
word-level	[...] mais si nous supprimons cette frontière , la seule frontière à gauche est notre imagination .
mix-ctrl	[...] mais si nous retirons ces limites , la seule frontière gauche est notre imagination .
mix-ctrl Bi	[...] mais si nous retirons cette frontière , la seule frontière est devenue notre imagination .
BPE	[...] mais si nous enlevons cette frontière , la seule frontière reste est notre imagination .
src	Egyptologists have always known the site of Itjtawy was located somewhere near the pyramids of the two kings [...] .
ref	les égyptologues avaient toujours présumé qu’ Itjtawy se trouvait quelque part entre les pyramides des deux rois [...] .
word-level	Nous avons toujours connu le site de Londres , situé quelque part prés des pyramides des deux rois [...]
mix-ctrl	les UNK ont toujours connu le site de la UNK était situé quelque part près des pyramides des deux rois [...].
mix-ctrl Bi	UNK a toujours connu le site de la UNK se situait vers les pyramides des deux rois [...]
BPE	les Egyptologistes ont toujours connu le site de Itjtawy a été situé quelque part près des pyramides des deux rois [...]

In the second example, the baseline chose the wrong meaning of “left”, and the same error is kept by mix-ctrl that also changes “cette limite” to “ces limites”, transforming it into a plural. Mix-ctrl-bi does not have the problem of the plural and adds “est devenue”, which is not a translation of “left”, but produces a nice paraphrasing. The BPE system instead uses a wrong verb tense that results in a non-fluent phrase.
In the third example, we have two words unseen during training, one is “Egyptologists”, the other is “Itjtawy”. Here, the baseline translate the site of “Itjtawy” with “Londres”, the French name for London, while our approaches choose the “UNK” token. The BPE system, instead, is capable to translate it correctly. For what concerns “Egyptologists”, the mix-ctrl system produces the article for the correct person followed by UNK, while the other two word-level approaches chose the wrong person. Compared with the baseline, the mix-ctrl has “Egyptologists” in its source external vocabulary. The BPE system produces an almost perfect translation for that word (the correct form would be “Égyptologistes”), even though it is not the one present in the reference. However, all the systems fail in producing a fluent translation for the whole sentence.
These examples show that the external embeddings can add meaning to the internal word vectors, but there seem to be some nasty interferences among very close word vectors that can lead to wrong translations.

7. Conclusions

We have presented a method for leveraging embeddings trained with an external monolingual tool into NMT. Our method produces consistent improvements over a word-level baseline, and has similar performance with a BPE system, while keeping translation at word-level.

The experimental results show that this approach, though limited, can open the way to a new approach for leveraging monolingual data into NMT, but it needs to go beyond the training of only the embeddings. As a future work we want to explore methods for pre-training larger models with monolingual data and integrate them in NMT for improving the word representations while overcoming the limitations we have highlighted.

8. Acknowledgements

This work has been partially supported by the EC-funded projects ModernMT (H2020 grant agreement no. 645487) and QT21 (H2020 grant agreement no. 645452). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs used for this research.

9. References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[2] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proc. of ICLR 2015*, 2015.

[3] P. Koehn, *Statistical machine translation*. Cambridge University Press, 2009.

[4] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, “Large language models in machine translation,” in *In Proc. of EMNLP*, 2007.

[5] N. Bertoldi and M. Federico, “Domain adaptation for

statistical machine translation with monolingual resources," in *Proc. of WMT09*. Association for Computational Linguistics, 2009, pp. 182–189.

[6] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[7] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. of ACL 2016*, 2016.

[8] J. Park, B. Na, and S. Yoon, "Building a neural machine translation system using only synthetic parallel data," *arXiv preprint arXiv:1704.00253*, 2017.

[9] A. Currey, A. V. M. Barone, and K. Heafield, "Copied monolingual data improves low-resource neural machine translation," in *Proc. of WMT 2017*, 2017, p. 148.

[10] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

[11] J. Zhang and C. Zong, "Exploiting source-side monolingual data in neural machine translation," in *Proc. of EMNLP 2016*, 2016.

[12] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T. Liu, and W.-Y. Ma, "Dual learning for machine translation," in *Advances in Neural Information Processing Systems*, 2016, pp. 820–828.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[14] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[15] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," in *Proc. of EMNLP*, 2016.

[16] P. Ramachandran, P. J. Liu, and Q. V. Le, "Unsupervised pretraining for sequence to sequence learning," in *Proc. of EMNLP*, 2017.

[17] M. A. Di Gangi and M. Federico, "Can monolingual embeddings improve neural machine translation?" in *Proc. of CLiC-it*, 2017.

[18] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proc. of EAMT*, Trento, Italy, May 2012, pp. 261–268.

[19] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. V. M. Barone, J. Mokry, *et al.*, "Nematus: a toolkit for neural machine translation," in *Proc. of EAMT*, 2017.

[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proc. of SSST-8*, 2014.

[23] M. A. Di Gangi, N. Bertoldi, and M. Federico, "FBK's participation to the English-to-German News Translation Task of WMT 2017," in *Proc. of WMT17*, 2017, pp. 271–275.

[24] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[25] T. Domhan and F. Hieber, "Using target-side monolingual data for neural machine translation through multitask learning," in *Proc. of EMNLP*, 2017, pp. 1501–1506.

[26] Z. Tu, Y. Liu, Z. Lu, X. Liu, and H. Li, "Context gates for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 87–99, 2017.

[27] X. Li, J. Zhang, and C. Zong, "Towards zero unknown word in neural machine translation," in *Proceedings of IJCAI*, 2016, pp. 2852–2858.

[28] M. Cettolo, J. Niehues, S. Stker, L. Bentivogli, and M. Federico, "The IWSLT 2016 evaluation campaign," in *Proc. of IWSLT 2016, Seattle, pp. 14, WA*, 2016.

[29] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR*, 2015.

[30] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, pp. 135–146, 2017.

[31] 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 *Proc. of ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 2007, pp. 177–180.

Effective Strategies in Zero-Shot Neural Machine Translation

Thanh-Le Ha, Jan Niehues, Alexander Waibel

Institute for Anthropomatics and Robotics
KIT - Karlsruhe Institute of Technology, Germany
firstname.lastname@kit.edu

Abstract

In this paper, we proposed two strategies which can be applied to a multilingual neural machine translation system in order to better tackle zero-shot scenarios despite not having any parallel corpus. The experiments show that they are effective in terms of both performance and computing resources, especially in multilingual translation of unbalanced data in real zero-resourced condition when they alleviate the language bias problem.

1. Introduction

The newly proposed neural machine translation [1] has shown the best performance in recent machine translation campaigns for several language pair. Being applied to multilingual settings, neural machine translation (NMT) systems have been proved to be benefited from additional information embedded in a common semantic space across languages. However, in the extreme cases where no parallel data is available to train such system, often NMT systems suffer a bad training situation and are incapable to perform adequate translation.

In this work, we point out the underlying problem of current multilingual NMT systems when dealing with zero-resource scenarios. Then we propose two simple strategies to reduce adverse impact of the problem. The strategies need little modifications in the standard NMT framework, yet they are still able to achieve better performance on zero-shot translation tasks with much less training time.

1.1. Neural Machine Translation

In this section, we briefly describe the framework of Neural Machine Translation as a sequence-to-sequence modeling problem following the proposed method of [1].

Given a source sentence $\boldsymbol{x} = (x_1, .., x_i, .., x_I)$ and the corresponding target sentence $\boldsymbol{y} = (y_1, .., y_j, .., y_J)$, the NMT aims to directly model the translation probability of the target sequence:

$$P(\boldsymbol{y}|\boldsymbol{x}) = \prod_{j=1}^{I} P(y_j|y_{<j}, \boldsymbol{x}; \boldsymbol{\theta})$$

[1] proposed an *encoder-attention-decoder* framework to calculate this probability.

A bidirectional recurrent *encoder* reads a word x_i from the source sentence and produces a representation of the sentence in a fixed-length vector $\boldsymbol{h}_i$ concatenated from those of the forward and backward directions:

$$\begin{aligned} \boldsymbol{h}_i &= [\overrightarrow{\boldsymbol{h}}_i, \overleftarrow{\boldsymbol{h}}_i] \\ \overrightarrow{\boldsymbol{h}}_i &= d(\overrightarrow{\boldsymbol{h}}_{i-1}, \boldsymbol{E}_s \bullet \boldsymbol{x}_i) \quad (1) \\ \overleftarrow{\boldsymbol{h}}_i &= d(\overleftarrow{\boldsymbol{h}}_{i+1}, \boldsymbol{E}_s \bullet \boldsymbol{x}_i) \quad (2) \end{aligned}$$

where $\boldsymbol{E}_s$ is the source word embedding matrix to be shared across the source words $x_i \in V_x$, d is the recurrent unit computing the current hidden state of the encoder based on the previous hidden state. $\boldsymbol{h}_i$ is then called an *annotation vector* which encodes the source sentence up to the time i from both forward and backward directions.

Then an *attention mechanism* is set up in order to choose which annotation vectors should contribute to the predicting decision of the next target word. Normally, a relevance score $rel(\boldsymbol{z}_{j-1}, \boldsymbol{h}_i)$ between the previous target word and the annotation vectors is used to calculate the context vector $\boldsymbol{c}_i$:

$$\alpha_{ij} = \frac{\exp(rel(\boldsymbol{z}_{j-1}, \boldsymbol{h}_i))}{\sum_{i'} \exp(rel(\boldsymbol{z}_{j-1}, \boldsymbol{h}_{i'}))}, \ \boldsymbol{c}_j = \sum_i \alpha_{ij} \boldsymbol{h}_i$$

In the other end, a *decoder* recursively generates one target word y_j at a time:

$$P(y_j|y_{<j}, \boldsymbol{x}; \boldsymbol{\theta}) = \frac{\exp(\boldsymbol{z}_j)}{\sum_{k=1}^{|V_y|} \exp(\boldsymbol{z}_k)}$$

Where:

$$\begin{aligned} \boldsymbol{z}_j &= g(\boldsymbol{z}_{j-1}, \boldsymbol{t}_{j-1}, \boldsymbol{c}_j) \\ \boldsymbol{t}_{j-1} &= \boldsymbol{E}_t \bullet \boldsymbol{y}_{j-1} \quad (3) \end{aligned}$$

The mechanism in the decoder is similar to its counterpart in the encoder, excepts that beside the previous hidden state $\boldsymbol{z}_{j-1}$ and target embedding $\boldsymbol{t}_{j-1}$, it also takes the context vector $\boldsymbol{c}_j$ from the attention layer as inputs to calculate the current hidden state $\boldsymbol{z}_j$. The predicted word y_j at time j then can be sampled from a softmax distribution of the hidden state. Basically, a beam search is utilized to generate the output sequence - the translated sentence in this case.

Original corpus		
Source Sentence 1	De	versetzen Sie sich mal in meine Lage !
Target Sentence 1	En	put yourselves in my position .
Source Sentence 2	En	I flew on Air Force Two for eight years .
Target Sentence 2	Nl	ik heb acht jaar lang met de Air Force Two gevlogen .
Preprocessed by [2]		
Source Sentence 1	De	**<en> <en>** *de*_versetzen *de*_Sie *de*_sich *de*_mal *de*_in *de*_meine *de*_Lage *de*_! **<en> <en>**
Target Sentence 1	En	*en*__ *en*_put *en*_yourselves *en*_in *en*_my *en*_position *en*_.
Source Sentence 2	En	**<nl> <nl>** *en*_I *en*_flew *en*_on *en*_Air *en*_Force *en*_Two *en*_for *en*_eight *en*_years *en*_. **<nl> <nl>**
Target Sentence 2	Nl	*nl*__ *nl*_ik *nl*_heb *nl*_acht *nl*_jaar *nl*_lang *nl*_met *nl*_de *nl*_Air *nl*_Force *nl*_Two *nl*_gevlogen *nl*_.
Preprocessed by [3]		
Source Sentence 1	De	**2en** versetzen Sie sich mal in meine Lage !
Target Sentence 1	En	put yourselves in my position .
Source Sentence 2	En	**2nl** I flew on Air Force Two for eight years .
Target Sentence 2	Nl	ik heb acht jaar lang met de Air Force Two gevlogen .

Table 1: Examples of preprocessing steps conducted by [2] and [3].

1.2. Multilingual NMT

State-of-the-art NMT systems have demonstrated that machine translation in many languages can achieve high quality results with large-scale data and sufficient computational power[4, 5]. On the other hand, how to prepare such enormous corpora for low-resourced languages and specific domains has remained a big problem. Especially in zero-resourced condition where we do not possess any bilingual corpus, building a data-driven translation system requires special techniques that can enable some sort of transfer learning. A simple but effective approach called pivot-based machine translation has been developed. The idea of the pivot-based approach is to indirectly learn the translation of the source and target languages through a bridge language. However, this pivot approach is not ideal since it is necessary to build two different translation systems for each language pair in order to perform the bridge translation, hence possibly produces more ambiguities cross languages as well as error-prone to the individual systems.

Recent work has started exploring potential solutions to perform machine translation for multiple language pairs using a single NMT system. One of the most notable differences of NMT compared to the conventional statistical approach is that the source words can be represented in a continuous space in which the semantic regularities are induced automatically. Being applied to multilingual settings, NMT systems have been proved to be benefited from additional information embedded in a common semantic space across languages, thus, by some means they are able to conduct some level of transfer learning.

In this section, we review the related work on constructing a multilingual NMT system involved in translating from several source languages to several target languages. Then we consider a potential application of such a multilingual system on zero-shot scenarios to demonstrate the capability of those systems in extreme low-resourced conditions.

We can essentially divided the work into two directions in applying the current NMT framework for multilingual scenarios. The first direction follows the idea that multilingual training of an NMT system can be seen as a special form of multi-task learning where each encoder is responsible to learn an individual modality's representation and each decoder's mission is to predict labels of a particular task. In such a multilingual system, each task or modality corresponds to a language. In [6], the authors utilizes a multiple encoder-decoder architecture to do multi-task learning, including many-to-many translation, parsing and image captioning. [7] proposed another approach which enable attention-based NMT to multilingual translation. Similar to [6], they use one encoder per source language and one decoder per target language for *many-to-many* translation tasks. Instead of a quadratic number of independent attention layers, however, their NMT system contains only a single, huge attention layer. In order to achieve this, the attention layer need to be provided some sort of aggregation layer between it and the encoders as well as the decoders. It is required to change their architecture to accommodate such a complicated shared attention mechanism.

The work along the second direction also considers multilingual translation as multi-task learning, although the tasks should be the same (i.e. translation) with the same modality (i.e. textual data). The only difference here is whether we decide which components are shared across languages or we let the architecture learns to share what. In [8], the authors developed a general framework to analyze which components should be shared in order to achieve the best multilingual translation system. Other works chose to share every components by grouping all language vocabularies into a large vocabulary, then use a single encoder-decoder NMT system

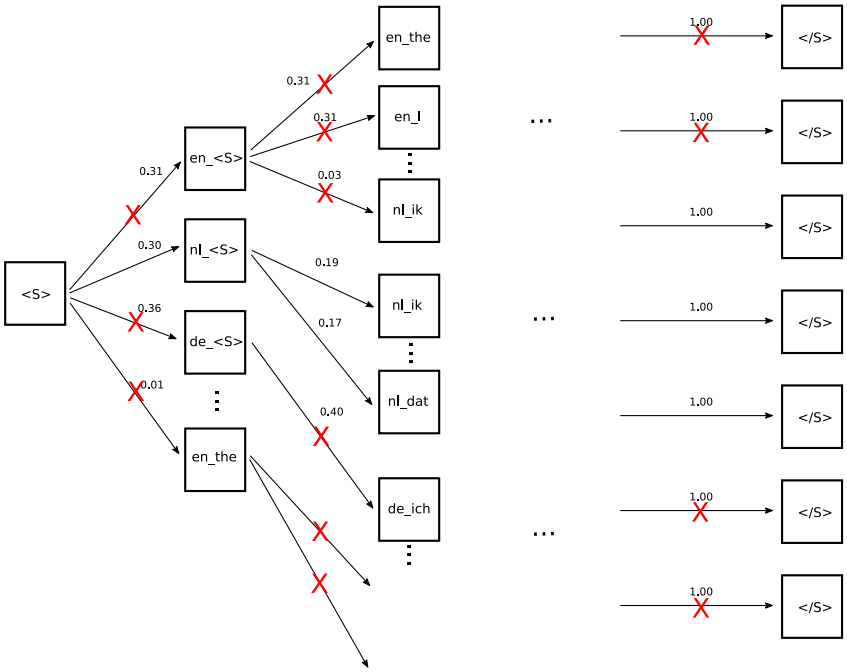


Figure 1: Effect of target dictionary filtering on the decoding process using beam search.

to perform many-to-many translation as each word is viewed as a distinct entry in the large vocabulary regardless of its language. By implementing such mechanism in the preprocessing step, those approaches require little or no modification in the standard NMT architecture. In our previous work[2], we performed a two-step preprocessing:

1. *Language Coding*: Add the language codes to every word in source and target sentences.

2. **Target Forcing**: Add a special token in the beginning of every source sentence indicating the language they want the system to translate the source sentence to.[1]

Concurrently, [3] proposed a similar but simpler approach: they carried out only the second step as in the work of [2]. They expected that there would be only a few cases where two words in difference languages (with different meanings) having the same surface form. Thus, they did not conduct the first step. An interesting side-effect of not doing language-code adding, as [3] suggested, is that their system could accomplish code-switching multilingual translation, i.e. it could translate a sentence containing words in different languages. The main drawback of these approaches is that the sizes of the vocabularies and corpus grow proportionally to the number of languages involved. Hence, a huge amount of time and memory are necessary to train such a multilingual system. Table 1 gives us a simple example illustrating those preprocessing steps.

[1] In fact, we add the target language token both to the beginning and to the end of every source sentence, each place two times, to make the forcing effect stronger. Furthermore, every target sentence starts with a pseudo word playing the role of a start token in a specific target language. This pseudo word is later removed along with sub-word tags in post-processing steps.

2. Multilingual-based Zero-Shot Translation

In this section, we follow the second direction of [2] and [3], hereby called *mix-language* approaches. First we built some baselines inspired of their approaches and participated in the new challenge of zero-shot translation at IWSLT 2017. Then we proposed two strategies, *filtered dictionary* and *language as a word feature*, in attempts to tackle the drawbacks of their approaches. The results in section 3.3 show that our strategies are highly effective in terms of both performance and training resources.

2.1. Target Dictionary Filtering

In [2], the authors discussed about observations of the language bias problem in our multilingual system: If the very first word is wrongly translated into wrong language, the following picked words are more probable in that wrong language again. The problem is more severe when the mixed target vocabulary is unbalanced, due to the language unbalance of the training corpora (whereas the zero-shot is a typical example). We reported a number of 9.7% of the sentences wrongly translated in our basic zero-shot German→French system.

One solution for this problem is to enhance the balance of the corpus by adding *target→target* corpora into the mul-

tilingual system as suggested in [2]. The beam search still need to consider, however, other candidates belonging to the target vocabulary that should not be considered. In this work, we propose a simple yet effective technique to eliminate this bad effect. In the translation process to a specific language, we filter out all the entries in the languages other than that desired language from the target vocabulary. It would significantly reduce the translation time in huge multilingual systems or big texts to be translated due to the fact that many search paths containing the unwanted candidates are removed. More importantly, it assures the translated words and sentences are in the correct language. The effect of this strategy in the decoding process is illustrated in Figure 1.

2.2. Language as a Word Feature

As briefly mentioned in Section 1.2, the main disadvantage of the *mix-language approaches* is the efficiency of training process. Usually in those systems, source and target vocabularies have a huge number of entries, in proportion to the number of languages whose corpora are mixed. It leads to immerse numbers of parameters laying between the embedding and hidden states of the encoder and the decoder. More problematic is the size of the output softmax - where most calculations take place.

There exist works on integrating linguistic information into NMT systems in order to help predict the output words[9, 10, 11]. In those works, the information of a word (e.g. its lemma or its part-of-speech tag) are integrated as a word features. It is conducted simply by learning the feature embeddings instead of the word embeddings. In other words, their system considers a word as a special feature together with other features of itself.

More specially, in the formula 1, 2 and 3, the embedding matrices are the concatenation of all features' embeddings:

$$\boldsymbol{E} \bullet \boldsymbol{x} = \Big[,\Big]_{f \in F} (\boldsymbol{E}^f \bullet \boldsymbol{x}^f)$$

Where $\Big[,\Big]$ is the vector concatenation operation, concatenating the embeddings of individual feature f in a finite, arbitrary set F of word features. The target features of each target word would be jointly predicted along the word. Figure 2 denotes this modified architecture.

Inspired by their work, we attempt to encode the language information directly in the architecture instead of performing language token attachment in the preprocessing step. Being applied in our model, instead of the linguistic information at the word level, our source word features are the language of the considering word and the correct language the target sentence The only target feature is the language of the produced word by the system. For example, when we would like to translate from the sentence "*put yourselves in my position*" into German, the features of each source word would be the word itself, e.g. "*yourselves*", and two additional features "*en*" and "*de*". Similarly, the features of the target words are

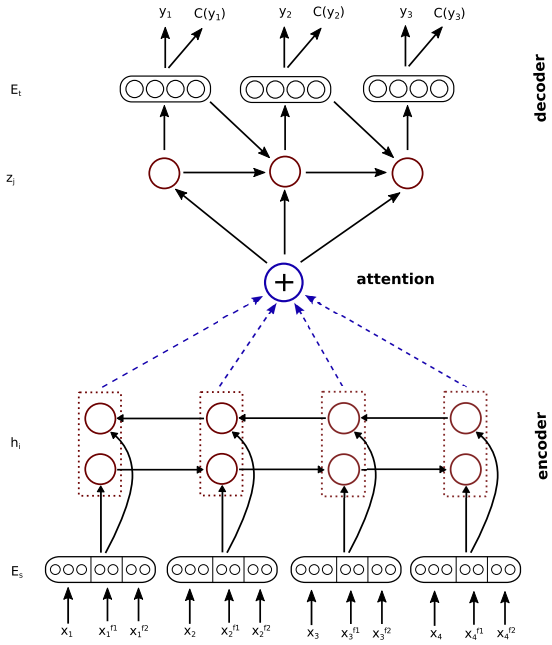


Figure 2: The NMT architecture which allows the integration of linguistic information as word features.

the word and "*de*". This scheme of using language information looks alike to [2], but the difference is the way the language information are integrated into the NMT framework. In [2], those information are implicitly injected into the system. In this work, they are explicitly provided along with the corresponding words. Furthermore, when being used together in the embedding layers, they can share useful information and constraints which would be more helpful in choosing both correct words and language to be translated to. During decoding, the beam search is only conducted on the target words space and not on the target features. When the search is complete, the corresponding features are selected along the search path. In our case, we do not need the output of the target language features excepts for the evaluation of language identification purpose.

3. Evaluation

In this section, we describe a thorough evaluation of the related methods in comparisons with the direct approach as well as the pivot-based approach.

3.1. Experimental Settings

We participated to this year's IWSLT zero-shot tasks for German→Dutch and German→Romanian[12]. The pivot language used in our experiments is English and the parallel corpora are German-English and English-Dutch or German-English and English-Romanian. The data are extracted from

	System	Zero-shot?	German→Dutch		German→Romanian	
			dev2010	tst2010	dev2010	tst2010
(1)	Direct	No	17.83	20.49	12.41	15.14
(2)	Pivot (via English)	Yes	16.11	19.12	12.88	15.04
(3)	Zero 2L [3]	Yes	4.79	5.75	1.55	2.05
(4)	Zero 4L [3]	Yes	6.31	7.93	3.15	3.73
(5)	Zero 6L [2]	Yes	11.58	14.95	8.61	10.83
(6)	Back-Trans [2]	No	17.33	20.36	12.92	15.62

Table 2: Results of the popular *mix-language* methods applied to German→Dutch and German→Romanian zero-shot tasks.

WIT3's[2] TED corpus[13]. The validation and test sets are `dev2010` and `tst2010` which are provided by the IWSLT17 organizers.

We use the Lua version of `OpenNMT`[3][14] framework to conduct all experiments in this paper. Subword segmentation is performed using Byte-Pair Encoding [15] with 40000 merging operations. All sentence pairs in training and validation data which exceeds 50-word length are removed and the rest are shuffled inside each of every minibatch. We use 1024-cell LSTM layers[16] and 1024-dimensional embeddings with dropout of 0.3 at every recurrent layers. The systems are trained using Adam[17]. In decoding process, we use a beam search with the size of 15.

3.2. Baseline Systems

Let us consider the scenario that we would like to translate from a *source* language to a *target* language via a *pivot* language. In order to evaluate the effectiveness of our proposed strategies, we reimplemented the following baseline systems:

- *Direct*: A system which does not exist in the real world is trained using the parallel corpus. It is only for comparison purpose.

- *Pivot*: A system which uses English as the pivot language. The output of the first *source→pivot* translation system was pipelined into the second system trained to translate from *pivot* to *target*.

- *Zero 2L*: To build this system, we followed the idea of [3]: we added a target token to every *source* sentences in the parallel corpus of *source→pivot*, added another target token to every *pivot* sentences in the parallel corpus of *pivot→target*, merged those two parallel corpora into a big corpus and used our standard NMT architecture mentioned in previous section to train and decode. The only differences are the actual data and a simpler NMT architecture we used to train the system.

- *Zero 4L*: Same as *Zero 2L* but in addition applying to two other directions *pivot→source* and *target→pivot*. The result is a parallel corpus two times larger than the corpus in *Zero 2L*.

- *Zero 6L*: This is an extended version of our previous work[2]. There are two main differences compared *Zero 2L* and *Zero 4L*: we conducted both Language Coding and Target Forcing preprocessing steps, the data used to trained are actually six parallel corpora: *source↔pivot*, *pivot→pivot*, *pivot↔target*, *target→target*. Finallly we merged them at the end to form a big parallel corpus.

- *Back-Trans*: This is not a real zero-shot system where we back-translated the English part of the *pivot-target* parallel corpus using a *target-pivot* NMT system. At the end we have a *source-target* parallel corpus with back-translation quality. After we obtained that direct corpus, we apply the same steps as in the *Zero 6L* setting to all corpora we have (8 parallel corpora in total).

3.3. Results

First we applied the baseline systems with respect to the IWSLT17 zero-shot tasks. From Table 2 we can see that in general, translating from German→Romanian is more difficult than German→Dutch, which is reasonable when German and Dutch are considered to be similar. The direct approach which uses a parallel German-*target* corpus and the pivot approach have similar performance in term of BLEU score[18]. Interestingly, the *Back-Trans* performed better that the direct approach on German→Romanian. We spectaculate that back translation might pose some translation noise which makes the translation from German→Romanian more robust.

Compared to the *Zero 6L* model (5), two other Google-inspired models *Zero 2L* (3) and *Zero 4L* (4) from [3] achieved quite low scores. This explains the language-bias problem when these models used less and unbalanced corpora than the *Zero 6L* system. However, the real zero-shot systems (2, 3, 4, 5), excepts the pivot one (2), performed worse than those using direct parallel corpora (1) and (7), since the zero-shot systems have not been shown the direct data, hence, having little or no guide to learn the translation. Among those real zero-shot non-pivot systems, the *Zero 6L*

[2] `https://wit3.fbk.eu/`
[3] `http://opennmt.net/`

	System	Zero-shot?	German→Dutch		German→Romanian	
			dev2010	tst2010	dev2010	tst2010
(1)	Zero 2L [3]	Yes	4.79	5.75	1.55	2.05
(2)	Zero 4L [3]	Yes	6.31	7.93	3.15	3.73
(3)	Zero 6L [2]	Yes	11.58	14.95	8.61	10.83
(3a)	Zero 6L Filtered Dict	Yes	12.50	16.02	9.10	11.00
(3b)	Zero 6L Lang Feature	Yes	13.95	17.15	9.88	11.37
(4)	Back-Trans [2]	No	17.33	20.36	12.92	15.62
(4a)	Back-Trans Filtered Dict	No	17.13	20.22	13.10	15.67
(4b)	Back-Trans Lang Feature	No	17.48	20.24	13.43	15.70

Table 3: Effects of the proposed strategies on performance of zero-shot translation systems

system got the best performance due to the amount and the balance of the data used to train. Thus, from hereinafter we consider the *Zero 6L* as the baseline to analyze the effectiveness of our proposed strategies.

When we applied the proposed strategies, it is interesting to see their effects on different types of systems. Since *Zero 2L* and *Zero 4L* do not have the language identity for words, we cannot directly apply our strategies on those systems. In contrast, it is straight-forward to adapt *Target Dictionary Filtering* and *Language as a Word Feature* on the systems described in [2].

Table 3 shows the performance of our strategies compared to [2] and [3] methods. When we applied the strategies on top of *Back-Trans* system, it seems that the data it used to train is sufficient to avoid the language bias problem. Thus, our strategies did not have a significant effect of performance on this system (4a vs. 4 and 4b vs. 4). But on the real zero-shot configuration (3), both strategies helped to improve the systems by notable margins. On `tst2010`, *Target Dictionary Filtering* (3a) brought an improvement of 1.07 on German→Dutch. On the same test set, *Language as a Word Feature* achieved the gains of 2.20 BLEU scores compared to *Zero 6L* (3b vs. 3). On German→Romanian zero-shot task, the improvements of our strategies were not as great as on German→Dutch, but they still helped, especially on `dev2010`.

Table 5 shows two examples where *Target Dictionary Filtering* clearly improves the quality and readability of the translation over the *Zero 6L* when applied.

Considering the effectiveness of our strategy *Language as a Word Feature* on computation perspective, which is shown in Table 4, we observed very positive results. We compared the *Zero 6L* configuration and our *Language as a Word Feature* system in term of training times, size of source&target vocabularies[4] and the total number of model parameters on both zero-shot translation tasks. The models were usually trained on the same GPU (Nvidia Titan Xp) for 8 epochs so they are fairly compared (seeing the same dataset the same number of times). Each type of models has the same configuration between two zero-shot tasks, excepts the parts related to vocabularies[5].

By encoding the language information into word features, the number of vocabulary entries reduces to almost half of the original method. Thus, it leads to the similar reduction in term of the parameter number. This reduction allows us to use bigger minibatches as well as perform faster updates, resulting in substantially decreased training time (from 7.3 hours to 1.5 hours for each epoch in case of German→Dutch and from 6.0 hours to 1.3 hours for each epoch in case of German→Romanian). The strategy requires minimum modifications in the standard NMT framework, yet it still achieved better performance with much less training time.

German→Dutch			
System	#parameters (millions)	Vocab Size (thousands)	Training Time (hours/epoch)
Zero 6L	243	68	7.3
Lang Feature	130	28	1.5
German→Romanian			
System	#parameters (millions)	Vocab Size (thousands)	Training Time (hours/epoch)
Zero 6L	247	69	6.0
Lang Feature	122	31	1.3

Table 4: Effects of the strategy *Language as a Word Feature* on model size and training time.

4. Conclusion and Future Work

In this paper, we present our experiments toward zero-shot translation tasks using a multilingual Neural Machine Translation framework. We proposed two strategies which substantially improved the multilingual systems in terms of both performance and training resources.

On the future work, we would like to look closer to the outputs of the systems in order to analyze better the effects of our strategies. We also have the plan to expand our strategies on full multilingual systems, for more languages and different data conditions.

[4] In all cases, these sizes are similar numbers.

[5] While the total number of parameters on German→Romanian is bigger than that of German→Dutch, the training time of German→Romanian systems is less due to the fact that its training corpus is smaller.

German→Dutch example	
Zero 6L	Een collega van mij had **toegang** tot **investeringsgegevens** van Fox guard
English meaning	A colleague of mine had **access** to **investment data** of Fox guard
Filtered Dict	Een collega van mij had **Zugang** tot **investment** van de autoriteiten van Fox guard
English meaning	A colleague of mine had **Zugang** to **investment** from the authorities of Fox guard
Reference	Een collega van me kreeg **toegang** tot **investeringsgegevens** van Vanguard
English meaning	A colleague of mine received **access** to **investment data** from Vanguard
German→Romanian example	
Zero 6L	Pentru că s-ar aștepta să apelăm la medic în **nächsten** dimineață .
English meaning	Because he would expect to call a doctor in **nächsten** morning .
Filtered Dict	Pentru că s-ar aștepta să-l chemăm pe doctori în **următorul** dimineață .
English meaning	Because he would expect us to call the doctors the **next** morning .
Reference	Răspunsul e că cei care fac asta se așteaptă ca noi să ne sunăm doctorii în dimineața **următoare** .
English meaning	The answer is that people who do this expect us to call our doctors the **following** morning .

Table 5: Examples of the sentences with the words in wrong languages produced by *Zero* systems and the corrected version produced by the same systems having the target dictionary filtered in decoding phase. Target Dictionary Filtering is not only helpful in producing readable and fluent outputs but also clearly affects to the choices of next words.

5. Acknowledgements

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n° 645452. The research by Thanh-Le Ha was supported by Ministry of Science, Research and the Arts Baden-Württemberg.

6. References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[2] T.-L. Ha, J. Niehues, and A. Waibel, "Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder," *CoRR*, vol. abs/1611.04798, 2016. [Online]. Available: http://arxiv.org/abs/1611.04798

[3] M. Johnson, M. Schuster, Q. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017. [Online]. Available: https://www.transacl.org/ojs/index.php/tacl/article/view/1081

[4] O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, *et al.*, "Findings of the 2016 Conference on Machine Translation (WMT16)," in *Proceedings of the First Conference on Machine Translation (WMT16)*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 12–58.

[5] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, R. Cattoni, and M. Federico, "The IWSLT 2016 Evaluation Campaign," in *Proceedings of the 13th International Workshop on Spoken Language Translation (IWSLT 2016)*, Seattle, WA, USA, 2016.

[6] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," in *International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2016.

[7] O. Firat, K. Cho, and Y. Bengio, "Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism," *CoRR*, vol. abs/1601.01073, 2016. [Online]. Available: http://arxiv.org/abs/1601.01073

[8] N.-Q. Pham, M. Sperber, E. Salesky, T.-L. Ha, J. Niehues, and A. Waibel, "KIT's Multilingual Neural Machine Translation systems for IWSLT 2017," in *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT 2017)*, Tokyo, Japan, 2017.

[9] R. Sennrich and B. Haddow, "Linguistic input features improve neural machine translation," *CoRR*, vol. abs/1606.02892, 2016. [Online]. Available: http://arxiv.org/abs/1606.02892

[10] C. D. V. Hoang, R. Haffari, and T. Cohn, "Improving neural translation models with linguistic factors," in *Proceedings of the Australasian Language Technology Association Workshop 2016*, 2016, pp. 7–14.

[11] J. Niehues, T.-L. Ha, E. Cho, and A. Waibel, "Using factored word representation in neural network language models." in *Proceedings of the First Conference*

on Machine Translation (WMT16). Berlin, Germany: Association for Computational Linguistics, 2016.

[12] M. Cettolo, M. Federico, L. Bentivogli, J. Niehues, S. Stüker, K. Sudoh, K. Yoshino, and C. Federmann, "Overview of the IWSLT 2017 Evaluation Campaign," in *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, 2017.

[13] 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.

[14] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "OpenNMT: Open-Source Toolkit for Neural Machine Translation," *ArXiv e-prints*, 2017.

[15] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Association for Computational Linguistics (ACL 2016)*, Berlin, Germany, August 2016.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[17] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] 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 on Association for Computational Linguistics (ACL 2002)*. Association for Computational Linguistics, 2002, pp. 311–318.

Improving Zero-Shot Translation of Low-Resource Languages

Surafel M. Lakew[1,2], *Quintino F. Lotito* [2,*], *Matteo Negri*[1], *Marco Turchi*[1], *Marcello Federico*[1]

[1]Fondazione Bruno Kessler, Trento, Italy
[2]University of Trento, Trento, Italy
federico@fbk.eu

Abstract

Recent work on multilingual neural machine translation reported competitive performance with respect to bilingual models and surprisingly good performance even on (zero-shot) translation directions not observed at training time. We investigate here a zero-shot translation in a particularly low-resource multilingual setting. We propose a simple iterative training procedure that leverages a duality of translations directly generated by the system for the zero-shot directions. The translations produced by the system (sub-optimal since they contain mixed language from the shared vocabulary), are then used together with the original parallel data to feed and iteratively re-train the multilingual network. Over time, this allows the system to learn from its own generated and increasingly better output. Our approach shows to be effective in improving the two zero-shot directions of our multilingual model. In particular, we observed gains of about 9 BLEU points over a baseline multilingual model and up to 2.08 BLEU over a pivoting mechanism using two bilingual models. Further analysis shows that there is also a slight improvement in the non-zero-shot language directions.

1. Introduction

Machine translation of low-resource languages represents a challenge for neural machine translation (NMT) [1]. Recent efforts in multilingual NMT (Multi-NMT) [2, 3] have shown to improve translation performance in low-resource settings. Multi-NMT models can be trained with parallel corpora of several language pairs to work in *many-to-one, one-to-many, or many-to-many* translation directions. A simple approach, named *target-forcing* [3], is to prepend to the source sentence a tag specifying the target language, both at training and testing time. In addition to performance gains for low-resource languages, the benefit of Multi-NMT is the possibility to perform zero-shot translation, i.e. across directions that were not observed at training time.

Application scenarios in which zero-shot translation can bootstrap the creation of new parallel data – *e.g.* via human post-editing– [2], show how translation performance in the initial zero-shot direction improves over time with the addition of new parallel data. In this work, we explore instead the possibility to enable a trained Multi-NMT model to further learn from its own generated data. Briefly, our method works as follows: first (1), we let the Multi-NMT engine generate zero-shot translations on some portion of the training data; then (2), we re-start the training process on both the generated translations and the original parallel data. We repeat this *training-inference-training* cycle for a few times. Notice that, at each iteration, the original training data is augmented only with the last batch of generated translations. We observe that the generated outputs initially contain a mix of words from the shared vocabulary, but after few iteration they tend to only contain words in the zero-shot target language thus becoming more and more suitable for learning.

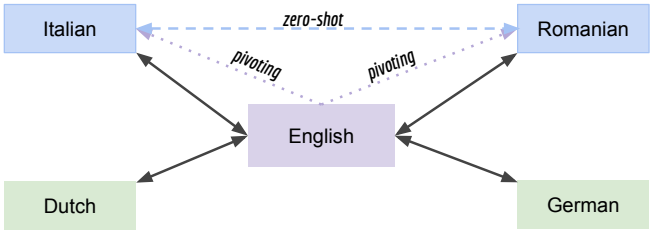


Figure 1: Our zero-shot translation setting for Italian-Romanian. Parallel data is available only for Italian-English, Romanian-English, German-English, and Dutch-English. We leverage multi-lingual neural machine translation trained on all available parallel data to translate across Italian-Romanian (in both directions), either directly (zero-shot) or through English (pivoting).

We test our approach on a Multi-NMT scenario including Italian, Romanian, English, German and Dutch, assuming that the zero-shot translation pair is Italian-Romanian. We also make the assumption that all languages have just parallel data with English (see Figure 1). We apply our approach on top of the multilingual NMT training method suggested by [2]. Experimental results show that our iterative training procedure not only significantly improves performance on the zero-shot directions, but it also boost multilingual NMT in general. Finally, our approach shows to outperform pivot-based machine translation, too.

* Work done during a summer internship at FBK.

2. Related Work

In this section we discuss relevant works on multilingual NMT, zero-shot NMT, and model training with self-generated data, which are closely related to our approach.

2.1. Multilingual NMT

Previous works in Multi-NMT are characterized by the use of separate encoding and/or decoding networks for every translation direction. Dong et al. (2015) [4] proposed a multi-task learning approach for a *one-to-many* translation scenario, based on a sharing representations between related tasks – *i.e* the source language – in order to enhance generalization on the target language. In particular, they used a single encoder in the source side, and separate attention mechanism and decoders for every target language. In a related work [5], used separate encoder and decoder networks for modeling language pairs in a *many-to-many* setting. Notably, they dropped the attention mechanism in favor of a shared vector space where to represent both text and multi-modal information. Aimed at reducing ambiguities at translation time, [6] employed a *multi-source* system that considers two languages on the encoder side and one target language on the decoder side. In particular, the attention model is applied to a combination of the two encoder states. In a *many-to-many* translation scenario, [7] introduced a way to share the attention mechanism across multiple languages. As in [4], but (*only on the decoder side*) and in [5], they used separate encoders and decoders for each source and target language.

Despite the reported improvements, the need of using additional encoder and/or decoder for every language added to the system tells the limitation of these approaches, by making their network complex and expensive to train.

In a very different way, [2] and [3] developed similar Multi-NMT approaches by introducing a *target-forcing* token in the input. The approach in [3] applies a language-specific code to words from different languages in a mixed-language vocabulary. In practice, they force the decoder to translate to a specific target language by prepending and appending an artificial token to the source text. However, their word and sub-word level language-specific coding mechanism significantly increase the input length, which shows to have an impact on the computational cost and performance of NMT [8]. In [2], only one artificial token is prepended to the source sentences in order to specify the target language. Prepending language tokens has permitted to eliminate the need of having separate encoder/decoder networks and attention mechanism for every new language pair.

2.2. Zero-Shot Translation

By extending the approach in [7], zero-resource NMT has been suggested in [9]. The authors proposed a *many-to-one* translation setting and used the idea of generating a pseudo-parallel corpus [10], using a pivot language, to fine tune their model. Moreover, also in this case the need of separate encoders and decoders for every language pair significantly increases the complexity of the model.

An attractive feature of the *target-forcing* mechanism comes from the possibility to perform zero-shot translation with the same multilingual setting as in [2, 3]. However, recent experiments have shown that the mechanism fails to achieve reasonable zero-shot translation performance for low-resource languages [11]. The promising results in [2] and [3] hence require further investigation to verify if their method can work in various language settings, particularly across distant languages.

2.3. Training with self-generated data

Training procedures using self-generated data have been around for a while. For instance, in statistical machine translation (SMT), [12, 13] showed how the output of a translation model can be used iteratively to improve results in a task like post-editing. Mechanisms like back-translating the target side of a single language pair have been used for domain adaptation [14] and more recently by [10] to improve an NMT baseline model. In [15], a dual-learning mechanism is proposed where two NMT models working in the opposite directions provide each other feedback signals that permit them to learn from monolingual data. In a related way, our approach also considers training from monolingual data along dual zero-shot directions. As a difference, however, our *train-infer-train* loop leverages the capability of the network to jointly learn multiple translation directions.

Although our brief survey shows that re-using the output of an MT system for further training and improvement has been successfully applied in different settings, our approach differs from past works in mainly two aspects: *i)* introducing for the first time a *train-infer-train* mechanism addresses Multi-NMT, and *ii)* we cast the approach into a *self-correcting* training procedure over two dual zero-shot directions, so that incrementally improved translations mutually reinforce each direction.

3. Neural Machine Translation

The standard NMT architecture comprises an encoder, a decoder and an attention-mechanism, which are all trained with maximum likelihood in an end-to-end fashion [16]. The encoder is a recurrent neural network (RNN) that encodes a source sentence into a sequence of hidden state vectors. The decoder is another RNN that uses the representation of the encoder to predict words in the target language [8] [17]. As the name suggests, *attention* is a mechanism used to improve the translation quality by deciding which part of the source sentence can contribute mostly in the prediction process [18]. As shown in Figure 2, which simplifies the NMT architecture, first the encoder takes the source words on the left (purple color), maps them to vectors and feeds them into the RNN. When the <eos> (*i.e* end of sentence) symbol is seen, the final time step initializes the decoder RNN (blue

color). At each time step, the attention mechanism is applied over the encoder hidden states and combined with the current hidden state of the decoder to predict the next target word. Then, the prediction is fed back to the encoder RNN to predict the next word, until the <eos> symbol is generated [19].

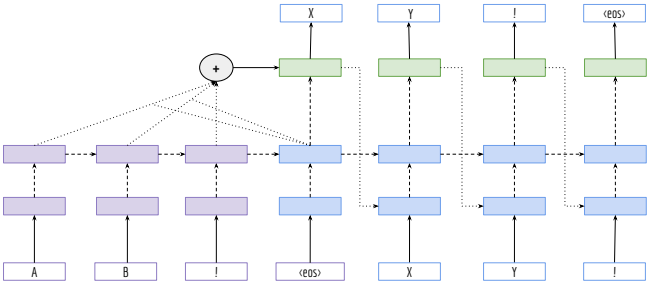


Figure 2: *NMT architecture with encoder-decoder and an attention mechanism, showing a source sentence "A B" translated into a target sentence "X Y".*

4. Mixed Language Input for Multi-NMT

Our goal is to improve translation in the zero-shot directions of a multilingual model with limited directions covered by the training data (see Figure 1). The training strategy of the proposed approach is summarized in Algorithm 1, while its flow chart is illustrated in Figure 3.

To address this problem, our training procedure is performed in three steps which are iterated for several rounds. In the first step (line 2), the multilingual NMT system is trained on the original data available. In the second step (line 5), the trained model is run to translate between the zero-shot directions. Then, in the third step (line 8), the output translations are combined with the corresponding source sentences and added to the original training data. The resulting expanded corpus is now ready to perform a new round of the training process.

According to our *train-infer-train* scheme, new *synthetic* data for the two zero-shot directions are generated at each round. This process creates a *duality* between the two zero-shot translation directions, which we can exploit for mutual improvement. Indeed, for each direction, sub-optimal translations t^* paired with the corresponding original (and well-formed) source s are used to obtain new "parallel" (t^*,s) sentence pairs that extend the training material for the other direction. The translated mixed-input for the two languages is represented as T^*, while the target side T represents the original sentences extracted for inference.

In the Multi-NMT scenario, the sub-optimal translations representing the source element of the new training pairs will likely contain a mixed-language that includes words from a vocabulary shared with other languages. The expectation is that, round after round, the model will generate better outputs by learning at the same time to translate and "correct" its own translations by removing spurious elements from other languages. If this intuition holds true, the iterative improvement will yield increasingly better results in translating between the $source \leftrightarrow target$ zero-shot directions. Ideally, this incremental training and inference cycle can continue until the model converges (line 10).

Algorithm 1: Iterative Learning Procedure

1: TRAIN: $D(src, tgt)$
2: Multi-NMT $\leftarrow$ initial training using dataset D
3: **repeat** INFER-TRAIN
4: **for** $s = 1,T$ **do**
5: $t^* \leftarrow$ inference in duality using Multi-NMT
6: **end for**
7: prepare $D^*([src + T^*], [tgt + T])$
8: Multi-NMT $\leftarrow$ reload Multi-NMT, train using D^*
9: return Multi-NMT
10: **until** Multi-NMT converges $\rightarrow$ Multi-NMT*

Table 1: *Iterative Learning algorithm of the proposed approach using the duality of zero-shot translation directions.*

5. Experiments

All the experiments are carried out using the open source OpenNMT-py[1] toolkit [19]. For training the models, we used the parameters specified in Table 2. Considering the high data sparsity of our low-resource setting, we applied a dropout of 0.3 [20] to prevent overfitting [21]. To train the baseline Multi-NMT, we used Adam [22] as the optimization algorithm with an initial learning rate of 0.001. In the subsequent *train-infer-train* rounds, we used SGD [23], with a learning rate of 1. If the perplexity does not decrease on the validation set or the number of epoch is above 7, a learning rate decay of 0.7 is applied. This combination of optimizers was found to be effective in accelerating the training in the first few iterations. In all the reported experiments the baseline models are trained until convergence, while each train round after the inference stage is assumed to iterate over 10 epochs. For decoding, a beam search of size 10 is applied.

Model parameters	Value
RNN type	LSTM
RNN size	1024
Embedding dim	512
Encoder	bidirectional
Encoder depth	2
Decoder depth	2
Beam size	10
Batch size	128

Table 2: *Hyper-parameters used to train all the models, unless specified in a different setting.*

[1] https://github.com/OpenNMT/OpenNMT-py

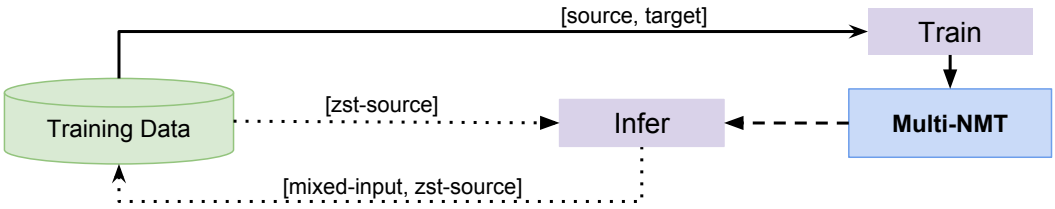


Figure 3: *Illustration of the proposed multilingual* train-infer-train *strategy. Using a standard NMT architecture, a portion of two zero-shot directions monolingual dataset is extracted for inference to construct a dual source↔target mixed-input and continue the training. The top solid line shows the training process, where as the dashed lines show the inference stage*

5.1. Dataset

To evaluate our approach, we consider five languages (*i.e* English (EN), Dutch (NL), German (DE), Italian (IT), and Romanian (RO)). To simulate a low-resource scenario, each language pair has $\approx$ $200k$ parallel sentences (see Table 3 for details). All the parallel datasets are from the IWSLT17[2] multilingual shared task [24].

Direction	Training	test2010	test2017
EN ↔ DE	197,489	1,497	1,138
EN ↔ IT	221,688	1,501	1,147
EN ↔ NL	231,669	1,726	1,181
EN ↔ RO	211,508	1,633	1,129
IT ↔ RO	209,668	1,605	1,127

Table 3: *Number of sentences used to train the multilingual model on eight directions. The IT ↔ RO pairs are used to train only the bilingual models.*

To train all models, we used the same pipeline, first to get a tokenized dataset. Then, we apply byte pair encoding (BPE) [25], using a jointly trained (on source and target dataset) shared BPE model to segment the tokens into sub-word units. For this operation we used $8,000$ BPE merging rules, with a minimum of 30 frequency threshold to apply the segmentation. When training the multilingual models, the pipeline includes adding the artificial language token at the source side of each parallel dataset both for the training and validation sets [2]. We evaluate our models using test2010, and for comparison we use test2017 of the IWSLT2017 evaluation dataset.

5.2. Models

Our baseline models are trained in a multilingual and bilingual settings. For each direction of the multilingual model and every bilingual model we report the BLEU [26] score computed using *multi-bleu.perl*[3] from the Moses SMT implementation. BLEU scores of the Multi-NMT systems trained on the parallel data in Table 3 are reported in Table 6 and 7 (second column). To compare our zero-shot translations against those of the bilingual models we trained two Italian↔Romanian models. Both bilingual are trained with the same amount of training data used by each direction of the Multi-NMT model (see Table 3). Moreover, as additional terms of comparison, we trained two pivoting-based systems (using English as a pivot language): Italian→English→Romanian and Romanian→English→Italian.

5.2.1. Bilingual models

The baseline models for comparison consist of: *i)* an eight direction multilingual model (Multi-NMT), and two bilingual NMT models.

System	tst2010	tst2017
Italian→Romanian	19.66	19.14
Romanian→Italian	22.44	20.69

Table 4: BLEU scores of two bilingual NMT models (Italian→Romanian and Romanian→Italian) on IWSLT data *tst2010* and *tst2017*

The results of the two bilingual models are shown in Table 4. From the Multi-NMT model (see row 9 and 10 of Table 6 and Table 7), we particularly focus on the performance of the zero-shot directions that can be compared with the results from these two models.

5.2.2. Pivoting

If data are available, the pivoting strategy is the most intuitive way to accomplish zero-shot translation, or to translate from/into under-resourced languages through high resource ones [27]. However, results in the pivoting framework are strictly bounded to the performance of the two combined translation engines, and especially to that of the weaker one. In contrast, Multi-NMT models that leverage knowledge acquired from data for different language combinations (similar to multi-task learning) can potentially compete or even outperform the pivoting ones. Checking this possibility is the motivation for our comparison between the two approaches.

In our experiment we take English as the bridge language between Italian and Romanian in both translation directions.

[2]https://sites.google.com/site/iwsltevaluation2017/
[3]http://www.statmt.org/moses

System	tst2010	tst2017
Italian→Romanian	16.4	15.00
Romanian→Italian	18.9	17.36

Table 5: Performance of the Italian↔Romanian pivot translation directions using English as a pivot on *tst2010* and *tst2017*

Unsurprisingly, compared with those of the bilingual models trained on Italian↔Romanian data, the results shown in Table 5 are lower.

On both translation directions, the bilingual models are indeed about 3.0 BLEU points better. Such comparison, however, is not the main point of our experiment, instead, we aim to fairly analyze performance differences between pivoting and zero-shot methods trained in the same condition which lacks Italian↔Romanian training data.

5.3. Zero-shot results

In this experiment, we show how our approach helps to improve the baseline Multi-NMT model. The *train-infer-train* procedure described in Section 4 was applied for five rounds, where each round consists of 10 iterations. Table 6, shows the improvement on the Italian↔Romanian zero-shot directions using the Multi-NMT* model. Specifically, the Italian→Romanian direction reached 17.38 BLEU score improving over the baseline (8.59) by 8.79 points. Romanian→Italian translation improved with an even larger margin (+10.71) from 8.65 to 19.36 BLEU score.

	Multi-NMT	Multi-NMT*
English→Italian	27.07	**28.47**
Italian→English	32.12	**33.16**
English→Romanian	24.65	**25.37**
Romanian→English	32.7	**34.00**
English→German	26.39	**26.42**
German→English	31.3	**31.79**
English→Dutch	30.27	**30.85**
Dutch→English	35.13	**35.77**
Italian→Romanian	8.59	**17.38**
Romanian→Italian	8.65	**19.36**

Table 6: Comparison on *test2010* set between a baseline Multi-NMT model against a Multi-NMT* model with our proposed *train-infer-train* approach for the Italian↔Romanian zero-shot direction. The best result for each direction is shown in bold.

In addition, and to our great surprise, the results from our self-correcting mechanism showed to perform even better than the pivoting strategy. To check the validity of our results, we also compared the baseline multilingual system

	Multi-NMT	Multi-NMT*
English→Italian	29.02	**30.43**
Italian→English	32.87	**33.61**
English→Romanian	20.96	**21.94**
Romanian→English	27.48	**28.21**
English→German	19.75	**19.85**
German→English	24.12	**24.25**
English→Dutch	25.37	**26.12**
Dutch→English	**29.25**	29.15
Italian→Romanian	8.18	**17.08**
Romanian→Italian	8.58	**19.25**

Table 7: Comparison on *test2017* set between a baseline Multi-NMT model against a Multi-NMT* model with our proposed *train-infer-train* approach for the Italian↔Romanian zero-shot direction.

and our approach on the IWSLT 2017 test set (*test2017*). As shown in Table 7, the results confirm those computed on *test2010*, with almost identical gains (+8.9 and +10.67). The other important advantage of our approach is evidenced by the performance gains obtained on the language directions supported by parallel training corpora. To put this into perspective, all translation directions have shown improvements, except for the slight drop (-0.10 BLEU) observed for the Dutch→English direction in *test2017* case.

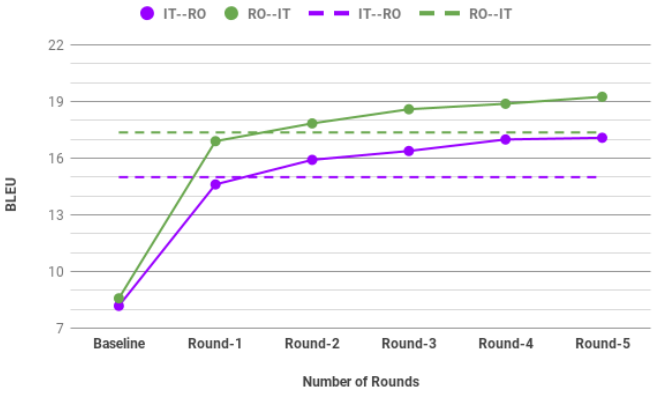


Figure 4: *Results from test2017 for the Italian↔Romanian zero-shot directions, comparing our iterative learning approach (solid lines) with the pivoting mechanism (dashed lines)*

Comparing the results from every rounds (see Figure 4), we observe that the training after the first inference step is responsible for the largest portion of the overall gain. This is mainly due to the initial introduction of (noisy) parallel data for the zero-shot directions. The contribution of the self-correcting process can be seen in the following rounds, i.e the improvement after each inference suggests that the generated data are getting cleaner and cleaner.

	Italian→Romanian
Source	... che rafforza la corruzione, l'evasione fiscale, la povertà, l'instabilità.
Pivot	... poarta de bază, evazia fiscală, sărăcia, instabilitatea.
Multi-NMT	... restrânge corupția, **fiscale de evasion, poverty, instabilitate**.
Multi-NMT*	... care rafinează corupția, **evasarea fiscală, sărăcia, instabilitatea**.
Reference	... care protejează corupţia, evaziunea fiscală, sărăcia şi instabilitatea.
	Romanian→Italian
Source	E o poveste incredibilă.
Pivot	È una storia incredibile
Multi-NMT	È una storia **incredible**.
Multi-NMT*	È una storia **incredibile**
Reference	È una storia incredibile .
	English→Italian
Source	We can't use them to make simple images of things out in the Universe.
Multi-NMT	Non possiamo usarli **per creare immagini semplici di** cose nell'universo.
Multi-NMT*	Non possiamo usarle **per fare semplici immagini di** cose nell'universo.
Reference	Non possiamo usarle per fare semplici immagini di cose nell'univero

Table 8: Top two examples: zero-shot translations generated by pivoting via English, multilingual translation(Multi-NMT) and multilingual translation enhanced with out approach (Multi-NMT*). Last example: multilingual and enhanced multi-lingual translation in a resourced translation direction.

Looking at the sample translation outputs using the different approaches in Table 8, we observe that the baseline Multi-NMT system produces mixed language output (*e.g.* "poverty" in Italian→Romanian and "incredible" in Romanian→Italian).Thanks to our approach, the Multi-NMT* system instead tends to produce more consistent target language ("poverty" becomes "sărăcia" in Italian→Romanian and "incredible" becomes "incredibile" in Romanian→Italian). Furthermore, even in the non-zero-shot directions there are case where the enhanced Multi-NMT* system produces better translations (see the last reported example).

6. Conclusions

We introduced a method to improve zero-shot translation in multilingal NMT under particularly resource-scarce training conditions. The proposed self-correcting procedure, by leveraging syntentic dual translations, achieved significant improvements over a multilingual NMT baseline and outperformed a pivoting NMT approach for the Italian-Romanian directions.

In future work, we plan to improve the train-infer-train stages to reach better performance in less time and with lower training complexity. In our current setup we did not consider any form of selection for the dataset to be translated at the inference stage of the train-infer-train procedure. We expect that applying frequency and similarity based approaches to select promising training candidates can bring further improvements. Moreover, we plan also to test our approach with additional monolingual data from the two zero-shot directions. Finally we plan to extend our approach to the translation of mixed language sentences (i.e code-mixing).

7. Acknowledgements

This work has been partially supported by the EC-funded projects ModernMT (H2020 grant agreement no. 645487) and QT21 (H2020 grant agreement no. 645452). This work was also supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1 and by a donation of Azure credits by Microsoft. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

8. References

[1] P. Koehn and R. Knowles, "Six challenges for neural machine translation," *arXiv preprint arXiv:1706.03872*, 2017.

[2] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *arXiv preprint arXiv:1611.04558*, 2016.

[3] T.-L. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," *arXiv preprint arXiv:1611.04798*, 2016.

[4] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-task learning for multiple language translation." in *ACL (1)*, 2015, pp. 1723–1732.

[5] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, "Multi-task sequence to sequence learning," *arXiv preprint arXiv:1511.06114*, 2015.

[6] B. Zoph and K. Knight, "Multi-source neural translation," *arXiv preprint arXiv:1601.00710*, 2016.

[7] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," *arXiv preprint arXiv:1601.01073*, 2016.

[8] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[9] O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Y. Vural, and K. Cho, "Zero-resource translation with multilingual neural machine translation," *arXiv preprint arXiv:1606.04164*, 2016.

[10] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[11] S. M. Lakew, A. D. G. Mattia, and F. Marcello, "Multilingual neural machine translation for low resource languages," in *CLiC-it 2017 – 4th Italian Conference on Computational Linguistics, to appear*, 2017.

[12] K. Oflazer and I. D. El-Kahlout, "Exploring different representational units in english-to-turkish statistical machine translation," in *Proceedings of the Second Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2007, pp. 25–32.

[13] H. Béchara, Y. Ma, and J. van Genabith, "Statistical post-editing for a statistical mt system," in *MT Summit*, vol. 13, 2011, pp. 308–315.

[14] N. Bertoldi and M. Federico, "Domain adaptation for statistical machine translation with monolingual resources," in *Proceedings of the fourth workshop on statistical machine translation*. Association for Computational Linguistics, 2009, pp. 182–189.

[15] Y. Xia, D. He, T. Qin, L. Wang, N. Yu, T. Liu, and W. Ma, "Dual learning for machine translation," *CoRR*, vol. abs/1611.00179, 2016.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[18] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[19] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:1701.02810*, 2017.

[20] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[21] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, *et al.*, "Large scale distributed deep networks," in *Advances in neural information processing systems*, 2012, pp. 1223–1231.

[24] M. Cettolo, C. Girardi, and M. Federico, "Wit3: Web inventory of transcribed and translated talks," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, vol. 261, 2012, p. 268.

[25] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[26] 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 on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[27] H. Wu and H. Wang, "Pivot language approach for phrase-based statistical machine translation," *Machine Translation*, vol. 21, no. 3, pp. 165–181, 2007.

Evolution Strategy Based Automatic Tuning of Neural Machine Translation Systems

Hao Qin[1], *Takahiro Shinozaki*[1], *Kevin Duh*[2]

[1]Tokyo Institute of Technology, Japan
[2]Johns Hopkins University, USA

qin.h.aa@m.titech.ac.jp, shinot@ict.e.titech.ac.jp, kevinduh@cs.jhu.edu

Abstract

Neural machine translation (NMT) systems have demonstrated promising results in recent years. However, non-trivial amounts of manual effort are required for tuning network architectures, training configurations, and preprocessing settings such as byte pair encoding (BPE). In this study, we propose an evolution strategy based automatic tuning method for NMT. In particular, we apply the covariance matrix adaptation-evolution strategy (CMA-ES), and investigate a Pareto-based multi-objective CMA-ES to optimize the translation performance and computational time jointly. Experimental results show that the proposed method automatically finds NMT systems that outperform the initial manual setting.

1. Introduction

Neural machine translation (NMT) is a new approach to translation and has shown promising results in recent years. Many active ongoing research are focused on developing new architectures and training methods. When developing a machine translation system based on neural network structure, the major design question is how to set the meta-parameter values of the network structure and training configuration, so that the system performs well in terms of translation performance and computational cost. For network structure design, important meta-parameters include what kind of network should be applied, the number of layers, the number of units per layer, and unit type. With the increase of layers, the problem becomes more complex. For training configurations, important meta-parameters include learning algorithm, learning rate, dropout ratio and so on. All of these meta-parameters interact with each other and affect the performance of the whole system in a subtle way, thus they need to be tuned simultaneously.

Usually, these meta-parameters are tuned by human experts based on their experience. Such work requires much effort. In some ways such bottleneck may limit the wider adoption of NMT, or lead us into locally-optimal design decisions. Meanwhile, more powerful computing resource are available for academic and public use. Our motivation is to replace tedious manual tuning work with automatic computation conducted by computers.[1] As neural network training process is conducted off-line and a well-trained model can be used repeatedly, it is meaningful to allocate more computational resources to meta-parameter tuning as it can alleviate manual work.

Grid search is a simple method for meta-parameter optimization. However, as the number of meta-parameters increases, it becomes less tractable. This is because the number of lattice points increases in an exponential way with the increase of the number of meta-parameters. For example, if there are ten meta-parameters to be tuned and we only try five values for each parameter, it requires more than 750 billion (5^{10}) evaluations. In the case of NMT, training and evaluating one instance requires significant computational resource and time. Thus grid search is not a feasible method even using the fastest super computer. A black box meta-parameter optimization framework that can intelligently search a proper solution is needed.

Related work has proposed many meta-heuristic optimization methods such as genetic algorithms (GA) [1], evolutionary strategies (ES) [2], and Bayesian optimization (BO) [3]. They have all demonstrated success in many practical problems. In this study, we focus on an ES method called covariance matrix adaptation-evolution strategy (CMA-ES) [4] and its multi-objective extension [5, 6]. CMA-ES has been shown to be a practical and simple-to-implement algorithm that finds good solutions under few instance evaluations. To the best of our knowledge, this is the first work on applying CMA-ES to NMT.

Experiments are implemented with the Nematus machine translation toolkit [7]. Both single-objective optimization based on BLEU and multi-objective optimization based on BLEU and validation time are investigated, where validation time represent the time cost of generating translations on a validation data set. We show that CMA-ES can automatically find NMT models that improves upon the initial setting. Further, we analyze the factors that affect translation performance and computational time cost.

In the following, we introduce CMA-ES and its multi-

[1]We use the term "tuning" to refer to "hyperparameter search" in neural networks; note this differs from the fine-tuning in neural networks and the development set tuning in statistical MT system building.

objective extension in Section 2, then describe the machine translation system used in this work in Section 3. Experiment setup will be described in Section 5. Experiment results and analysis are in Section 6 and Section 7, followed by related work and conclusion.

2. CMA-ES META-PARAMETER OPTIMIZATION

2.1. CMA-ES

CMA-ES is a population-based meta-heuristics optimization method. Like GA, it encodes potential solutions as genes. The differences between GA and CMA-ES are that CMA-ES uses a fixed length vector $\boldsymbol{x}$ of real values as a gene, and uses a full covariance Gaussian distribution as gene distribution instead of directly representing them by a set of genes. In CMA-ES, it is assumed that the value of the objective function $f(\boldsymbol{x})$ can be evaluated, but the availability of an analytical form of the objective function $f(\boldsymbol{x})$ and its differentiability are not needed. Figure 1 shows the basic process of using CMA-ES.

In our experiment, the objective function $f(\boldsymbol{x})$ represents the performance of the machine translation system trained with a gene $\boldsymbol{x}$ encoding a set of meta-parameters. The meta-parameters include model structure and training configurations. Specifically, mean and covariance parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ of a Gaussian distribution for $\boldsymbol{x}$ is estimated by CMA-ES so that the distribution is concentrated in a region where $f(\boldsymbol{x})$ has a high value[2] by maximizing expectation $\mathbb{E}\left[f(\boldsymbol{x})|\boldsymbol{\theta}\right]$ as shown in Eq. (1).

$$\begin{aligned}\widehat{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{x}|\widehat{\boldsymbol{\theta}}) \ \text{ s.t. } \ \widehat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}\left[f(\boldsymbol{x})|\boldsymbol{\theta}\right] \\ &= \arg\max_{\boldsymbol{\theta}} \int f(\boldsymbol{x})\mathcal{N}(\boldsymbol{x}|\boldsymbol{\theta})d\boldsymbol{x}.\end{aligned} \tag{1}$$

In order to solve the problem efficiently, the natural gradient based gradient ascent is used. The expectation can be approximately computed by Monte Carlo sampling with the function evaluation $y_k = f(\boldsymbol{x}_k)$ as shown in Eq. (2).

$$\tilde{\nabla}_{\boldsymbol{\theta}}\mathbb{E}[f(\boldsymbol{x})|\boldsymbol{\theta}] \approx \frac{1}{K}\sum_{k=1}^{K} y_k \boldsymbol{F}_{\boldsymbol{\theta}}^{-1}\nabla_{\boldsymbol{\theta}} \log\mathcal{N}(\boldsymbol{x}_k|\boldsymbol{\theta}), \tag{2}$$

where $\boldsymbol{x}_k$ is a sample drawn from the previously estimated distribution $\mathcal{N}(\boldsymbol{x}|\hat{\boldsymbol{\theta}}_{n-1})$, and $\boldsymbol{F}$ is the Fisher information matrix.

Analytical forms of the updates of $\widehat{\boldsymbol{\mu}}_n$ and $\widehat{\boldsymbol{\Sigma}}_n$ are obtained by substituting the concrete Gaussian form into Eq.(2), leading to:

$$\begin{cases}\widehat{\boldsymbol{\mu}}_n = \widehat{\boldsymbol{\mu}}_{n-1} + \epsilon_{\boldsymbol{\mu}} \sum_{k=1}^{K} w(y_k)(\boldsymbol{x}_k - \widehat{\boldsymbol{\mu}}_{n-1}) \\ \widehat{\boldsymbol{\Sigma}}_n = \widehat{\boldsymbol{\Sigma}}_{n-1} + \epsilon_{\boldsymbol{\Sigma}} \sum_{k=1}^{K} w(y_k) \\ \qquad \cdot\big((\boldsymbol{x}_k - \widehat{\boldsymbol{\mu}}_{n-1})(\boldsymbol{x}_k - \widehat{\boldsymbol{\mu}}_{n-1})^{\intercal} - \widehat{\boldsymbol{\Sigma}}_{n-1}\big)\end{cases} \tag{3}$$

[2]Importantly, it is worth emphasizing that CMA-ES is a black-box method that makes no assumption on the relationship between gene value and system performance. The search distribution used to sample next generation genes is Gaussian, but $f(\boldsymbol{x})$ is not assumed to be Gaussian.

where $^{\intercal}$ is the matrix transpose. Note that as in [8], y_k in Eq.(2) is approximated in Eq.(3) as a weight function $w(y_k)$, which is defined as :

$$w(y_k) = \frac{\max\{0, \log(K/2+1) - \log(\mathrm{R}(y_k))\}}{\sum_{k'=1}^{K}\max\{0, \log(K/2+1) - \log(\mathrm{R}(y_{k'}))\}} - \frac{1}{K}, \tag{4}$$

and $\mathrm{R}(y_k)$ is a ranking function that returns the descending order of y_k among $y_{1:K}$. That is, $\mathrm{R}(y_k) = 1$ for the highest y_k, and $\mathrm{R}(y_k) = K$ for the smallest y_k. This equation only considers the order of y, which makes the updates less sensitive to the choice of evaluation measurements. As the correspondence to GA, the set of sampled genes $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_K\}$ represents a population of a generation, and an iteration of the gradient ascent corresponds to a generation.

2.2. Multi-objective CMA-ES using the Pareto frontier

In addition to the accuracy of translation, objectives such as time cost are also important in practice. Suppose we want to maximize J objectives $F(\boldsymbol{x}) \triangleq [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_J(\boldsymbol{x})]$ jointly with respect to $\boldsymbol{x}$. To handle the situation where the objectives may conflict with each other, we adopt Pareto optimality [9, 10]. We say $\boldsymbol{x}_k$ *dominates* $\boldsymbol{x}_{k'}$ if $f_j(\boldsymbol{x}_k) \geq f_j(\boldsymbol{x}_{k'})\ \forall\ j = 1, .., J$ and $f_j(\boldsymbol{x}_k) > f_j(\boldsymbol{x}_{k'})$ for at least one objective j, and write $F(\boldsymbol{x}_k) \triangleright F(\boldsymbol{x}_{k'})$. When given a set of candidate solutions, $\boldsymbol{x}_k$ is *Pareto-optimal* iff no other $\boldsymbol{x}_{k'}$ exists such that $F(\boldsymbol{x}_{k'}) \triangleright F(\boldsymbol{x}_k)$. There are several Pareto-optimal solutions given a set of candidates. The subset of all Pareto-optimal solutions is known as the Pareto frontier. Compared to combining multiple objectives into a single objective via an weighted linear combination, the Pareto definition has an advantage that weights need not be specified and it is more general.

CMA-ES can be extended to optimize multiple objectives by modifying the rank function $\mathrm{R}(y_k)$ used in Eq.(4). Given a set of solutions $\{\boldsymbol{x}_k\}$, we first assign rank = 1 to those on the Pareto frontier. Then, we exclude these rank 1 solutions and compute the Pareto frontier again for the remaining solutions, assigning them rank 2. This process is iterated until no $\{\boldsymbol{x}_k\}$ remain, and we obtain a ranking of all solutions according to multiple objectives in the end. Figure 2 shows the intuition behind multi-objective optimization in our work, where BLEU score and negative validation time are used as the objectives. We expect superior individuals with higher BLEU score and lower translation time than the initial one are obtained by the automatic optimization by CMA-ES.

3. Neural Machine translation

3.1. Encode-Decoder Model

The neural machine translation (NMT) system we used in this experiment is based on an attentional encoder-decoder architecture as implemented in Nematus [7]. This is very similar to the structure proposed by [11].

The NMT model is part of the family of models using

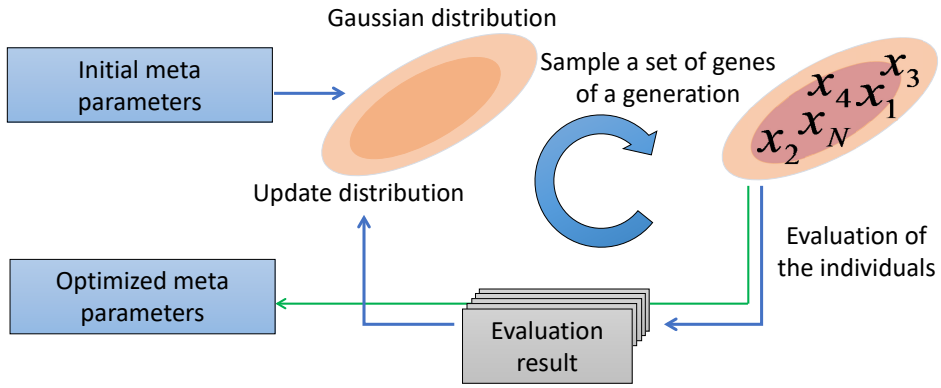


Figure 1: Automatic system tuning process using CMA-ES.

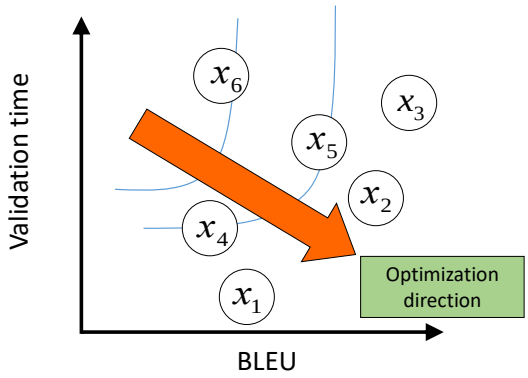


Figure 2: Pareto based optimization for two objectives.

encoder-decoder with recurrent neural networks. The encoder is implemented as a bidirectional neural network with gated recurrent unit [12]. First, it reads the input sentence, which is a sequence of m words $x = x_1, ..., x_m$. The forward RNN reads the input sequence from x_1 to x_m and the reverse RNN reads the sequence from x_m to x_1. The hidden states of the two RNN at each time-step are concatenated to form the encoding, or annotation vectors $h_1, ..., h_m$.

The decoder is trained to predict the target word sequence $y = (y_1, ..., y_n)$, and is also implemented as a recurrent neural network. The decoder predicts each word y_i based on a recurrent state s_i, previous word y_{i-1} and a context vector c_i. The context vector c_i is computed as a weighted sum of annotations $c_i = \sum_{j=1,...,m} \alpha_{ij} h_j$, where the weight α_{ij} is based on a single-layer feedforward neural network. The weights can be viewed as "attention" on the input. During training, we use the previous word y_{i-1} according to the target reference; during evaluation or test, we use the word previously predicted by the RNN decoder as y_{i-1} and run a beam search to generate the translation (beam is 5 in our case).

3.2. Subword Preprocessing

We follow the work of [13] in subword preprocessing, which uses byte pair encoding (BPE) to split words into subword units. The motivation is to reduce the number of distinct vocabulary items in the Encoder-Decoder model. Large vocabulary may lead to slower models and sparser statistics.

We briefly describe the BPE preprocessing procedure here: First the symbol vocabulary is initialized with all characters in the training set. The frequency of each symbol pair is calculated, and we iteratively merge the most frequent pair to create a new symbol. In other words, each merge operation produces a new vocabulary item that represents a character n-gram. Very frequent character n-grams, such as frequent words, eventually become a single symbol. The final vocabulary size of BPE equals to the size of the initial character set, plus the number of BPE merge operations.

While BPE is a simple preprocessing method to handle the large vocabulary problem in NMT, the optimal number of BPE merge operations in unclear. Intuitively, a larger vocabulary size should lead to better translation accuracy, assuming sufficient data to estimate the model parameters. The effect on translation time is uncertain: on one hand, smaller vocabulary implies a faster softmax operation in the RNN decoder, but also a longer sequence to process. Finally, the impacts of BPE vocabulary size may be different for source and target.

4. EXPERIMENTAL SETUP

4.1. Data

We performed two sets of experiments: single-objective experiment and multi-objective experiment. In the single-objective experiment, we optimize translation accuracy, which is measured by BLEU on the validation (development) set. In the multi-objective one, we optimize translation accuracy and computational cost jointly. The computational cost is measured by the translation time (seconds) on validation set. We use the data from Kyoto Free Translation Task version 1.4 (KFTT)[3] for both experiments. KFTT contains Wikipedia articles about Kyoto tourism and traditional Japanese culture, religion, and history. These articles are originally in Japanese and are manually translated

[3] http://www.phontron.com/kftt/

Table 1: Data statistics

	Articles	Sentence pairs	Japanese words	English words
Train	14126	330k	6.2M	5.9M
Dev	15	1166	27.8k	24.3k
Test	15	1160	29.6k	26.7k

into English by NICT.[4] The English side is preprocessed (i.e. tokenized, lowercased, filtered to exclude sentences more than 40 words) using standard machine translation tools from Moses, and the Japanese side is word-segmented using Kytea[5]. Both sides are then broken in subword units independently using BPE, where the exact BPE meta-parameter (number of merge operations) is automatically tuned via CMA-ES.

The bitext is separated into training, validation (dev), and test sets. The training set is used for training the NMT models, development set used for measuring BLEU and computation time, the objectives to be optimized. The test set is only used for reporting final results. We focus on the Japanese-English direction, and the baseline results using Giza++/Moses PBMT on the KFTT leaderboard is 15.41 BLEU for dev and 17.68 BLEU for test. There is a stronger result of 16.93 BLEU for dev and 19.35 BLEU for test on the leaderboard. It is a Moses PBMT system that utilizes pre-ordering (permuting Japanese words into English SVO order prior to training and translation), which have demonstrated substantial gains in Japanese-English tasks [14, 15]. We compare with the standard 15.41 BLEU baseline using bitext in their original word order, and leave pre-ordering's effect on NMT to future work. Table 1 summarizes the data used for experiments.

4.2. Meta-parameters

Table 2 shows meta-parameters that are subject to tuning by CMA-ES. All Nematus meta-parameters that are not shown in the table are set to their defaults. The meta-parameters we tune for can be divided into model architecture (e.g. size of embedding, LSTM unit) and training configuration (learning rate, drop out). Their initial values were manually tuned slightly to achieve a reasonable starting BLEU of 16.48 on the dev set and 15.13 on the test set. The corresponding computation times for decoding the dev and test sets are 248 and 230 seconds, respectively.

Our goal in the experiment is to run evolution and observe whether these initial values and corresponding BLEU/time can be automatically improved without manual effort. If evolution can search through a large range for meta-parameters, we can expect a highly optimized system. That is the generalization of this black-box approach to automatic optimization. The generality can also help us investigate the association of some meta-parameters with the machine translation system's performance. The same initial values are used for both single-objective and multiple-objective experiments.

In order to apply CMA-ES, we first need to encode the meta-parameters into a fixed-dimensional gene vector. Depending on the domain and possible values of each meta-parameter, a mapping from a real number to a desired domain is needed to translate the gene value to the actual configuration. For the meta-parameters BPE merge operations (bpe_op_src, bpe_op_trg), word embedding dimension (dim_word) and LSTM dimensions (dim_lstm), we used $int(exp(x))$ since they may be large positive integers and $exp(x)$ can represent a large number with a small exponent. For the other meta-parameters such as dropout, alignment regularization, and learning rate, which are positive but small, we used $abs()$ to ensure they are positive as the genes sampled from Gaussian distribution might be negative. There were 10 meta-parameters to tune so the dimension of gene vector was 10, for both single and multi-objective experiment.

4.3. Details of the CMA-ES Setup

Experiments were performed using the TSUBAME 2.5 supercomputer that equips with NVIDIA K20X GPGPU's[6]. We have conducted 10 CMA-ES generations for single-objective experiment and 5 generations for multi-objective experiment. Each generation consisted of 30 individuals for both single and multi-objective optimization. In the single-objective experiment, the training time was limited to a maximum of 48 hours for each generation; in multi-objective experiment, the training time was a maximum of 36 hours. We limited the maximum training time for computational reasons: we found that sometimes the training process of an model may take a week until convergence, but in practice the BLEU scores are not very different from the model at 36+ hours. (See Figure 3 for an example). We think that in CMA-ES, high precision estimates of the final BLEU or time values at convergence are not necessary. It is more efficient to run more generations of CMA-ES, as opposed to spending a long time to obtain the most precise estimate of a gene's BLEU/time rank. The 36 or 48 hours limit on training time is a practical tradeoff.

The experimental process is shown in Figure 4. After sampling genes from the Gaussian search distribution, genes will be converted into meta-parameter configurations. Then the model will be trained using the training set for up to 36 or 48 hours. We call each set of configurations an individual or gene, interchangeably. All individuals of one generation are executed in parallel. After training, the models are used to translate the dev set, and BLEU scores and computation time scores are collected. We rank all individuals based on

[4] http://alaginrc.nict.go.jp/WikiCorpus/index_E.html

[5] http://www.phontron.com/kytea/

[6] http://www.gsic.titech.ac.jp/en

Table 2: Meta-parameters tuned in this study. The initial values are the baseline settings obtained by manual tuning, and is the first individual seeded in CMA-ES. Example results of single and multiple objective evolution are shown: (a) is the individual with maximum dev BLEU of single-objective evolution, achieved at generation 8; see Figure 5. (b) is the individual with minimum computation time in multi-objective evolution's final generation, (c) is the individual with the maximum dev BLEU in multi-objective evolution, achieved at generation 3, and (d) is another individual on the Pareto frontier, achieved at generation 2. Note that (b), (c), and (d) are three of the five points on the Pareto frontier in Figure 6. All of them are considered "optimal" in the multi-objective sense and the single model to deploy in practice should be the human designer's decision.

Meta-parameter	Initial value	(a) Single objective	(b) Multiple objective	(c) Multiple objective	(d) Multiple objective
# BPE merge operations on Source (bpe_op_src)	5000	5250	5345	5011	5102
# BPE merge operations on Target (bpe_op_trg)	5000	6617	4622	5706	5877
# dimension of word embedding (dim_word)	100	121	333	99	104
# of LSTM units (dim_lstm)	400	496	123	459	430
alignment regularization (alpha_c)	0	0.188	0.158	0.249	0.043
learning rate	0.0001	0.100	0.213	0.295	0.083
dropout prob. of embedding (dropout_embedding)	0.2	0.148	0.017	0.147	0.070
dropout prob. of LSTM hidden unit (dropout_hidden)	0.2	0.152	0.036	0.099	0.103
dropout prob. of source words (dropout_source)	0.1	0.026	0.044	0.117	0.013
dropout prob. of target words (dropout_target)	0.1	0.204	0.094	0.019	0.102
dev BLEU	16.48	18.83	17.42	18.04	18.02
dev computation time	248	264	222	269	241

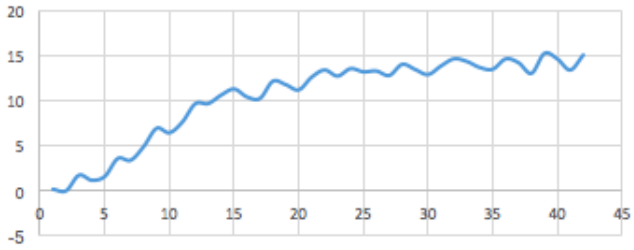


Figure 3: BLEU (y-axis) by number of epoch (x-axis) for an example model/gene.

their scores and update the distribution, via CMA-ES update equations. We then sample new genes and the whole process is repeated for a number of generations until our budget constraint (e.g. 10 generations for single-objective experiment).

5. RESULTS

5.1. Single-objective evolution

For the single-objective evolution experiment, we evaluated a total of $10 \times 30 = 300$ models. For visualization purposes, we choose those individuals with the highest dev BLEU in each generation and plot them on Figure 5. The figure shows how development set BLEU and validation time varies with the number of generation in single-objective evolution that optimizes for development set BLEU.

We observe a general trend of increasing BLEU as evolution progresses. For example, the 8-th generation achieves 18.83 BLEU, the highest among all results, and significantly improves from the baseline of 16.48. There is no guarantee that the improvements are monotonic, however; for example, note that an individual in generation 7 achieves lower BLEU compared to that of generation 6. There is also a slight increase in computation time during the evolution process, which is expected since our single-objective CMA-ES does not account for that objective.

To summarize, the best individual of CMA-ES, achieved at generation 8, has a dev BLEU of 18.83. This outperforms the dev BLEU of our NMT baseline initial setting (16.48) and the KFTT Moses baseline (15.41). In terms of BLEU on the *test set*, this model achieves 16.45, which is an improvement over the NMT baseline initial setting (15.13). So we conclude that CMA-ES has demonstrated its ability to improve upon manually-tuned results. This is done at the expense of considerable computational resources, but the process is entirely automatic and required no human intervention.[7] Meta-parameters of this model is shown in Table 2, column (a).

5.2. Multi-objective evolution

Figure 6 shows a visualization of our multi-objective evolution results, where we evaluated a total of $5 \times 30 = 150$ models. The Pareto optimal models of each generation are plotted. Note that there is a general trend toward individuals moving towards the lower-right hand side of the plot. If we compute the Pareto frontier on all points aggregated in Figure 6, we will obtain 5 points: (18.04 BLEU, 269 seconds), (18.02, 241), (17.42, 222), (16.82, 209), (16.66, 206). The first three of these are shown as examples (b), (c), (d) in Table

[7]However, note that our best NMT test BLEU is still lower than the KFTT Moses baseline test BLEU (17.68). Further work is needed to examine the differences between NMT and PBMT on this dataset.

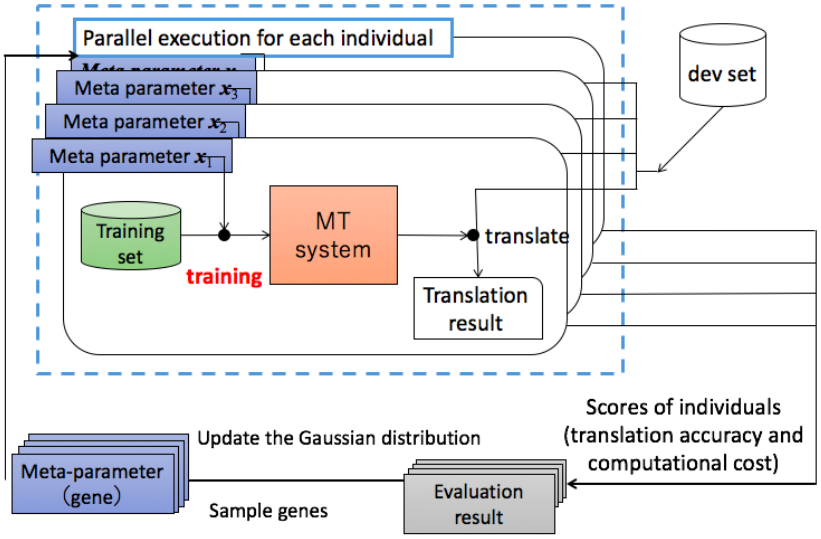


Figure 4: Experimental process of applying CMA-ES to automatically tune NMT models.

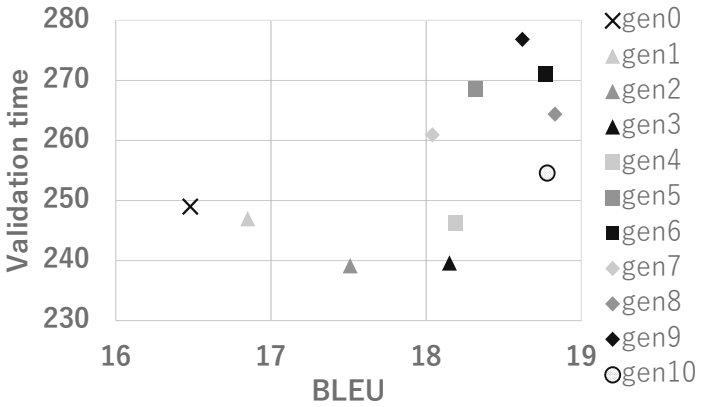


Figure 5: Single-objective evolution results, from generation (gen) 1 to 10. The baseline model with initial value settings is labeled as gen0 and indicated by a cross (x). Note the general improvement of BLEU from the early generations (gen1-3, labeled as triangles) to the later ones (rhombus and circle).

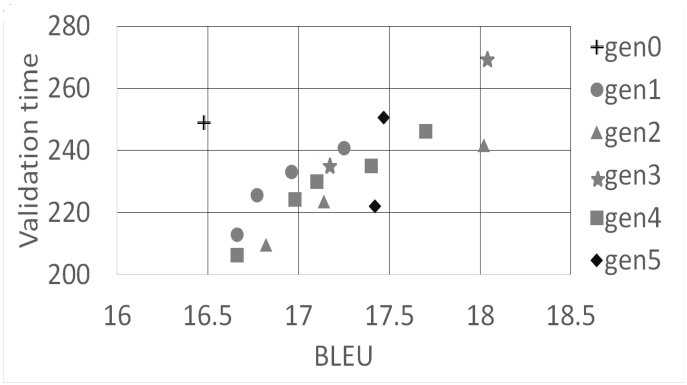


Figure 6: Multi-objective evolution results. The initial model (gen0) is labeled (+), followed by generation 1 models (circles), generation 2 models (triangles), etc.

2. The meta-parameter settings of these Pareto-optimal models are quite distinct. For instance, example (b) has small LSTM units while examples (c) and (d) have larger LSTM units but smaller word embedding dimensions. The target vocabulary (bpe_op_trg) of example (b) is smaller than the initial setting, while those of (c) and (d) are larger; all have larger source target vocabulary.

All the Pareto points in the multi-objective evolution results outperform the baseline initial setting in terms of dev BLEU; some of them outperform the baseline in both BLEU and computation time. Therefore we conclude that the Pareto extension to CMA-ES is achieving its expected effect. There are no improvements in terms of BLEU over the single-objective CMA-ES setting, however. One reason might be that the computation resources used for the multi-objective experiment is less than that of the single-objective experiment. In any case, ideally multi-objective optimization will subsume the single-objective case, and we plan to investigate this further in future work.

6. Analysis

While the results are promising, we want to analyze the statistics of our experiments in order to improve the efficiency of CMA-ES for future work. Figure 7 plots the distribution of various meta-parameters computed across the 300 and 150 models in single- and multi-objective experiments. We note the distribution of word and LSTM dimensions has much wider variance in the multi-objective case compared to the single-objective case (Figure 7 (d) vs (c)), which is expected. Interesting, the range of BPE merge operations (and

thus, the final vocabulary size) is relatively small for both cases (Figure 7 (b) vs (a)). We hypothesize there needs to be some more aggressive (or diverse) sampling in order to fully explore the meta-parameter space. We also think our mapping function that converts real numbers from the CMA-ES Gaussian sample to training configurations may require some re-design: for example, the range of $int(exp())$ may be too narrow, and the use of $abs()$ may induce symmetric properties and confound positive and negative values.

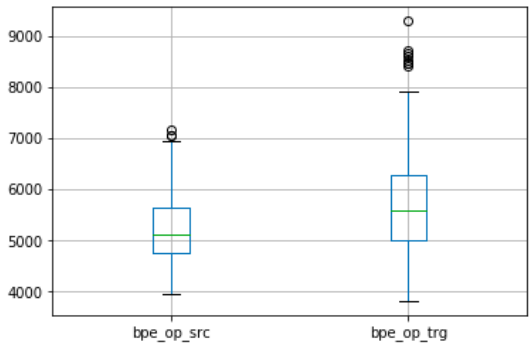
7. Conclusion & Related Work

We demonstrate that an evolution strategy like CMA-ES can be use to automate the tuning of neural network based machine translation system. We start with an initial manually-tuned NMT baseline on KFTT, and show that our our single-objective and multi-objective CMA-ES method can create models that perform better in BLEU and/or computation time.

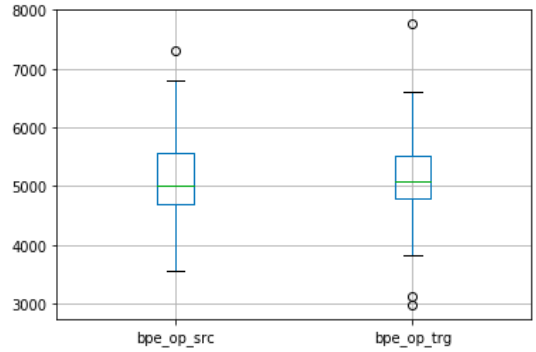
There is a large literature on blackbox optimization, with many successes in practical problems that are difficult to characterize. The main approaches include evolutionary methods (GA or ES) [1, 2] and Bayesian optimization [3, 16]. Recently, in the context of automatic tuning of neural network systems, reinforcement learning [17] and a bandit learning [18] approaches have been proposed. Each approach has its strengths: Evolutionary strategies are efficiently parallelizable. Bayesian optimization models uncertainty in a principled fashion. Reinforcement learning captures sequential dependencies among hyperparameters. Bandit learning provides a framework for trading-off computational resources. In future work, it will be interesting to compare these different approaches on a wider array of datasets.

Pareto optimality has been applied to statistical MT in the context of optimizing multiple evaluation metrics such as BLEU and TER [19, 20]. We are not aware of previous work that performs multi-objective optimization on BLEU and computation time.

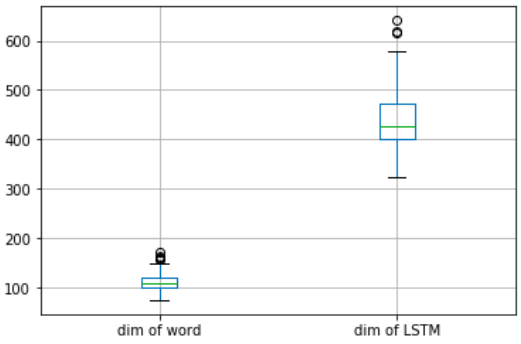
For automatic tuning of neural networks, evolutionary strategies have demonstrated strong results in image classification [21], acoustic modeling [6], and language modeling [22], among others. In NMT, a grid search of meta-parameters is performed in [23]. They used a total of more than 250,000 GPU hours to explore common variations in NMT architectures. Their conclusions include: (a) deep encoders are more difficult to optimize than decoders, (b) dense residual connections are good, (c) LSTMs outperform GRUs. Our work investigates different meta-parameters; it will be interesting to validate their findings with CMA-ES.



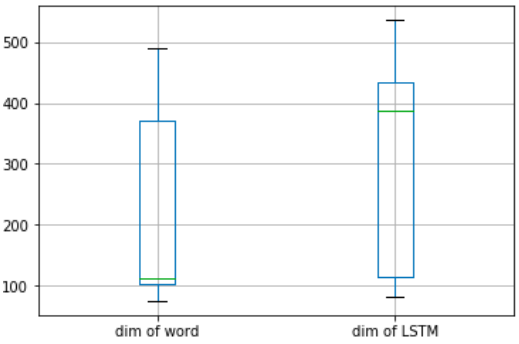
(a) Single objective, BPE



(b) Multi objective, BPE



(c) Single objective, dimensions



(d) Multi objective, dimensions

Figure 7: Boxplot showing the distributions of meta-parameters searched by single-objective and multi-objective CMA-ES.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Numbers JP26280055 and JP17K20001.

8. References

[1] L. Davis, Ed., *Handbook of genetic algorithms*. Van Nostrand Reinhold New York, 1991, vol. 115.

[2] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, "Natural evolution strategies," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 949–980, 2014.

[3] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithm s," in *Advances in Neural Information Processing Systems 25*, 2012.

[4] N. Hansen, "The CMA evolution strategy: a comparing review," in *Towards a new evolutionary computation*. Springer, 2006, pp. 75–102.

[5] S. Rostami and A. Shenfield, "CMA-PAES: Pareto archived evolution strategy using covariance matrix adaptation for multi-objective optimisation," in *2012 12th UK Workshop on Computational Intelligence (UKCI)*, Sept 2012, pp. 1–8.

[6] T. Moriya, T. Tanaka, T. Shinozaki, S. Watanabe, and K. Duh, "Automation of system building for state-of-the-art large vocabulary speech recognition using evolution strategy," in *Proceedings of the IEEE 2015 Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.

[7] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, A. V. Miceli Barone, J. Mokry, and M. Nadejde, "Nematus: a toolkit for neural machine translation," in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain: Association for Computational Linguistics, April 2017, pp. 65–68. [Online]. Available: http://aclweb.org/anthology/E17-3017

[8] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[9] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer, 1998.

[10] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, 2004.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[12] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translations," in *Conference on Empirical Methods in Natural Language Processing (EMNLP) 2014*, ser. cs.CL, no. 1406.1078, 2014. [Online]. Available: http://arxiv.org/abs/1406.1078

[13] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 1715–1725. [Online]. Available: http://www.aclweb.org/anthology/P16-1162.pdf

[14] H. Isozaki, K. Sudoh, H. Tsukada, and K. Duh, "Head finalization: A simple reordering rule for sov languages," in *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 244–251. [Online]. Available: http://www.aclweb.org/anthology/W10-1736

[15] G. Neubig, T. Watanabe, and S. Mori, "Inducing a discriminative parser to optimize machine translation reordering," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 843–853. [Online]. Available: http://www.aclweb.org/anthology/D12-1077

[16] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, p. 28, 12/2015 2016.

[17] B. Zoph and Q. Le, "Neural architecture search with reinforcement learning," in *Proceedings of the International Conference on Representation Learning (ICLR)*, 2017.

[18] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[19] K. Duh, K. Sudoh, X. Wu, H. Tsukada, and M. Nagata, "Learning to translate with multiple objectives," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2012.

[20] B. Sankaran, A. Sarkar, and K. Duh, "Multi-metric optimization using ensemble tuning," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 947–957. [Online]. Available: http://www.aclweb.org/anthology/N13-1115

[21] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. [Online]. Available: http://arxiv.org/abs/1703.01041

[22] T. Tanaka, T. Moriya, T. Shinozaki, S. Watanabe, T. Hori, and K. Duh, "Automated structure discovery and parameter tuning of neural network language model based on evolution strategy," in *Proceedings of the 2016 IEEE Workshop on Spoken Language Technology*, 2016.

[23] D. Britz, A. Goldie, M.-T. Luong, and Q. V. Le, "Massive exploration of neural machine translation architectures," *CoRR*, vol. abs/1703.03906, 2017.

Continuous Space Reordering Models for Phrase-based MT

Nadir Durrani *Fahim Dalvi*

Qatar Computing Research Institute – HBKU

{ndurrani,faimaduddin}@qf.org.qa

Abstract

Bilingual sequence models improve phrase-based translation and reordering by overcoming phrasal independence assumption and handling long range reordering. However, due to data sparsity, these models often fall back to very small context sizes. This problem has been previously addressed by learning sequences over generalized representations such as POS tags or word clusters. In this paper, we explore an alternative based on neural network models. More concretely we train neuralized versions of lexicalized reordering [1] and the operation sequence models [2] using feed-forward neural network. Our results show improvements of up to 0.6 and 0.5 BLEU points on top of the baseline German→English and English→German systems. We also observed improvements compared to the systems that used POS tags and word clusters to train these models. Because we modify the bilingual corpus to integrate reordering operations, this allows us to also train a *sequence-to-sequence* neural MT model having explicit reordering triggers. Our motivation was to directly enable reordering information in the encoder-decoder framework, which otherwise relies solely on the `attention model` to handle long range reordering. We tried both coarser and fine-grained reordering operations. However, these experiments did not yield any improvements over the baseline Neural MT systems.

1. Introduction

Source-target bilingual sequence models have been used successfully as feature in phrase-based SMT [3, 2]. They are based on minimal translation units, and overcome independence assumption by handling non-local dependencies across phrasal boundaries, thus providing better translation and reordering mechanism. Such models however suffer from data sparsity and fall back to very small context sizes during test time. This shortcoming is addressed by learning factored models [4, 5, 6], learned over POS and morphological tags or using word classes [7, 8, 9].[1]

An alternative way to address data sparsity and learn better generalizations is to use continuous representations. Neural networks (NN) have shown success in Statistical Machine translation with n-best re-ranking [12, 13] or directly as a feature [14, 15] used during decoding. More recently, attention-based encoder-decoder Recurrant Neural Network (RNN) model [16], which trains a single large neural network, has emerged as the new state-of-the-art in MT.

In this work, we neuralize two commonly used reordering models namely lexicalized reordering [1] and the operation sequence model (OSM) [2] and integrate them as feature in phrase-based MT. We convert word-aligned bi-text into a sequence of operations through a deterministic algorithm (See Algorithm 1 in [17]), the resulting vocabulary and number of model parameters can become very large. A model trained on such representation may suffer from data sparsity. To overcome this, we separate the streams of source and target sequences and concatenate them to simulate the jointness. A feed-forward neural network is then trained on such concatenated n-gram sequences.

The OSM model exhibit very rich reordering operations varying from `Insert GAP` to `JUMP Forward` and `JUMP Backward` to multiple open gaps which may be hierarchically created. In an alternative method, we replace complicated reordering operations with `Monotonic`, `Swap and Discontinuous` operations, and train a neural model with coarser tags. This model is similar to the lexicalized reordering model, however much richer as it is conditioned on longer source-target contextual history and also previous reordering decisions.

We experimented with German-to-English and English-to-German language pairs. German is syntactically divergent from English and also exhibit very rich morphology, thus prone to data sparsity. These are the two problems we are addressing in this work. Our results show improvements of up to +0.6 and +0.5 BLEU points in German-to-English and English-to-German baselines respectively. We also demonstrated that neuralized OSM model performed better than the ones trained on POS tags and word-clusters. The neuralized OSM model outperformed the simpler lexicalized variant, although only slightly.

While training the Neural OSM (and Neural lexicalized reordering model) we embed reordering information in form of operations in the training corpus. This also allows us to train `sequence-to-sequence` neural MT system, where the target side is conditioned on both lexical and reordering states. [18] recently showed that integrating structural bias such as `Position bias`, i.e. relative positions of a source and corresponding target word, improves the attention mechanism. We tried to replicate this effect by i)

[1] as obtained during GIZA training [10] or using brown clusters [11]

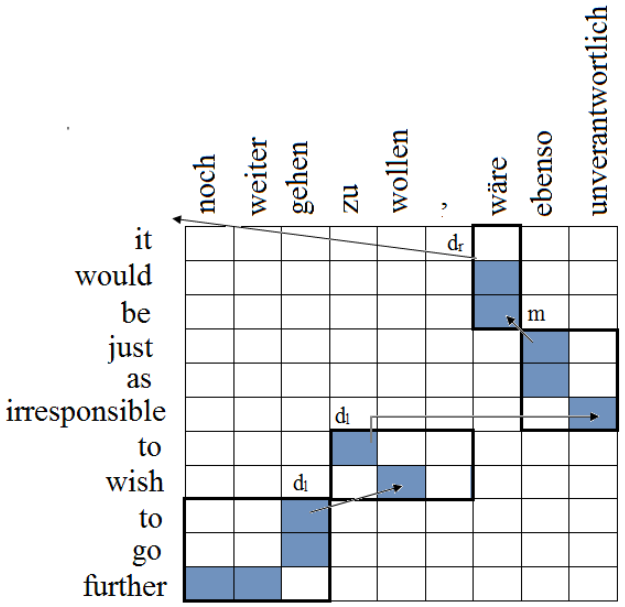


Figure 1: Lexical Reordering Models [20]: m = monotonic, s = swap, d = discontinuity (l: left, r: right)

linearizing the source to be in the same order as the target using word-alignments and ii) incorporating reordering states. Our motivation was that such reordering triggers will aid the attention model to better handle reordering. However, our results did not yield any improvements.

The remainder of this paper has been organized as follows: Section 2 describes the operation sequence and the lexicalized reordering model. We then present the neuralized versions of these models. Sections 3 and 4 describe our experimental setup and discusses the results. Section 5 gives account on related work and Section 6 concludes the paper.

2. Reordering Models

In this section we briefly revisit the two commonly used reordering models in the phrase-based Moses [19] namely the lexicalized reordering model and the operation sequence model. We then describe our neural versions of these models.

2.1. Lexicalized Reordering Model

The lexicalized reordering model originally proposed by [1] is the defacto reordering model used in phrase-based SMT (PBSMT). The idea is to learn orientation of a phrase w.r.t to previous phrase (or the last word of the previous phrase). An orientation could be one of the three reordering operations namely **M**`onotonic`, **S**`wap`, **D**`iscontinuous`. If the source phrases $F_{a(j-1)}$ and $F_{a(j)}$[2] are adjacent and in the same order as the target phrases E_{j-1} and E_j, the orientation is `Monotonic`. If they are in the opposite order of E_{j-1} and E_j, then the orientation is `Swap`, otherwise it is `Discontinuous`. See Figure 1 for illustration. For each phrase, we compute its probability of being reordered with the orientations $o = M, S, D$ as below:

$$p_r(o|F_{a(j)}, E_j) = \frac{count(o, F_{a(j)}, E_j)}{count(F_{a(j)}, E_j)})$$

Improved versions [20, 21] have been subsequently integrated into Moses toolkit. The former computes orientation only based on the last word of the previous phrase, rather than the entire phrase and the latter, hierarchically combines all previous phrases to compute the probability. In our work, we will compute orientation based on previous source word, but condition on n previous source-target units. This is because our model is based on minimal translation units [3] and does not contain phrasal boundaries.

2.2. Operation Sequence Model

The operation sequence model (OSM) converts aligned bilingual corpus into a sequence of operations using a deterministic algorithm. An operation is either joint source-target lexical generation, or a reordering operation such as `Insert Gap` or `Jump Forward` or `Backward` to a specific open gap. A Markov model is estimated from the resulting operation corpus. More formally a bilingual sentence pair (T, S) and its word-alignment A is transformed deterministically to a heterogeneous sequence of translation and reordering operations $(o_1, o_2, \ldots, o_J)$. A 5-gram model is then learned over these sequences:

$$P_{osm}(T, S) \approx \prod_{j=1}^{J} P(o_j|o_{j-n+1}...o_{j-1})$$

The operation sequence for the example shown in Figure 1 according to the algorithm described in the original paper is given below:

Generate Target Only (it) – Insert Gap – Generate (wäre, would be) – Generate (ebenso, just as) – Generate (unverantwortlich, irresponsible) – Jump Back (1) – Insert Gap – Generate (zu, to) – Generate (wollen, wish) – Generate Source Only (,) – Jump Back (1) – Insert Gap – Generate (gehen, to go) – Jump Back (1) – Generate (noch weiter, further)

The OSM is trained on minimal translation units (MTUs) and does not adhere to phrasal boundaries. Access to joint source target information enables it to better handle long distance dependencies. The jumps and gap operations allow OSM to learn more complex reordering patterns. However, due to data sparsity it is impossible to observe all possible reordering patterns during the training. The model therefore falls back to very small context sizes. Earlier work has addressed this problem by estimating estimating the bilingual sequence models on POS tags or word clusters [22, 5, 6].

[2]The mapping function $a(j)$ aligns the target phrase E_j to the source phrase F_i, where $F_i = F_{a(j)}$.

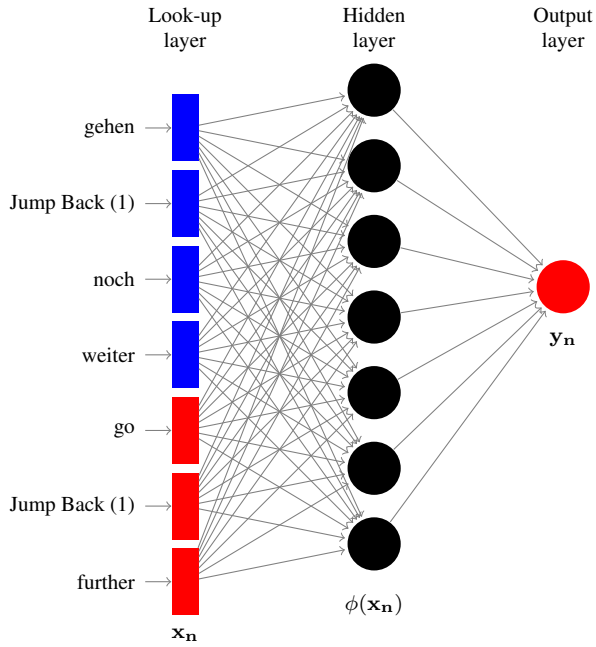


Figure 2: Neural OSM model where we use 3-gram target words and a source context window of size 4. For illustration, the output y_n is shown as a single categorical variable (scalar) as opposed to the traditional one-hot vector representation.

2.3. Neural Reordering Models

In this paper we take a different approach to address the problem of data sparsity by training the model using a feed forward neural network. Below we present the proposed neural versions of the OSM and lexicalized reordering models.

2.3.1. Neural Operation Sequence Model

A straight forward way is to build a neural language model using the generated sequences of operations. However, because of the joint nature of the model, the vocab size becomes quadratic ($M\times$N) causing severe data sparsity. A way to alleviate this problem is to separate out source and target streams and concatenate them to form history. See Table 1 for mapping operations into separate streams of source and target operations. Here are the considerations that we made: i) When a source or target word is unaligned (`Generate Source Only (Y)` or `Generate Target Only (X)` operations), we don't append anything on the other side, ii) Whenever there is a reordering operation (`Insert Gap/Jump Forward/Jump Back (N)`) we append it on both sides, iii) We replace source words on both sides for the `Generate Self` operation, iv) Multi-word source and target cepts are collapsed together even if they appear in a different order in the original sequence, v) Note that source-side is now reordered to be order of target just as in the original model. We generate separate streams of source and target operation and then concatenate them to train the neural model. Let $s_o = s_{o_1}, s_{o_2}...s_{o_n}$ and $t_o = t_{o_1}, t_{o_2}...t_{o_m}$ be streams of source and target operations, the model is defined as:

$$P(T,S) \approx \prod_{j=1}^{J} P(t_{o_j}|t_{o_{j-n+1}}...t_{o_{j-1}}, s_{o_j}...s_{o_{j-m+1}})$$

where m and n are the source and target word histories which we concatenate to form input to the neural network. As exemplified in Figure 2, this is essentially an $(m+n)$-gram neural network LM (NNLM) originally proposed by [23]. Each input word i.e. source or target vocabulary word or a reordering operation in the context is represented by a D dimensional vector in the shared look-up layer $L \in \mathbb{R}^{|V_i|\times D}$ where V_i is the input vocabulary.[3] The look-up layer then creates a context vector $\mathbf{x_n}$ representing the context words of the $(m+n)$-gram sequence by concatenating their respective vectors in L. The concatenated vector is then passed through non-linear hidden layers to learn a high-level representation, which is in turn fed to the output layer. The output layer has a `softmax` activation over the output vocabulary V_o of target words. Formally, the probability of getting k-th word in the output given the context $\mathbf{x_n}$ can be written as:

$$P(y_n = k|\mathbf{x_n}, \theta) = \frac{exp\ (\mathbf{w}_k^T \phi(\mathbf{x_n}))}{\sum_{m=1}^{|V_o|} exp\ (\mathbf{w}_m^T \phi(\mathbf{x_n}))} \quad (1)$$

where $\phi(\mathbf{x_n})$ defines the transformations of $\mathbf{x_n}$ through the hidden layers, and $\mathbf{w}_k$ are the weights from the last hidden layer to the output layer. For notational simplicity, henceforth we will use $(\mathbf{x_n}, y_n)$ to represent a training sequence. By setting m and n to be sufficiently large, neural OSM can capture long-range cross-lingual dependencies between words, while still overcoming the data sparseness issue by virtue of its distributed representations (i.e., word vectors).

2.3.2. Neural Lexicalized Reordering Model

We train the neural lexicalized reordering model in the same manner as that of the Neural OSM model. Traditional lexicalized reordering models use `Monotonic, Swap and Discontinuous`. We retained the `Swap (SW)` operation and divided the `Discontinuous (D)` category into `Forward Discontinuity (FD)` and `Backward Discontinuity (BD)` following [24]. We also removed the `Monotonic` orientation from the generation as it is obvious that words flow monotonically when there is no reordering. This is also done similarly in the OSM generation. Again like the Neural OSM generation, the reordering tags are split across both the streams. See Table 1 for the sample generation (last 2 columns).

Note that this model is not exactly the neural version of the lexicalized reordering in which the task is just to predict orientation/reordering decision (`Monotonic, Swap, Discontinuous`) based on previous source-target word (or phrase). Here we are trying to score the entire sequence

[3] Note that L is a model parameter to be learned.

Operations	Source Stream	Target Stream	Source Stream	Target Stream
Generate Target Only (it)	it		it	
Insert Gap	Insert Gap	Insert Gap	Jump Fwd	FD
Generate (wäre, would be)	wäre	would be	wäre	would be
Generate (ebenso, just as)	ebenso	just as	ebenso	just as
Generate (unverantwortlich, irresponsible)	unverantwortlich	irresponsible	unverantwortlich	irresponsible
Jump Back (1)	Jump Back (1)	Jump Back (1)	BD	BD
Insert Gap	Insert Gap	Insert Gap		
Generate (zu, to)	zu	to	zu	to
Generate (wollen, wish)	wollen	wish	wollen	wish
Generate Source Only (,)	,		,	
Jump Back (1)	Jump Back (1)	Jump Back (1)	BD	BD
Insert Gap	Insert Gap	Insert Gap		
Generate (gehen, to go)	gehen	to go	gehen	to go
Jump Back (1)	Jump Back (1)	Jump Back (1)	BD	BD
Generate (noch weiter, further)	noch weiter	further	noch weiter	further

Table 1: Operation Sequences and corresponding streams for Neural OSM and Lexicalized RM training

which contains both lexical (word generation) and reordering choices. The task is to find most probable sequence of lexical and reordering decisions. The difference compared to the OSM is the granularity of the reordering tags. In this model, we just have one reordering decision per lexical generation. In the OSM model, the model can have very complex sequence of reordering operations in between adjacent lexical generations. A more accurate version of the neural lexicalized reordering is described in [25]. They cast it as a classification problem, and use a continuous space representation treating a phrase as a dense real-valued vector. But unlike traditional model, they condition reordering probabilities also on the words of previous phrase to capture longer dependencies. This is similar to our work, except that our context information can go even beyond previous phrases and previous reordering decisions are also part of the context.

2.3.3. Neural Lexical Sequence Model

In this variation, we simply remove the reordering operations from the sequences and train the neural model only on the lexical sequences. This allows us to study how much of the improvement is obtained due reordering triggers integrated within these lexical sequences versus addressing sparsity by learning generalized representations. However note that such a lexical sequence model can still be considered a reordering model because the source was pre-ordered (or linearized) based on target (See Table 1) and generated in the target order. This model is based on the tuple sequence model [26] and several neural variants of it are presented in [13]. Another variation is presented in [15], but rather than pre-ordering the source, they select m neighboring word on the left and right sides of the source word s_i that is aligned to the target word t_i being modeled.

2.3.4. Decoding

We integrate these models as a feature in phrase-based decoding. Word alignments for the current phrase along with the history of previously generated operations are used to generate a new sequence of (lexical and reordering) operations. This sequence is then scored to give probability of the hypothesized phrase.

3. Experiments

3.1. Training Data

We experimented with German↔English language pairs using the data made available for the International Workshop on Spoken Language Translation (IWSLT'14). The data contains roughly 5M bilingual sentence pairs. We used only TED corpus [27] plus a subset of 800K parallel sentences from the rest of the parallel data to train the neural models.[4] We concatenated dev- and test-2010 for tuning and used test2011-2013 for evaluation.

3.2. MT Settings

We trained a Moses phrase-based system [19] following the settings described in [28]: maximum sentence length of 80, Fast-align [29] for word-alignments, an interpolated Kneser-Ney smoothed 5-gram language model [30], lexicalized reordering [31] and a 5-gram OSM model [2]. We used k-best batch MIRA [32] for tuning.[5] We trained alternative baselines by adding OSM models trained on POS and word clus-

[4]Training models on the entire data required roughly 18 days of wall-clock time (18 hours/epoch on a Linux Ubuntu 12.04.5 LTS running on a 16 Core Intel Xeon E5-2650 2.00Ghz and 64Gb RAM) on our machines. We ran one baseline experiment with all the data and did not find it better than the system trained on randomly selected subset of the data. In the interest of time, we therefore reduced the NN training to a subset (1M).

[5]All systems were tuned twice.

German-English				
System	test11	test12	test13	Avg.
Baseline	35.0	30.3	27.1	30.8
OSM_{pos}	35.3	30.5	27.1	31.0
OSM_{mkcls}	35.1	30.1	26.8	30.7
OSM_{neural}	35.8	31.5	27.0	31.4
$Lex.reo_{neural}$	35.5	31.1	27.2	31.3
Lex_{neural}	35.3	30.8	26.9	31.0
English-German				
Baseline	25.7	21.7	23.4	23.6
OSM_{pos}	25.9	21.9	23.8	23.9
OSM_{mkcls}	25.8	21.8	23.4	23.7
OSM_{neural}	26.1	22.1	24.2	24.1
$Lex.reo_{neural}$	26.1	22.4	23.7	24.1
Lex_{neural}	26.0	22.2	23.7	24.0

Table 2: Comparing performance of Neural Reordering Models against N-gram-based Models. Quality measured in cased-bleu [34]

ters (50) obtained by running `mkcls` [6]. We used LoPar for German and MXPOST tagger for English POS tags. We trained 7-gram models to enable wider context than the regular word-based models.

3.3. NN Training

We trained our neural reordering models using NPLM[6] toolkit [14] with the following settings. We used a target context of 6 words (including reordering operations) and a corresponding source window of 7 words (also including reordering operations), forming a joint stream of 14-grams for training. We restricted source and target side vocabularies to 20K and 40K most frequent words. We used an input embedding layer of 150 and an output embedding layer of 750. Only one hidden layer is used with a Noise Contrastive Estimation[7] or NCE [33]. Training was done using mini-batch size of 1000 and using 100 noise samples. All models were trained for 25 epochs.

3.4. Results

Table 2 compares the results for our neural reordering models against baseline containing traditional reordering models. The baseline system is equipped with lexicalized and OSM model trained over word forms using count-based/n-gram-based models. We see that adding OSM models trained over generalized representation such as POS tags help slightly (+0.2 BLEU improvement in DE-EN and +0.3 in EN-DE). Using word clusters instead of POS tags did not help as much.

The next set of rows show results when using neuralized OSM and Lexicalized reordering models. The neural OSM model gave an improvement of +0.6 and +0.5 in DE-EN and EN-DE pairs. Neuralized lexical reordering performed almost as good as the neural OSM model suggesting that fine-grained reordering tags and hierarchical jumps add little value. The lexical sequence model without reordering tags (last row) performed lower (in the DE-EN pair) showing that there is some value in integrating reordering tags[8] during generation. In the EN-DE pair the difference is insignificant showing that much of the gains are coming from addressing lexical sparsity and not better reordering.

4. Neural Machine Translation

Neural Machine Translation [16, 35] is quickly becoming the predominant approach to machine translation. Rather than modeling different linguistic aspects (lexical generation, reordering, fertility etc.) as feature components and tuning them to optimize BLEU, NMT is trained in an end-to-end fashion. Given a bilingual sentence pair, we first generate a vector representation of the source sentence using `encoder` and then map this vector to target sentence using a `decoder`. The long distance source and target contextual dependencies are modeled using recurrent neural networks (RNN) with bilingual Long Short Term Memory (LSTM) [36]. The attention model [16] serves as an alignment model which enables the decoder to focus on different parts of the source as it generates the target sentence. Unlike phrase-based decoding, the reordering window is not limited to a frame of 6 words. This allows NMT to capture very long range reordering like syntax-based models [37].

In this work, we tried to explore whether explicitly integrating reordering triggers into the RNN-based `encoder` and `decoder`, improve the performance of the attentional model. We use the training data generated earlier (to train the neural OSM models – See Table 1), to train the `sequence-to-sequence` NMT model. This allows the `decoder` to condition on both lexical and reordering states when generating the new target word, which itself can be a word or a reordering operation. Our motivation was that such reordering triggers and pre-ordering of source[9] might help the attention mechanism with its task.

Note that the target sequence and alignments are both latent variables during decoding, we need to predict the pre-ordered (or reordering augmented sequence). To do this, we additionally train a source→pre-ordered (or reordering augmented) source sequence using another `sequence-to-sequence` model.

[6] http://nlg.isi.edu/software/nplm/

[7] Training neural language model with backpropagation could be prohibitively slow because for each training instance, the softmax layer requires a summation over the entire output vocabulary. One way to avoid this repetitive computation is to use a Noise Contrastive Estimation of the loss function.

[8] We also tried variations with reordering tags either on source or target side. The current variation with tags on both sides worked best.

[9] Remember that we linearize the source based on target using word-alignments

German-English				
System	test11	test12	test13	Avg.
Baseline	33.9	29.2	27.5	30.2
OSM	32.2	27.6	25.6	28.5
Lex.reo	29.2	24.8	22.8	25.6
Lex	30.8	26.6	23.9	27.1

Table 3: Training NMT systems with pre-ordered data, with lexical reo. operations, OSM operations

German-English				
System	test11	test12	test13	Avg.
OSM	45.7	42.0	36.6	41.4
Lex.reo	48.0	45.2	43.2	45.5
Lex	52.0	50.8	49.3	50.7

Table 4: Source to pre-ordered (or reordering augmented) system

4.1. System Settings

We trained a 2-layered LSTM encoder-decoder with attention. We used `seq2seq-attn` implementation [38] with the following settings: word vectors and LSTM states have 500 dimensions, SGD with initial learning rate of 1.0 and rate decay of 0.5, and dropout of 0.3. The MT systems are trained for 20 epochs, and the model with best dev loss is used for extracting features for the classifier.

4.2. Results

Table 3 shows the results from training NMT systems from pre-ordered data and using reordering augmented data. No gains were observed compared to the baseline system. In fact there was significant drop in all cases. One reason for this drop could be inaccuracy in predicting pre-ordered (reordering augmented) sequences. This can be seen in the BLEU scores shown in Table 4.[10] [39] also found pre-ordering the source-side in Neural MT deteriorated system performance in Japanese↔English and Chinese↔English pairs. They conjectured that pre-ordering introduces noise in terms of word-order hindering the learning process more difficult.

5. Related Work

A significant amount of research has been carried to alleviate data sparsity when translating into or from morphologically rich languages. [4] integrated different levels of linguistic information as factors into the phrase-based translation model. The idea of translating to stems and then inflecting the stems in a separate step has been studied by several researchers [40, 41]. POS tags are used in bilingual sequence models to enable wider context by [5, 22, 6]. Several researchers used word clusters in training data to obtain smoother distributions and better generalizations [8, 7, 9]. [42] used factors and parallel back-offs to address the issue of data sparsity. Continuous space models are used earlier for n-best re-ranking or directly as a feature in phrase-based MT [12, 13, 43, 44, 15]. [45] recently proposed an LSTM recurrent neural reordering model which directly models word pairs and their alignment. However, because SMT decoder requires fixed history, it is only possible to use the feature in the n-best re-ranking.

A whole new paradigm based on deep neural network evolved as a parallel framework for machine translation [16, 35]. The RNN-based sequence-to-sequence model learns generalized representations to overcome data sparsity problems and learn long distance dependencies successfully. This is further enhanced by using sub-word [46] or character-based models [47] to address the OOV-word problem. [18] has recently shown that integrating structural biases based on relative positions and fertilities improves the attention mechanism. [48] and [49] used side-constraints i.e. adding suffix tag at the end of the source sentence or prefix tag in the beginning of the target sentence to control the behavior of the decoder i.e. politeness in the case of former and domain in the latter. Our work is similar in a sense that we are trying to add reordering constraints, forcing the decoder to produce a specific reordering pattern. However, our method did not yield any improvements.

6. Conclusion

Traditional reordering models in phrase-based system suffer from data sparsity. In this paper, we presented neuralized versions of these reordering models (the OSM and Lexicalized reordering models) and used them as a feature in Phrase-based SMT. Our evaluation on German-English language pairs showed an improvement of up to 0.6 BLEU points. We also demonstrated gains compared to the previous solution where these models are trained on parts-of-speech tags and word clusters, to address data sparsity and for better generalization. The code will be pushed to Moses toolkit.[11] We also tried our pre-ordered and reordering augmented training data to train sequence-to-sequence neural MT models, with a motivation to explicitly add reordering triggers in the encoder representation and aid the attention mechanism. However, our modification to the natural source order and integration of reordering symbols in the training data, did not yield improvement.

7. Acknowledgements

We would like to thank the anonymous reviewers for their useful feedback.

[10] The BLEU scores are computed using pre-ordered (or reordering augmented) references generated using word-alignments of original source-target evaluation sets.

[11] http://www.statmt.org/moses/

8. References

[1] C. Tillman, "A unigram orientation model for statistical machine translation," in *HLT-NAACL 2004: Short Papers*, D. M. Susan Dumais and S. Roukos, Eds., Boston, Massachusetts, USA, May 2004, pp. 101–104.

[2] N. Durrani, A. Fraser, H. Schmid, H. Hoang, and P. Koehn, "Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT?" in *Proceedings of ACL 2013*, Sofia, Bulgaria, August 2013, pp. 399–405.

[3] H. Zhang, K. Toutanova, C. Quirk, and J. Gao, "Beyond left-to-right: Multiple decomposition structures for smt," in *Proceedings of NAACL-HLT 2013*, Atlanta, Georgia, June 2013, pp. 12–21.

[4] P. Koehn and H. Hoang, "Factored Translation Models," in *Proceedings of the Joint EMNLP-CoNLL 2007*, Prague, Czech Republic, June 2007, pp. 868–876.

[5] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Proceedings of the WMT-11*, Edinburgh, Scotland, July 2011, pp. 198–206.

[6] N. Durrani, P. Koehn, H. Schmid, and A. Fraser, "Investigating the usefulness of generalized word representations in smt," in *Proceedings of COLING 2014*, Dublin, Ireland, August 2014, pp. 421–432.

[7] V. Chahuneau, E. Schlinger, N. A. Smith, and C. Dyer, "Translating into morphologically rich languages with synthetic phrases," in *Proceedings of the EMNLP 2013*, Seattle, USA, October 2013, pp. 1677–1687.

[8] J. Wuebker, S. Peitz, F. Rietig, and H. Ney, "Improving Statistical Machine Translation with Word Class Models," in *Proceedings of the EMNLP 2013*, Seattle, USA, October 2013, pp. 1377–1381.

[9] A. Bisazza and C. Monz, "Class-based language modeling for translating into morphologically rich languages," in *Proceedings of COLING 2014*, Dublin, Ireland, August 2014, pp. 1918–1927.

[10] 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.

[11] P. F. Brown, P. V. deSouza, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, pp. 467–479, 1992.

[12] H. Schwenk, "Continuous space translation models for phrase-based statistical machine translation," in *Proceedings of COLING 2012*, Mumbai, India, Dec 2012.

[13] H.-S. Le, A. Allauzen, and F. Yvon, "Continuous space translation models with neural networks," in *Proceedings of the NAACL-HLT 2012*, Montréal, Canada, June 2012, pp. 39–48.

[14] A. Vaswani, Y. Zhao, V. Fossum, and D. Chiang, "Decoding with large-scale neural language models improves translation," in *Proceedings of the EMNLP 2013*, Seattle, USA, October 2013, pp. 1387–1392.

[15] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *Proceedings of the ACL 2014*, Baltimore, USA, June 2014, pp. 1370–1380.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015. [Online]. Available: http://arxiv.org/pdf/1409.0473v6.pdf

[17] N. Durrani, H. Schmid, and A. Fraser, "A Joint Sequence Translation Model with Integrated Reordering," in *Proceedings of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT'11)*, Portland, OR, USA, 2011.

[18] T. Cohn, C. D. V. Hoang, E. Vymolova, K. Yao, C. Dyer, and G. Haffari, "Incorporating structural alignment biases into an attentional neural translation model," in *Proceedings of the NAACL-HLT 2016*, San Diego, California, June 2016, pp. 876–885.

[19] 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 ACL 2007*, Prague, Czech Republic, 2007.

[20] 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 *Proceedings of the International Workshop on Spoken Language Translation (IWSLT'05)*, Pittsburgh, PA, USA, 2005.

[21] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *Proceedings of EMNLP*, Honolulu, Hawaii, Oct 2008, pp. 848–856.

[22] J. M. Crego and F. Yvon, "Improving reordering with linguistically informed bilingual n-grams," in *In Proceedings of Coling 2010*, Beijing, China, August 2010, pp. 197–205.

[23] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[24] M. Nagata, K. Saito, K. Yamamoto, and K. Ohashi, "A clustered global phrase reordering model for statistical machine translation," in *Proceedings of COLING 2006*, Sydney, Australia, July 2006, pp. 713–720.

[25] P. Li, Y. Liu, M. Sun, T. Izuha, and D. Zhang, "A neural reordering model for phrase-based translation," in *Proceedings of COLING 2014*, Dublin, Ireland, August 2014, pp. 1897–1907.

[26] 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.

[27] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 11th IWSLT Evaluation Campaign," *Proceedings of the IWSLT, Lake Tahoe, US*, 2014.

[28] A. Birch, M. Huck, N. Durrani, N. Bogoychev, and P. Koehn, "Edinburgh SLT and MT system description for the IWSLT 2014 evaluation," in *Proceedings of the 11th International Workshop on Spoken Language Translation*, ser. IWSLT '14, Lake Tahoe, CA, USA, 2014.

[29] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," in *Proceedings of NAACL'13*, 2013.

[30] K. Heafield, "KenLM: Faster and Smaller Language Model Queries," in *Proceedings of the Sixth WMT*, Edinburgh, Scotland, UK, July 2011, pp. 187–197.

[31] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," in *Proceedings of the tenth Machine Translation Summit*, Phuket, Thailand, 2005.

[32] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *Proceedings of NAACL 2012*, Montréal, Canada, 2012, pp. 427–436.

[33] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of AISTATS*, ser. JMLR W&CP, vol. 9, 2010, pp. 297–304.

[34] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of ACL 2002*, Philadelphia, PA, USA, 2002, pp. 311–318.

[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer, "Scalable inference and training of context-rich syntactic translation models," in *Proceedings of the ACL 2006*, Sydney, Australia, July 2006.

[38] Y. Kim, "Seq2seq-attn," https://github.com/harvardnlp/seq2seq-attn, 2016.

[39] J. Du and A. Way, "Pre-reordering for neural machine translation: Helpful or harmful?" *prague bulletin of mathematical linguistics*, vol. 108, pp. 171–182, 2017.

[40] K. Toutanova, H. Suzuki, and A. Ruopp, "Applying Morphology Generation Models to Machine Translation," in *Proceedings of ACL-08: HLT*, Columbus, Ohio, June 2008, pp. 514–522.

[41] A. Fraser, M. Weller, A. Cahill, and F. Cap, "Modeling Inflection and Word-Formation in SMT," in *Proceedings of EACL 2012*, Avignon, France, April 2012, pp. 664–674.

[42] Y. Feng, T. Cohn, and X. Du, "Factored markov translation with robust modeling," in *Proceedings of CoNNL 2014*, Ann Arbor, Michigan, June 2014, pp. 151–159.

[43] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of EMNLP 2013*, Seattle, USA, October 2013, pp. 1700–1709.

[44] J. Gao, X. He, W.-t. Yih, and L. Deng, "Learning continuous phrase representations for translation modeling," in *Proceedings of ACL 2014*, Baltimore, Maryland, June 2014, pp. 699–709.

[45] Y. Cui, S. Wang, and J. Li, "Lstm neural reordering feature for statistical machine translation," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 977–982. [Online]. Available: http://www.aclweb.org/anthology/N16-1112

[46] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," *arXiv preprint arXiv:1508.07909*, 2015.

[47] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware Neural Language Models," *arXiv preprint arXiv:1508.06615*, 2015.

[48] R. Sennrich, B. Haddow, and A. Birch, "Controlling politeness in neural machine translation via side constraints," in *Proceedings NAACL-HLT 2016*, San Diego, California, June 2016, pp. 35–40.

[49] C. Kobus, J. M. Crego, and J. Senellart, "Domain control for neural machine translation," *CoRR*, vol. abs/1612.06140, 2016. [Online]. Available: http://arxiv.org/abs/1612.06140

Data Selection with Cluster-Based Language Difference Models and Cynical Selection

Lucía Santamaría, Amittai Axelrod

Amazon.com

{lucsan,amittai}@amazon.com

Abstract

We present and apply two methods for addressing the problem of selecting relevant training data out of a general pool for use in tasks such as machine translation. Building on existing work on class-based language difference models [1], we first introduce a cluster-based method that uses Brown clusters to condense the vocabulary of the corpora. Secondly, we implement the cynical data selection method [2], which incrementally constructs a training corpus to efficiently model the task corpus. Both the cluster-based and the cynical data selection approaches are used for the first time within a machine translation system, and we perform a head-to-head comparison.

Our intrinsic evaluations show that both new methods outperform the standard Moore-Lewis approach (cross-entropy difference), in terms of better perplexity and OOV rates on in-domain data. The cynical approach converges much quicker, covering nearly all of the in-domain vocabulary with 84% less data than the other methods.

Furthermore, the new approaches can be used to select machine translation training data for training better systems. Our results confirm that class-based selection using Brown clusters is a viable alternative to POS-based class-based methods, and removes the reliance on a part-of-speech tagger. Additionally, we are able to validate the recently proposed cynical data selection method, showing that its performance in SMT models surpasses that of traditional cross-entropy difference methods and more closely matches the sentence length of the task corpus.

1. Data Selection, Previously

1.1. Moore-Lewis Data Selection

The standard data selection method of Moore and Lewis [3] uses cross-entropy difference as the similarity metric to estimate the relevance of each sentence in the general pool corpus. This method takes advantage of the presumed mismatch between the pool data and the task domain. It first trains an in-domain language model (LM) on the task data, and then trains another LM on the full pool of general data. The average per-word perplexity of each sentence in the pool data is computed relative to each of these models. The cross-entropy $H_{lm}(s)$ of a sentence s, according to language model lm, is the log of the perplexity of the language model on that sentence. The cross-entropy difference score of [3] is:

$$H_{LM_{TASK}}(s) - H_{LM_{POOL}}(s).$$

Sentences that are most like the task data, and most unlike an average sentence in the full pool will have lower cross-entropy difference scores. A modification of this method, the bilingual Moore-Lewis criterion [4] used for selecting bilingual data for machine translation. This is a simple extension, combining the cross-entropy difference scores from each side of the corpus; i.e. for sentence pair $\langle s_1, s_2 \rangle$

$$\begin{aligned} & \left(H_{LM_{TASK_1}}(s_1) - H_{LM_{POOL_1}}(s_1)\right) \\ + \quad & \left(H_{LM_{TASK_2}}(s_2) - H_{LM_{POOL_2}}(s_2)\right). \end{aligned}$$

For both the regular and bilingual Moore-Lewis methods, data selection is performed by sorting the sentences according to the corresponding criterion and picking the top n sentences (or sentence pairs). Determining the optimal value of n is typically done empirically, training systems on subsets of increasing size, and evaluating on a held-out set.

1.2. Class-based Language Difference Models for Data Selection

The cross-entropy difference method can be improved by using language difference models (LDMs) instead of normal language models to compute the cross-entropy scores [1]. The standard and bilingual Moore-Lewis data selection methods use n-gram language models to calculate the cross-entropy difference scores needed to rank sentences in the data pool. However, this creates a structural mismatch in the algorithm. The standard language models used in the computation are *generative* models; they can be used to predict the next word. Yet, the actual cross-entropy difference score is *discriminative* in nature, because it asks: is the sentence more like the task corpus, or more like the pool corpus?

This conceptual gap is well-known, and has led to data selection approaches that use classifiers to determine domain membership. However, to build a classifier is to fall into a trap! Only the Moore-Lewis score is discriminative; the underlying corpora themselves are not. This is readily seen by noting that a sentence can appear in both the task and the pool corpora without any contradiction: "task-ness" and "pool-ness" are defined by construction rather than by any inherent characteristic. The two could overlap by 1%, or by 99%, and they would still be two corpora labeled 'task' and 'pool'.

It may help to reframe the 'task' corpus as "a pile of data that we already know we like", and the 'pool' corpus as "a pile of data about which we do not yet have an opinion". It is not necessary to know why we like the data in the task corpus; it is also not necessary to have any opinion about whether the pool data looks useful, or not. With this view, the two corpora are not in opposition. Because they do not form opposing ends of a spectrum, then there is no underlying "in-domain vs out-of-domain" classification problem.[1]

We previously defined a discriminative representation of the corpus as one that explicitly marks how the corpora differ. This helps quantify the difference between the task and the pool corpora. In [1], every word in the corpora was replaced by a synthetic tag consisting of a class label and a discriminative marker. This procedure led to a representation of the text that explicitly encoded language differences between the corpora. Once the text had been transformed, the regular Moore-Lewis cross-entropy difference method is applied: two "language models" are trained on the task and the pool. As the representation is discriminative, we have snuck discriminative information into the generative framework of the language models, so the two models are actually language difference models. Each sentence is then scored with the two models, and the scores are subtracted and used to sort the data pool and select the top n lines. The bilingual version of class-based language difference models is exactly the same as bilingual Moore-Lewis: the corpus representation has changed, but the algorithm has not.

The tags in that work combined part-of-speech (POS) tags plus a suffix indicating the relative bias of each word. Both they and [5] showed improved translation results when using the class-based difference labels to train the models for cross-entropy difference computation, instead of just using the words themselves. A variationused 20 class labels derived from an unsupervised POS tagger to create the language difference model [6] , but they did not obtain positive results when selecting monolingual data for back-translation and then subsequently using the artificially-parallel data to train a neural MT system.

[1] We still use 'in-domain' and 'task' interchangeably.

2. Proposed Methods

2.1. Cluster-Based Language Difference Models

Using POS tags, as the basis for the discriminative tags that reduce the lexicon, creates a dependency on such a part-of-speech tagger. Such a tool is not always reliable, nor even available, for many languages nor specialized kinds of language. [1] posited that other methods of creating classes that capture underlying relationships within sentences (such as clustering or topic labels) might yield similar improvements.

Following that hypothesis, we experimented with data selection using class-based language difference models. The synthetic difference representations were created using Brown cluster labels (generated from all of the words in the corpora) plus a relative-bias qualifier. Brown clustering [7] is a way of partitioning a lexicon into classes according to the context in which the words occur in a corpus. Context, in this case, means the distribution of the words to their immediate left and right. The process of creating the clusters also generates a hierarchy above them, in the form of an unbalanced binary tree. Each word is assigned a bit string, and words that are statistically similar with respect to their neighbors will have similar bit strings and thus will be close together in the tree. An advantage of this method is that the number of clusters is freely specifiable, with a theoretical maximum of V, the size of the vocabulary. Choosing the correct amount is important, as too low a number would lead to poor-quality clusters, but generating a high number of them is computationally expensive.

Following standard practice, we chose 1,000 as the number of clusters and added a suffix to indicate how much more likely a word is to appear in the task than in the pool corpus. Consistent with Table 1 of [1], we binned the probability ratios by order of magnitude (powers of e), from e^{-3} to e^{3}. We indicated $e^{-3} < x < e^{-2}$ with the suffix "−−", $e^{1} < x < e^{2}$ as "+", and so on. The following is an example of the text's new discriminative representation:

Original	*massive biotische krisen ... in vulkanen , gletschern , ozeanen .*
Transformed	`682/0 UNK/+ 935/0 3/- 7/0 890/0 1/0 890/0 1/0 862/+ 2/0`

The number before the slash indicates the cluster ID, and the marker after it represents the first digit of the log (*i.e.* exponent) of the ratio of the word's probability to appear in the task corpus divided by its probability in the pool corpus.

Our class-based language difference model representation condensed the vocabulary of each of the corpora by at least 97%. Table 1 contains the sizes of the corpus vocabularies before and after the cluster-based reduction.

It is on this transformed text that the language difference models were trained and the cross-entropy difference scores computed. After ranking and selecting, the sentences were

Corpus	Word Types	Condensed	Reduction
Task (DE)	93,767	1,691	-98.20 %
Task (EN)	53,284	1,562	-97.07 %
Pool (DE)	1,135,226	2,570	-99.77 %
Pool (EN)	894,270	2,375	-99.73 %

Table 1: Size of the vocabularies that form the representation of the corpora used by the language difference models.

transformed back to the original words and the MT systems were trained as usual.

2.2. Cynical Data Selection

The Moore-Lewis cross-entropy difference method has proved enduring, despite the subsequent development of several other methods with slightly better performance. Cross-entropy difference has had the advantage of being intuitive, reasonably effective, and easy to implement and integrate into existing MT pipelines. That said, it also has some structural problems.

Its subtractive relevance score implicitly defines the task and pool corpora as being opposing ends of a single spectrum: if the in-domain LM likes a sentence, it must be good, and if the pool LM likes it, then the sentence is irrelevant. This is never true, because language does not decompose cleanly into disjoint subsets, much less disjoint domains nor topics. The cross-entropy difference method is particularly weak when the task and pool corpora are similar, because the scores cancel out. Furthermore, the cross-entropy difference score indicates onlythat the selected sentences are well-liked by the in-domain model. It does not know whether the sentences are known to actually help model the in-domain data, nor if they even cover the in-domain vocabulary.

Cynical data selection [2] is a recent method to incrementally construct an efficient training corpus that models the in-domain corpus as closely as possible. Each sentence is scored by how much it would help model a particular task corpus, if it were added to the existing training corpus at the current iteration. The core idea was described as "an incremental greedy selection scheme based on relative entropy, which selects a sentence if adding it to the already selected set of sentences reduces the relative entropy with respect to the in-domain data distribution" [8].

Cynical selection[2] is an iterative algorithm that keeps track of how well the currently-selected data can model the task data. This is done via measuring the perplexity of a unigram LM trained on the selected sentences and evaluated on the in-domain corpus. The method iterates through all the words in the lexicon, and computes the expected entropy gain from adding a single instance of that word to the selected data. This step enables the algorithm to depend on the number of words in the lexicon rather than the number of sentences in the pool. The best word (that lowers entropy the most) is chosen. Given that word, the algorithm iterates through all the available (un-picked) sentences containing that word, and computes the expected entropy change from adding that single sentence by itself to the previously-selected set. The sentence with the most negative change is added, and the task perplexity is recomputed, taking into account the sentence that was just selected.

[2] Sentences are only selected if they are of provable utility, regardless of whether an in-domain LM would like it, hence the name.

3. Experimental Setup

3.1. Data

We experimented on the German-to-English parallel data from the MT evaluation campaign for IWSLT 2017[3]. Our task data was the TED Talks corpus [9], comprising 218k parallel training sentences. The pool of available data consisted of 17.6M parallel sentences assembled from multiple sources: the preprocessed dataset from the WMT 2017 translation task[4] (containing the Europarl v7, Common Crawl and News Commentary corpora) and the OpenSubtitles2016 collection[5]. We tuned on `dev2010` and tested on the concatenation of the `test2010`, `test2011`, `test2012`, `test2013`, `test2014`, and `test2015` datasets released for IWSLT 2017.

All corpora were preprocessed with the standard Moses [10] tools following the same pipeline employed in the preparation of the WMT 2017 preprocessed MT data[6]. The sizes of the resulting datasets are in Table 2.

Corpus	Contents	Sentences	Tokens (DE)
Task	TED Talks	218,020	4.0 M
Pool	WMT17	5,852,458	134.8 M
	OpenSubtitles 2016	11,811,574	100.1 M
	Total	17,664,032	235 M
Tune	`dev2010`	920	19.3 k
Test	`test2010-2015`	8,431	154.8 k

Table 2: German-English parallel data statistics.

[3] `https://sites.google.com/site/iwsltevaluation2017/data-provided`

[4] `http://www.statmt.org/wmt17/translation-task.html`

[5] `http://opus.lingfil.uu.se/OpenSubtitles2016.php`

[6] `http://data.statmt.org/wmt17/translation-task/preprocessed/de-en/prepare.sh`

3.2. SMT Training Pipeline

We trained our models with a statistical machine translation pipeline built upon a combination of open-source tools. Input data was further subjected to various normalizations, such as lowercasing, diacritic normalization, and the standardization of quotation marks. We split compound nouns on the German input using `ASVToolBoox` [11].

Translation was done with the `Joshua` decoder [12], an implementation of hierarchical phrase-based statistical machine translation. In some experiments, an additional target-side background language model was used while decoding, to promote fluent output and provide a more realistic use case. Word alignments were learned using `fast_align` [13] with alignment models estimated in both directions and symmetrized using *grow-diag-final-and* [14]. Grammar extraction was performed using the open-source framework `Thrax`[7] and run on potent Elastic MapReduce (EMR) clusters during training. Tuning was done with the Margin Infused Relaxed Algorithm (MIRA) [15] and optimized on BLEU [16].

3.3. Data Selection Tools

The standard Moore-Lewis method uses n-gram language models to compute the cross-entropy score of each sentence according to the task and pool LMs. We used `kenlm` [17] to estimate 6-gram Kneser-Ney (KN) smoothed language models, padding the vocabulary to 1.5M.

Our implementation of class-based difference models used Brown clusters and unigram frequency ratios to automatically produce discriminative representations of the task and pool corpora. These followed the steps in [1], with every word replaced by a new token consisting of a cluster label and a bias suffix. We employed the unsupervised Brown clustering algorithm [7] for the construction of the clusters. The label+suffix tokens explicitly show how and where the two corpora's distributions differ from each other. We then trained 6-gram KN language models on this new representation. These models were used to compute cross-entropy difference scores over the new representation, and then the sentences were sorted by score. The discriminative representations were replaced by the original sentences after the selection process completed.

We wrote (and released[8]) an open-source implementation of the cynical selection algorithm. Reducing the vocabulary size, by collapsing words into a single label, makes the algorithm's approximations tractable. We used the algorithm's default heuristics for vocabulary reduction, shown in Table 3. Each criterion was applied (in order) to every word v in the joint lexicon of the corpora. If the criterion was met, the word was replaced, and no further criteria were applied to the word. After all the criteria were applied, most of the vocabulary types had been collapsed down to a handful of labels, and the only words that remained intact were ones whose probability ratios were biased towards the task distribution.

Criteria	Word Types	Replaced By
$C_{\text{TASK}}(v) = 0$	1,050,590	`__useless`
$C_{\text{POOL}}(v) = 0$	9,131	`__impossible`
$C_{\text{TASK}}(v) < 3$ AND $C_{\text{POOL}}(v) < 3$	4,946	`__dubious`
$\frac{P_{\text{TASK}}(v)}{P_{\text{POOL}}(v)} < e^{-1}$	3,945	`__bad`
$e^{-1} < \frac{P_{\text{TASK}}(v)}{P_{\text{POOL}}(v)} < e$	22,453	`__boring`

Table 3: Criteria used to reduce the German lexicon from 1.14M to 29k for cynical data selection, baseed on the counts C and probabilities P for each word. Only the first criterion to match each word was applied, the word then being replaced by the corresponding tag.The second column shows how many word types were replaced by each rule.

The intuition for the replacements is as follows: If a word v does not appear in the task corpus, then it is `useless` for estimating relevance because it does not figure into the entropy calculations. If a word is in the task corpus but not in the pool, then it is `impossible` to change its empirical probability by adding sentences from the data pool. The probability of rare words (occurring once or twice in both corpora) cannot be estimated reliably, so their statistics are `dubious`. Words that are heavily skewed towards the pool distribution are `bad` for determining usefulness or information gain, because there is a danger that they will be overrepresented in the selected sentences. We also tried appending a bias suffix to the `bad` label, following the procedure from the class-based language difference model approach.

Even if we selected sentences randomly, we can expect to accurately estimate the probabilities of words occurring at roughly the same rate in both corpora, so their probability ratios are `boring`. We experimented with further dividing this category based on the frequency of these words in the task corpus, with the goal of limiting the number of sentences in which each token appears. This is important because the cynical algorithm implementation avoids computational complexity in terms of the number of sentences by replacing it with computational complexity in terms of the number of sentences in which words appear. However nearly every sentence in the pool contained a `boring` word, and it is not clear that this had any effect.

Due to the size of the pool corpus, we enabled the cynical data selection's "batchmode", where a variable amount ($\log k$) of sentences are selected per iteration. This variable batch size is computed from k, the number of sentences that contain the "most useful word" for the current iteration.

[7] `https://github.com/joshua-decoder/thrax`
[8] `https://github.com/amittai/cynical`

4. Experiments and Results

We evaluated three data selection approaches: standard Moore-Lewis (monolingual and bilingual), class-based language difference models using Brown clusters (also monolingual and bilingual), and cynical data selection (monolingual only). The monolingual methods were used on each of the input and output languages, so we have results for all methods on both languages. Our contribution is the first published use of Brown clusters for class-based language difference models, and also of cynical data selection. We present a head-to-head comparison of both, as well as comparing against the cross-entropy difference standards.

Each data selection method produced a subset of the pool corpus in which sentences are ranked by their relevance. The first four assign an absolute relevance score (some variation on cross-entropy difference) for each sentence. The cynical method provides a ranking, but the score for each sentence is the relative relevance score of each sentence with respect to all the higher-ranked sentences that precede it. For each experiment, we examined increasingly larger slices of the data ranging from the best $n = 100$k to the best $n = 12$M sentences out of the 17.6M sentence pairs available.

4.1. Perplexity of Modeling In-Domain Data

In all cases, we first evaluated the selected data by itself, examining how well the selected data can model the task data. For this, we measured the perplexity and OOV count on the in-domain corpus, using models trained on only the selected data. For each of the data selection methods, we trained language models on the most relevant subsets of various sizes. The language models were similar to those used for selection (n-gram order 4, and vocabulary padded to 1.5M). We evaluated these models on their perplexity on the entire in-domain TED training set (218k sentences). Figure 1 and 2 show the full language modeling perplexity results for the input (German) and output (English) languages, respectively.

Each of the cluster-based data selection methods on the German side outperformed their vanilla Moore-Lewis counterparts (comparing monolingual cluster-based vs monolingual Moore-Lewis, both on the German side, and comparing bilingual cluster-based vs bilingual Moore-Lewis). At 6M sentences selected, near convergence, the cluster-based methods are each 20 perplexity points better than the standard cross-entropy difference, but the cynical selection method is slightly better. At 2M sentences selected, where the cynical method is nearly at its optimal perplexity, the gap between the cluster-based and standard approaches is 40 points, but the cynical method is 20 points better still, as highlighted in Table 4, despite being a monolingual method.

On the English side, the improvements are similar in pattern though smaller in magnitude. This is expected, as English is easier to model than German. The cluster-based methods significantly outperform the regular Moore-Lewis methods. The cynical method once again converges the fastest of all the methods (after 2M sentences, compared to 6M for the others), though the cluster-based methods reach the lowest perplexity.

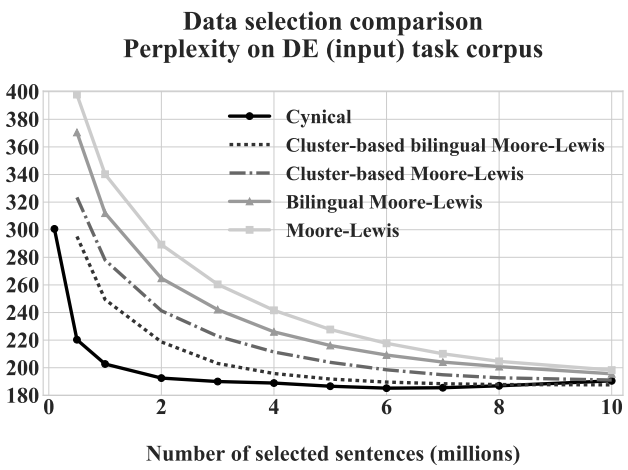


Figure 1: Evaluating the selected data via perplexity scores on the TED DE (input) corpus. The cluster-based methods are better than their standard counterparts, and the cynical method is better still.

Method	ppl at 2M	ppl at 6M
Moore-Lewis, mono (DE)	289.2	217.7
Cluster-based, mono (DE)	241.3	198.5
Moore-Lewis, bilingual	264.8	209.2
Cluster-based, bilingual	218.7	189.6
Cynical, mono (DE)	192.5	185.2

Table 4: Perplexity scores on the TED DE (input) corpus for the models trained with 2M and 6M selected sentences.

4.2. Out-of-Vocabulary Rate on In-Domain Data

Next, we computed the out of vocabulary (OOV) token count on the task corpus, using language models trained on only the selected data. Figures 3 and 4 show the OOV curves for the selected data with respect to the roughly 4M-token TED corpus in the input (German) and output (English) languages, respectively. In both graphs, the cluster-based Moore-Lewis methods converge to their final OOV count after selecting 6 to 8 million sentences. At the 6M sentence mark, the cluster-based methods have one-third fewer OOV tokens in the TED corpus than the vanilla Moore-Lewis methods. This substantial improvement corroborates the results from the method as proposed in [1].

However, the OOV rate of data selected using the new cynical data selection method is better still, by a large margin. At 1M sentences, the cynical subset has 85% fewer

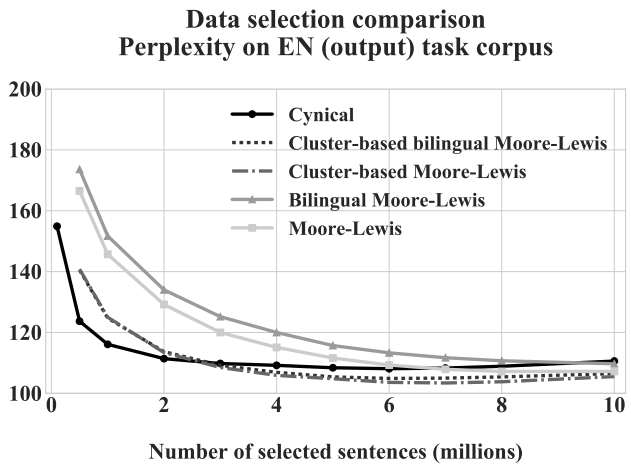


Figure 2: Comparison of perplexity scores on the TED EN (output) corpus. The new methods perform better than the standard approaches.

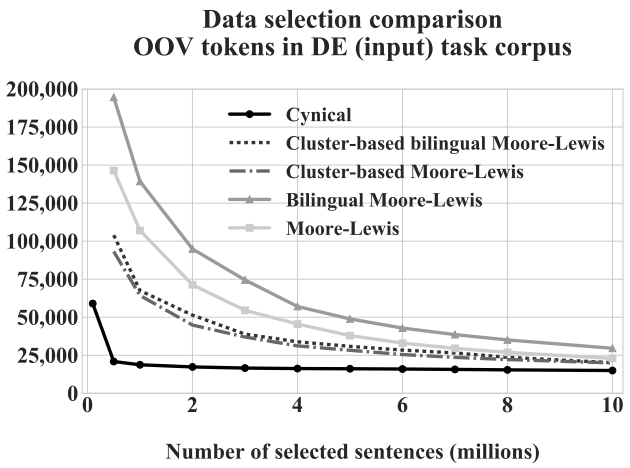


Figure 3: Number of OOV tokens in the TED DE (input) corpus, according to LMs trained on the selected data.

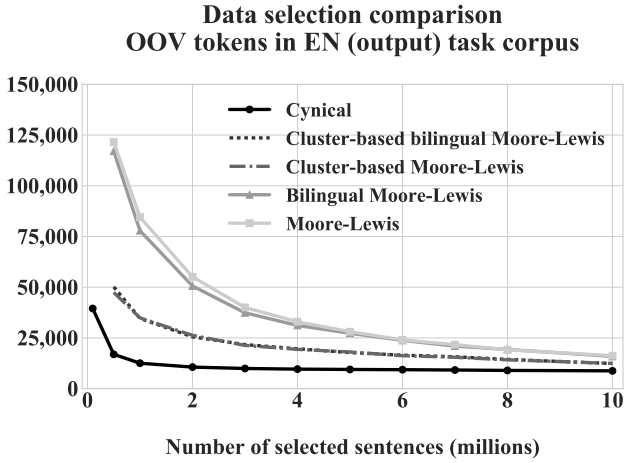


Figure 4: Number of OOV tokens in the TED EN (output) corpus, according to LMs trained on the selected data.

OOVs than the monolingual Moore-Lewis, and 65% fewer than the monolingual cluster-based version. More importantly, the first million sentences selected via the cynical method cover more of the task vocabulary than any quantity of data selected via the other methods. This rapid convergence to the minimum possible OOV rate – the number of OOV tokens relative to all of the pool data – results from the heuristic used by the cynical algorithm to select the word that needs to be covered by the next selected sentence.

4.3. Improving an In-Domain System with Selected Data

Next, we performed an extrinsic evaluation, using the selected data to train machine translation models to be used in combination with the baseline in-domain system. In this way we tested the ability of the data selection methods to select subsets of the data that were actually useful in practice.

Figure 5 shows the machine translation results using BLEU. The horizontal dashed line is a static baseline that uses all of (and only) the available in-domain training data. The other curves are from multi-model systems where a model trained on selected data is used in combination with one trained on the task data. Each system curve in Figure 5 shows the average score over 3 tuning and decoding runs, to mitigate the variability of MT tuning.

The baseline adapted systems using data selected via vanilla monolingual Moore-Lewis and the bilingual version performed better than the in-domain-only system, as expected. The difference between the monolingual and the bilingual versions' scores were minor, with the bilingual versions slightly ahead. The cluster-based versions of Moore-Lewis, which used language difference models to compute the cross-entropy difference scores, were roughly half a point better than the standard versions. The cynical methods performed as well as the fine-grained cluster-based approaches, despite collapsing approximately 850,000 vocabulary items down to a dozen coarse labels.

BLEU scores corresponding to models trained with 2M and 6M selected sentences are compared in Table 5. These results demonstrate that completely automatic clustering methods can be used to construct language difference models, so class-based version of cross-entropy difference need not depend on the availability of linguistically-derived labels.

As a further test, we examined the use of these selected corpora inside a more robust system: one that has both in-domain parallel data, and a large background target-side language model. Figure 6 shows the BLEU scores of multi-model systems that also incorporate a large background language model for decoding.

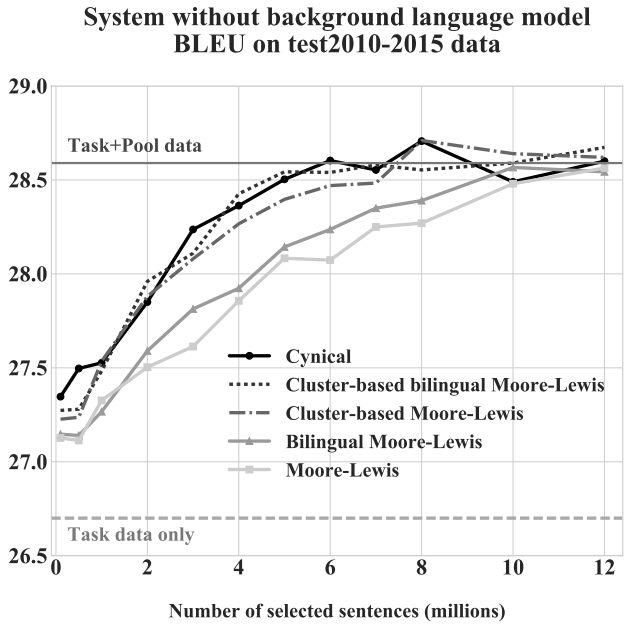


Figure 5: Comparison of BLEU scores on multi-model systems using both task data and subcorpora (from 100k to 12M sentences) selected via each data selection method. The dashed line corresponds to a system trained on the task data only. The thin solid line indicates the result (28.59) obtained from training with the entire pool of 17.6M sentences.

Table 6 shows numeric values of the scores at 2M and 6M selected sentences, computed by averaging 2 training runs with target-side language model for fluency. Incorporating the background LM leads to overall score increases of +1 to +2 BLEU points compared to the results in Figure 5. Again, the cluster-based extension of Moore-Lewis outperforms the vanilla version. However, the cynical selection method exhibits bimodal performance: For smaller amounts of selected data, up to 4M sentences, it follows the performance of the class-based methods. After that, it switches sharply to tracking the performance of the vanilla methods. The gap is not large, so it might be due to jitter from tuning, but it is curious.

4.4. Better Matching of In-Domain Sentence Length

One of the advantages of data selection is that it allows for significantly smaller translation systems that perform at least as well as one trained on the full large-scale pool corpus. This holds true for all of the methods compared in this work: The size reduction of the translation systems is roughly proportional overall to the reduction in training corpus size. However, we noticed that the translation systems trained on the cynical subcorpora are twice as large as the ones selected using cross-entropy difference variants. We discovered that the Moore-Lewis style selection methods produced subcorpora that were almost identical in size, both on disk and in the number of tokens. The cynical method produced subsets con-

Method	BLEU at 2M	BLEU at 6M
Moore-Lewis, mono (DE)	27.50	28.07
Cluster-based, mono (DE)	27.88	28.47
Moore-Lewis, bilingual	27.59	28.24
Cluster-based, bilingual	27.96	28.54
Cynical, mono (DE)	27.85	28.60

Table 5: BLEU scores on preprocessed data at 2M and 6M selected sentences from averaging 3 runs for a system, configured without background language model.

Method	BLEU at 2M	BLEU at 6M
Moore-Lewis, mono (DE)	28.55	29.61
Cluster-based, mono (DE)	29.19	29.90
Moore-Lewis, bilingual	28.72	29.56
Cluster-based, bilingual	29.34	29.93
Cynical, mono (DE)	29.33	29.69

Table 6: BLEU scores on preprocessed data at 2M and 6M selected sentences from averaging 2 runs using the more robust configuration with background language model.

taining significantly longer sentences than the other methods. Upon examination the sentences seemed fairly ordinary (*i.e.* normal sentences, not particularly long), and it was the other methods that were producing significantly shorter sentences. Figure 7 shows how the average sentence length changes with the number of sentences selected for each method.

The in-domain average sentence length is 19 tokens per sentence, and the pool corpus average is 14. All of the Moore-Lewis variants have average sentence lengths even shorter than the pool average, and never greater. The cynical method mostly selected pool data matching the task average sentence length, despite having no explicit way to note the length of the selected sentences. This appears to substantiate the assertion that "the length biases of the penalty and the gain terms counteract each other, guarding the algorithm from the Moore-Lewis method's fixation on one-word sentences with a very common token" [2].

5. Conclusions

We have shown that both cluster-based language difference models and cynical data selection can be used to train better task-specific machine translation systems and more closely model a task corpus. This is the first published use of both methods. Using Brown clusters instead of POS tags makes

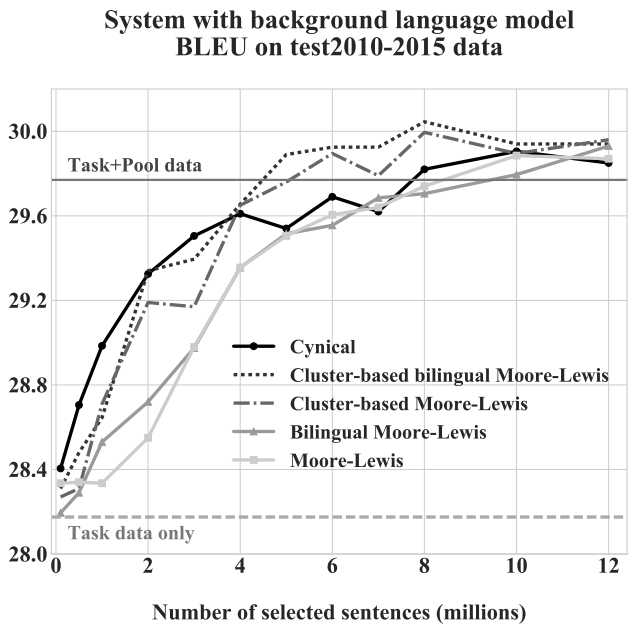


Figure 6: Same as Figure 5 but for the system with background language model. The BLEU score for the configuration with the entire pool data set is 29.77.

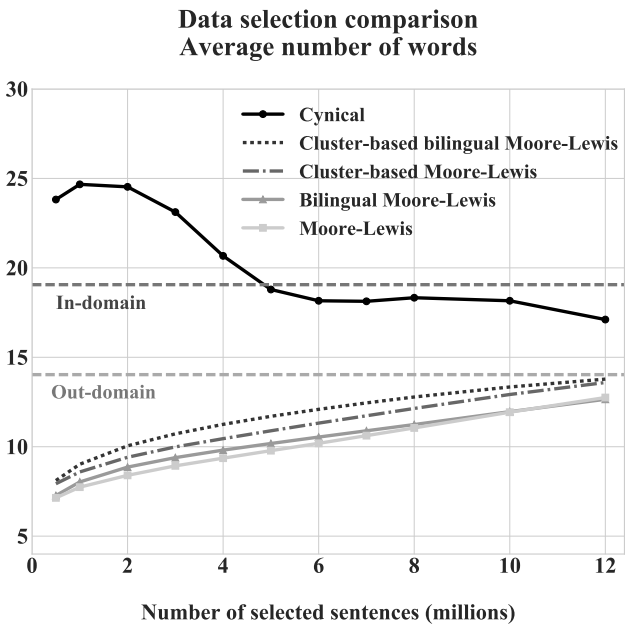


Figure 7: Average number of tokens per sentence, as a function of selected corpus size.

the language difference model variant of Moore-Lewis be both language- and situation-agnostic. We have not compared the two directly, but have shown that automatic clustering can be used successfully. This is good for domain adaptation scenarios where the particular kind of language is either low-resource or wildly different from the kind of data used to train NLP tools.

Also, we have presented empirical validation of the cynical selection method. Despite some of the crude algorithmic choices (4 labels for 97% of the lexicon) as well as running in batch mode, the cynical selection method's performance matches the best variant of Moore-Lewis. Further improvements might well be possible with more fine-grained labels (e.g. adopting the Brown clustering labels from this work). The cynical method, as implemented, converges after roughly 66% less data has been selected, compared to any of the cluster-based and vanilla Moore-Lewis methods, and has the best out-of-vocabulary word coverage.

The tradeoff is that while cynical selection picks better sentences, leading to smaller selected corpora, it also uses significant amounts of RAM (60gb for 17m sentences) and time (1 day; after all, $n \log n$ is still super-linear in complexity). Our implementation is inefficient, but the memory requirements will always be larger than the class-based language difference model version of Moore-Lewis which was developed to reduce the run-time requirements for data selection. This is because cynical selection must store the entire pool corpus in memory, whereas the reduced lexicon of the class-based approach means the algorithm runs in roughly constant space. Where time or computation are at a premium, the cluster-based version is the best and most efficient version of cross-entropy difference. Where the resources are available, the cynical selection method is more accurate.

6. Acknowledgements

We wish to thank Stephan Walter, the Machine Learning Engineering team in Berlin, and Felix Hieber for their support and helpful discussions, as well as the anonymous reviewers for their detailed comments. We also wish to acknowledge the work of the Amazon Saar team to create the software infrastructure we used to train the MT systems.

7. References

[1] A. Axelrod, Y. Vyas, M. Martindale, and M. Carpuat, "Class-Based N-gram Language Difference Models for Data Selection," *IWSLT (International Workshop on Spoken Language Translation)*, 2015.

[2] A. Axelrod, "Cynical Selection of Language Model Training Data," *arXiv [cs.CL]*, pp. 1–19, 2017.

[3] R. C. Moore and W. D. Lewis, "Intelligent Selection of Language Model Training Data," *ACL (Association for Computational Linguistics)*, 2010.

[4] A. Axelrod, X. He, and J. Gao, "Domain Adaptation Via Pseudo In-Domain Data Selection," *EMNLP (Empirical Methods in Natural Language Processing)*, 2011.

[5] M. Kazi, E. Salesky, B. Thompson, J. Taylor, J. Gwinnup, T. Anderson, G. Erdmann, E. Hansen, B. Ore, K. Young, and M. Hutt, "The MITLL-AFRL IWSLT 2016 Systems," *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2016.

[6] N.-Q. Pham, J. Niehues, T.-L. Ha, E. Cho, M. Sperber, and A. Waibel, "The Karlsruhe Institute of Technology Systems for the News Translation Task in WMT 2017," *WMT Conference on Statistical Machine Translation*, 2017.

[7] P. F. Brown, V. J. Della Pietra, P. V. DeSouza, J. C. Lai, and R. L. Mercer, "Class-Based N-gram Models of Natural Language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[8] A. Sethy, P. G. Georgiou, and S. Narayanan, "Text Data Acquisition for Domain-Specific Language Models," *EMNLP (Empirical Methods in Natural Language Processing)*, 2006.

[9] M. Cettolo, C. Girardi, and M. Federico, "WITˆ3 : Web Inventory of Transcribed and Translated Talks," *EAMT (European Association for Machine Translation)*, 2012.

[10] P. Koehn, H. Hoang, A. Birch-Mayne, 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," *ACL (Association for Computational Linguistics) Interactive Poster and Demonstration Sessions*, 2007.

[11] C. Biemann, U. Quasthoff, G. Heyer, and F. Holz, "ASV Toolbox: a Modular Collection of Language Exploration Tools," in *LREC (Language Resources and Evaluation)*, 2008.

[12] J. Ganitkevitch, Y. Cao, J. Weese, M. Post, and C. Callison-Burch, "Joshua 4.0: Packing, PRO, and Paraphrases," in *WMT (Workshop on Statistical Machine Translation)*, 2012.

[13] C. Dyer, V. Chahuneau, and N. A. Smith, "A Simple, Fast, and Effective Reparameterization of IBM Model 2," in *NAACL (North American Association for Computational Linguistics)*, 2013.

[14] 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.

[15] D. Chiang, Y. Marton, and P. Resnik, "Online Large-Margin Training of Syntactic and Structural Translation Features," in *EMNLP (Empirical Methods in Natural Language Processing)*, 2008.

[16] K. Papineni, S. Roukos, T. Ward, and W.-j. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," *ACL (Association for Computational Linguistics)*, 2002.

[17] K. Heafield, "KenLM : Faster and Smaller Language Model Queries," *WMT (Workshop on Statistical Machine Translation)*, 2011.

CHARCUT: Human-Targeted Character-Based MT Evaluation with Loose Differences

Adrien Lardilleux, Yves Lepage***

*Fujitsu Technology Solutions, Luxembourg
*DGT, European Commission, Luxembourg
**IPS, Waseda University, Japan
adrien.lardilleux@ext.ec.europa.eu, yves.lepage@waseda.jp

Abstract

We present CHARCUT, a character-based machine translation evaluation metric derived from a human-targeted segment difference visualisation algorithm. It combines an iterative search for longest common substrings between the candidate and the reference translation with a simple length-based threshold, enabling loose differences that limit noisy character matches. Its main advantage is to produce scores that directly reflect human-readable string differences, making it a useful support tool for the manual analysis of MT output and its display to end users. Experiments on WMT16 metrics task data show that it is on par with the best "untrained" metrics in terms of correlation with human judgement, well above BLEU and TER baselines, on both system and segment tasks.

1. Introduction

A large number of metrics have been proposed in the past years for the task of objective evaluation of Machine Translation. To this day, trained or combined metrics (e.g. BEER [1], DPMFCOMB [2], UOW.REVAL [3], COBALTF [4], among others), generally attain top results in terms of average correlation with human judgement, as was concluded in recent WMT conferences [5, 6].

On the other hand, endogenous metrics present the advantage of versatility, the most widely used remaining BLEU [7] and TER [8]. Such versatility is crucial in environments where MT systems are built and tuned to support numerous languages, some of which having little resources available.

Among those metrics, character-based ones have received more and more interest, starting from BLEU in characters [9, 10] to the recent CHRF [11] and CharacTER [12]. Operating at the character level is all the more important as MT systems working at a sub-word level are getting more widely used, e.g. with segmentation schemes like Byte Pair Encoding in Neural MT [13]. Since character-based metrics can implicitly account for sub-word linguistic phenomena, they have shown to correlate much better with human judgements than BLEU and TER—sometimes by tremendous margins [5, 6]. Nowadays, such metrics seem to be safe for use as drop-in replacements for BLEU.

Another key aspect of MT evaluation is the *display* of MT output and its comparison with human references. Highlighting differences between a candidate and a reference translation is a standard feature of many translation interfaces (e.g. MateCat's edit log [14] or SDL Studio's SDLXLIFF Compare[1]). Basically, any string comparison tool operating at a sub-segment level could be used to that end. The potential associated scores, however, typically simple word or character match percentages, may not reflect all aspects expected from a MT metric. One could just replace it with another metric, but inconsistencies between score and visible differences would ensue [15], which might be confusing for non-specialists. Advanced analysis environments proposing multiple measures along with word-based highlighting, such as Asiya [16] or MT-ComparEval [17], among others, are thus aimed at MT researchers rather than end users.

Some metrics allow to naturally derive user-friendly visual correspondences between candidate and reference translations. This is typically the case of word alignment based metrics (e.g. TER or METEOR [18]), as opposed to those based e.g. on overlapping n-grams (such as BLEU or CHRF), or character-based approaches, which are often subject to noise [12].

We propose an approach that benefits from fine character-based differences while getting rid of their main drawback, namely noise. Initially designed as a mean for displaying differences to end users, it is also a full-fledged MT evaluation metric, as a score can be directly inferred from those human-targeted differences. In this view, a good metric is only the *consequence* of a good visual representation.

This paper is organized as follows: Section 2 describes how user perception of similarities led to the design of our metric; Section 3 builds up on those observations to describe the method in details; Section 4 evaluates it in terms of correlation with human judgements; and Section 5 concludes this work.

[1] http://appstore.sdl.com/app/sdlxliff-compare/89/

2. Defining noticeable similarities

From a translator's point of view, a useful MT output is one that requires little time to comprehend and edit in order to turn it into a quality translation. When the MT contains too many mistakes, it is faster to rewrite a new translation from scratch than to attempt to correct it [15, 19].

A similar process takes place when humans compare two segments. No matter how many common substrings there are, they have no interest if they cannot be identified by the user due to their being lost in a flow of differences. As noted by Wang et al. [12], character-based comparisons typically suffer from noisy matches on languages using alphabets, because letters tend to be repeated frequently. Consider for instance the following segment pair taken from WMT16 evaluations (C denotes the candidate segment, R the reference):

C: It was **also remarkable for** personal reasons.
R: It was **noteworthy because of** personal reasons.

The strings in bold are likely to look like "atomic" replacements for most human eyes. Yet they have characters in common. Indeed, a longest common subsequence between the two strings in bold could be `oerbe␣f` (spaces have the same status as any other character), underlined below:

C: also remarkable for
R: noteworthy because of

Not only would highlighting those prove useless to the user, but it would also direct the eye focus onto meaningless pieces of strings that would go naturally unnoticed without highlighting. It gets even worse when taking *shifts* of isolated characters into account, which is precisely why CharacTER only considers shifts of words. Note that in this particular example, word-based differences would yield much more satisfying results. But those are generally too coarse in the general case, especially when words differ only by e.g. a single ending, or when dealing with morphologically rich languages.

We propose a simple approach to account for subword differences, while showing only meaningful character matches to the user. In order to keep comparisons intelligible, we reduce the number of highlighted substrings (be they matches or differences) within segments by allowing *loose* differences, i.e. differences that may still contain a few common characters. To this end, we rely on standard string difference operations, with the addition of a single constraint: only substrings longer than a given threshold are considered for matching. In our experiments (Sec. 4), we have found that the best value is generally around 3 characters for European languages, and manual investigations suggest that an optimum would be 1 or 2 characters for Chinese, which is in line with the findings of Li et al. [10]. This single constraint significantly reduces the amount of displayed information, helping the user focus more on meaningful differences.

To our knowledge, this approach was first used as a similarity measure by [20] in a clinical context for patient record matching. More recently, it was successfully applied to toponym matching [21]. It also presents similarities with the more complex MUMmer, a genome alignment system first introduced in [22], where what we call loose differences are the counterpart of what is known in bioinformatics as "highly polymorphic regions," i.e. short regions of DNA that have undergone many mutations.

3. Method description

CHARCUT consists of three phases:

1. an iterative search for longest common substrings between the candidate and the reference translations;
2. the identification of string shifts;
3. a scoring phase based on the lengths of remaining differences.

3.1. Iterative segmentation algorithm

In the first phase, we identify a set of non-overlapping matches by applying an iterative search for the longest common substring (hereafter LCSubstr[2]) between a candidate C_0 and a reference R_0, and cutting off this LCSubstr from both segments:

$$\begin{aligned} C_{n+1} &= C_n - \text{LCSubstr}(C_n, R_n) \\ R_{n+1} &= R_n - \text{LCSubstr}(C_n, R_n) \end{aligned} \quad (1)$$

When several LCSubstr's are possible (same length), the leftmost one in C_n is processed first, and is paired with the leftmost corresponding match in R_n. A LCSubstr removed is replaced with a (zero-length) hard boundary that subsequent LCSubstr's cannot cross. We iterate until the length of LCSubstr(C_n, R_n), which monotonously decreases at each step, is below a certain threshold (typically around 3 characters).

Our first investigations have revealed that pure character-based matching, treating spaces as any other character, could lead to misinformed segmentations in presence of shifts of words with identical prefixes or suffixes (see Fig. 1 for an example). For this reason, we consider only a subset of all possible substrings of C_0 and R_0 when searching for the LCSubstr, by considering only those that match any of the three following regular expressions:

- `\W*\w+\W*` (intra-word substring, does not span multiple words);
- `\W*\b.+\b\W*` (inter-word substring, stops at word boundaries or non-word characters);
- `\W+` (run of non-word characters).

[2] Contrary to the Longest Common Subsequence (LCS), the LCSubstr is exclusively made up of adjacent characters.

C: [...] der Europäischen Gemeinsamen Strategie zur Unterstützung Palästinas [...]

R: [...] der Gemeinsamen Europäischen Strategie zur Unterstützung Palästinas [...]

Figure 1: *A common pitfall where the raw character based longest-first approach can lead to a counter-intuitive segmentation. The first LCSubstr is underlined. Because the two swapped German words* Europäischen *and* Gemeinsamen *share the same ending, this ending has been integrated into the LCSubstr, preventing the more natural full word matches. We circumvent this issue by making our algorithm aware of word separators.*

n	C_n	R_n	LCSubstr(C_n, R_n)	length
0	Before the game, it had arrived at the stadium to riots.	Before the match there was a riot in the stadium.	␣the␣stadium	12
1	Before the game, it had arrived at| to riots.	Before the match there was a riot in|.	Before␣the␣	11
2	|game, it had arrived at| to riots.	|match there was a riot in|.	␣riot	5
3	|game, it had arrived at| to|s.	|match there was a| in|.	at	2

Figure 2: *Example of iterative search for longest common substrings (LCSubstr). At each step, the LCSubstr (underlined) is cut off and replaced with a zero-length boundary (noted with a pipe character "\|") that subsequent LCSubstr's may not cross. The process stops when the length of the LCSubstr is below a given threshold—here, 3 characters, preventing smaller common substrings, starting with* at *at step 3, to be considered as matches. The longest common suffix (single full stop) is eventually added to the list of LCSubstr's, while the longest common prefix was already extracted as a regular LCSubstr.*

C_0: Before the game, it had arrived at the stadium to riots.

R_0: Before the match there was a riot in the stadium.

Figure 3: *Segmentation resulting from the iterative search of Fig. 2. Matches (= LCSubstr's) are underlined, and the remaining substrings are loose differences. Here, those differences still have around 68% of characters in common (16), while no meaningful lexical correspondences are visible: the length-based threshold has successfully prevented a large amount of noise that would otherwise make the output unreadable.*

This leads to a mix of word- and character-based LCSubstr's which we felt more natural than pure character-based ones in our experiments. In the case of scripts without word separators such as Chinese, most LCsubstr's match the first expression.

Eventually, we also add the longest common prefix and the longest common suffix between C_0 and R_0 to the list of LCSubstr's, independently of their length, providing they match the second or third regular expression and were not already extracted as a regular LCSubstr. This addition had almost no impact in terms of correlation with human judgement in our experiments, but it improves highlighting by fixing frequent cases of true negatives, such as final punctuations or segments shorter than the minimum match size, that most users would expect to be considered as matches.

Then:

- the set of LCSubstr's extracted up to this point (including longest common prefix and suffix) are matches;
- the remaining strings, i.e. the last computed C_n and R_n, are loose differences.

Figures 2 and 3 give an example. Contrary to edit distances, our approach does not yield a minimal sequence of operations that would turn C_0 into R_0; instead, it seeks to lower the *number* of matches and differences, hence the user reading effort.

3.2. Identifying string shifts

CHARCUT naturally handles string shifts, as the position change between ␣the␣stadium and ␣riot in Fig. 3 illustrates. For the purpose of highlighting and scoring, we mark the shortest one (␣riot) as a shift, and the other one as a regular match.

More generally, when faced with multiple alternative shifts, we identify the longest common subsequence, in total number of characters, between the sequence of LCSubstr's from C_0 (hereafter noted C_{match}) and that from R_0 (hereafter R_{match}), and any LCSubstr left out is marked as a shift. The two input sequences have exactly the same tokens, but in a different order (here 4 tokens = 4 LCSubstr's, delimited by a pipe "|"):

C_{match} = Before␣the␣|␣the␣stadium|␣riot|.
R_{match} = Before␣the␣|␣riot|␣the␣stadium|.

The longest common subsequence has three tokens: Before␣the␣|␣the␣stadium|. for a total of 12+11+1=24 characters. Those tokens will be referred to as *regular matches* in the following. Tokens left out (here, the single token ␣riot) are marked as *shifts*, and will be scored and highlighted accordingly later on. Note that our definition

of the longest common subsequence deviates from the general LCS definition, since we do not base its computation on the number of tokens, but on the sum of their lengths.

3.3. Scoring scheme

The result of the previous phases is a segmentation of the input segments in three types of substrings: regular matches, shifts, and loose differences. Loose differences include deletions (from the candidate segment) and insertions (into the reference segment). We derive a score from those substrings by assigning a cost to each character of the candidate and reference segments:

- characters from regular matches have no cost;
- shifted, deleted, and inserted characters have a cost of 1 (shifted characters are counted only once although they appear in both segments).

We do not consider the combination *deletion + insertion = replacement* as a single operation because, by definition, there is no correspondence inside loose differences. While this combination appears natural when dealing with word units, it makes much less sense on characters, as identification of replacement pairs within differences would be arbitrary, especially if their lengths highly differ between the candidate and the reference.

While CharacTER assigns a shift cost equal to the average word length of the shifted phrase, we use the total number of shifted characters instead, thus keeping the computation rather straightforward. There is little risk of over-evaluating the cost of shifts because, by definition (Sec. 3.2), their length is minimal.

In our setting, the cost of post-edition is thus the total number of edited characters. An intuitive normalization scheme would be to divide this number by the total length of the candidate and reference segments in order to produce a score between 0 and 1:

$$\text{score}_{\text{orig}} = \frac{\#\text{deletions} + \#\text{insertions} + \#\text{shifts}}{|C_0| + |R_0|} \quad (2)$$

However, following Wang et al. [12], we tried using only the length of the candidate, and we could confirm that it generally leads to higher correlation with human judgements (see experiments in next section). We thus consider also the following variant, where we divide by twice the candidate length instead, and limit the final score to 1 in case the number of edited characters exceeds the denominator:

$$\text{score}_{C} = \min\left(1, \frac{\#\text{deletions} + \#\text{insertions} + \#\text{shifts}}{2 \times |C_0|}\right) \quad (3)$$

The lower the scores, the better. A score of zero means that the candidate and reference segments are identical. In the example of Fig. 3, we obtain $\text{score}_{\text{orig}} = \frac{27+21+5}{56+49} \simeq 0.50$ and $\text{score}_{C} = \frac{27+21+5}{2\times 56} \simeq 0.47$.

4. Experiments

4.1. Task description

We evaluate CHARCUT on the WMT16 system- and segment-level news metrics tasks [6], using the official evaluation scripts. We report results obtained with the "direct assessment" golden truth (hereafter DA), as it was concluded in WMT16 that it was more reliable than relative ranking, and it was also chosen as the official human evaluation at WMT17. Under this evaluation scheme, humans evaluate the adequacy of translations on an absolute scale in isolation from other translations, and the correlation with automatic scores is measured by means of absolute Pearson correlation coefficient. The data consist of news texts from Czech, Finnish, German, Romanian, and Turkish, into English, plus the Russian-English language pair in both directions.[3]

In addition, we report results of the segment-level tasks under the "HUMEseg" evaluation scheme [23], which was also an official human evaluation of WMT17. Similarly to DA, scores are compared using the Pearson correlation coefficient, but with human judgements of semantic nodes aggregated over each sentence rather than single absolute scores. The data used for those tracks are texts from the medical domain from English into Czech, German, Romanian, and Polish.

4.2. Optimizing for correlation with human judgement

Figure 4 reports the average Pearson correlations between human judgements and various set-ups of CHARCUT. On average, the correlations obtained with the score_C scheme (eq. 3) is greater than that obtained with $\text{score}_{\text{orig}}$ (eq. 2) by 0.01, which confirms the findings of Wang et al. [12].

Although in practice varying the minimum match size leads to visually very different outputs, especially with low values, they seem to have a limited impact on correlations with human judgements: the average range of the absolute Pearson coefficient (difference between maximum and minimum) is 0.01. The system- and segment-level DA graphs show curves that tend to increase slightly then decrease, with a maximum correlation when the minimum match size equals 2 or 3 characters. This is consistent with the sense we get from the corresponding highlighting, which "looks right" to the eye—too small values leading to noisy matches, and too high values to silence.

On the contrary, the monotonously decreasing curves of the segment-level HUME graph suggest that smaller minimum match sizes would be better, which is in contradiction with the other results. We will nevertheless restrict our following experiments to a minimum match size of 3 characters, as it constitutes a good compromise between the above three

[3] The DA evaluations of WMT17 cover more diverse target languages, in particular Chinese, which constitutes a good test for character based approaches, but the official evaluation scripts were not publicly released at the time this paper was written. For consistency, we therefore chose to stick to the WMT16 evaluations.

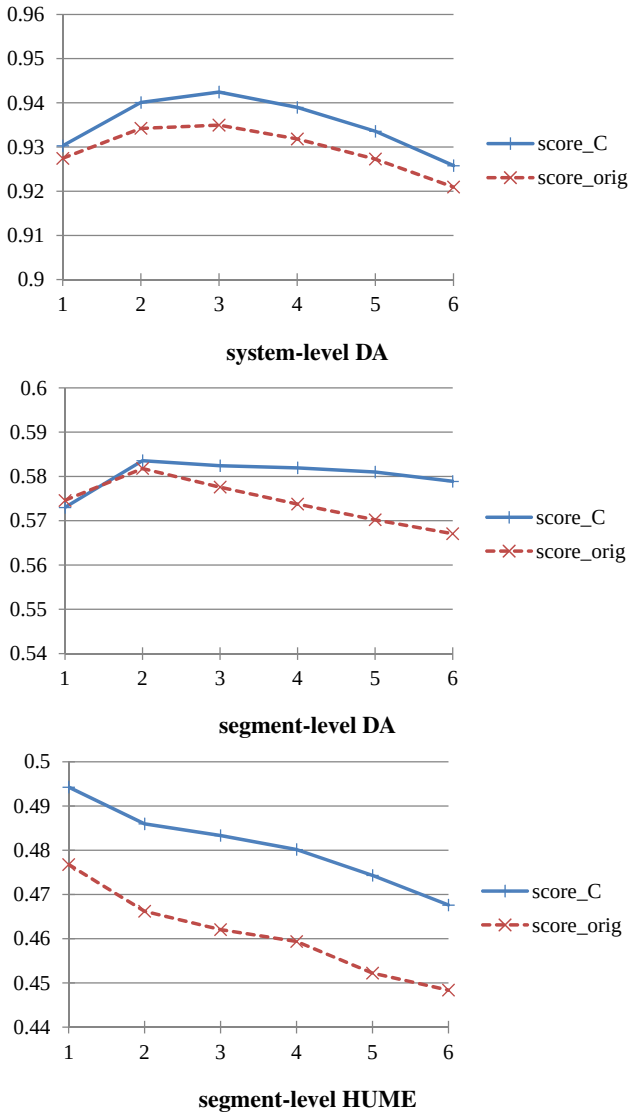


Figure 4: *Average correlations between* CHARCUT *scores and human judgements on WMT16's metrics tasks. In ordinates, the absolute Pearson correlation coefficient. In abscissae, the minimum match size in characters (length-based threshold). The reported numbers are averages over all language pairs. Normalizing with the candidate segment length only ($score_C$) consistently outperforms using both the candidate and reference lengths ($score_{orig}$).*

evaluations and our manual investigations, in which we observed that a 2 character threshold still produced too many noisy matches.

4.3. Comparison with other metrics

Table 1 compares CHARCUT with metrics that took part in the WMT16 evaluations. For conciseness, we only report average correlations over all language pairs. The rankings differ from the official WMT16 results since we chose to focus on the Direct Assessment and HUME evaluations, while the official evaluations were based on Relative Ranking. The reported results make therefore no pretence to (re-)define the "best metrics;" rather, they are only meant to show that the scores produced by CHARCUT, which are first and foremost intended to be presented to users along with segment highlighting, are globally as good as other recent metrics, well above well-known baselines.

In these experiments, CHARCUT uses the $score_C$ normalization scheme and a minimum match size of 3-characters. We also report additional correlations obtained with the Levenshtein distance, normalized with the sum of the source and target segment lengths, to serve as a character-based baseline; as well as with TER and CharacTER on segment-level tasks.

Globally, CHARCUT's results are very close to those of MPEDA [24], which relies yet on additional training corpora. Compared with other endogenous metrics (chrF, wordF, CharacTER, variants of BLEU and TER, Levenshtein distance), CHARCUT produces top average correlations on the system- and segment-level DA evaluations, and is only superseded by chrF on the HUME evaluation. A fortiori, its correlations are much higher than those of the BLEU and TER baselines: from +9% relative Pearson correlation (MTEVALBLEU, system-level DA) up to +23% (TER, segment-level HUME).

Unexpectedly, the simple normalized character-based Levenshtein distance performs quite well, outperforming even metrics like BEER and CharacTER on the DA evaluations. CHARCUT nevertheless represents a consistent improvement over it, by +0.03 absolute Pearson correlation on average.

4.4. Processing time

We used a random sample of 10,000 segment pairs from WMT16 to measure the speed of CHARCUT. The average reference length in this sample was 113 characters. On a 2.8 GHz processor, our Python implementation could process 260 segment pairs per second, using a minimum match size of 3 characters, which is faster than required in most situations. For comparison, CharacTER and CHRF, also Python implementations, could process respectively 54 and 600 segment pairs per second with default settings on the same machine.

5. Conclusion

We have presented CHARCUT, a character-based machine translation evaluation metric. It relies on *loose differences*, residuals from an iterative search for longest common substrings. Initially designed for displaying differences between reference and candidate segments to end users, it also produces scores that should look consistent to most, since they directly reflect those differences. In this view, good correlation with human judgement is only a consequence of a good

Table 1: *Comparison of* CHARCUT*'s performances with metrics that took part in the system-level DA, segment-level DA, and segment-level HUME tasks of WMT16. We report the average Pearson correlation coefficients over all language pairs. Averages within brackets refer to metrics that did not participate in the English-to-Russian evaluation, so they are based on one less figure. Asterisks indicate our own runs; all other averages are based on figures from [6].* CHARCUT *is globally on par with the best metrics in those evaluations.*

system-level DA

Metric	Avg. corr. ± stddev.
UOW.REVAL	(0.972 ± 0.013)
MPEDA	0.945 ± 0.044
*CHARCUT	0.942 ± 0.037
CHRF2	0.934 ± 0.038
CHRF3	0.934 ± 0.035
*Lev. distance	0.930 ± 0.049
BEER	0.928 ± 0.054
CHRF1	0.927 ± 0.051
CHARACTER	0.922 ± 0.055
MTEVALNIST	0.886 ± 0.068
MTEVALBLEU	0.867 ± 0.060
MOSESCDER	0.861 ± 0.061
MOSESTER	0.851 ± 0.061
MOSESPER	0.842 ± 0.096
WORDF3	0.836 ± 0.069
WORDF2	0.836 ± 0.069
WORDF1	0.831 ± 0.071
MOSESWER	0.812 ± 0.099
MOSESBLEU	0.810 ± 0.082

segment-level DA

Metric	Avg. corr. ± stddev.
DPMFCOMB	(0.633 ± 0.048)
METRICS-F	(0.631 ± 0.049)
COBALT-F.	(0.617 ± 0.040)
MPEDA	0.584 ± 0.053
*CHARCUT	0.582 ± 0.076
UPF-COBALT	(0.582 ± 0.060)
CHRF3	0.560 ± 0.082
CHRF2	0.559 ± 0.081
*Lev. distance	0.556 ± 0.065
BEER	0.556 ± 0.082
CHRF1	0.548 ± 0.079
*CHARACTER	0.537 ± 0.074
UOW.REVAL	0.530 ± 0.035
WORDF3	0.524 ± 0.055
WORDF2	0.522 ± 0.055
WORDF1	0.514 ± 0.055
SENTBLEU	0.510 ± 0.039
*TER	0.485 ± 0.052
DTED	0.330 ± 0.058

segment-level HUME

Metric	Avg. corr. ± stddev.
CHRF3	0.519 ± 0.096
CHRF2	0.517 ± 0.092
BEER	0.513 ± 0.079
CHRF1	0.503 ± 0.079
MPEDA	0.492 ± 0.073
*CHARCUT	0.483 ± 0.121
WORDF3	0.452 ± 0.092
WORDF2	0.450 ± 0.091
WORDF1	0.439 ± 0.088
*CHARACTER	0.438 ± 0.126
*Lev. distance	0.437 ± 0.109
SENTBLEU	0.401 ± 0.101
*TER	0.394 ± 0.125

visual representation. Experiments on WMT16 metrics tasks have thus shown that those scores are well correlated with human judgements, globally on par with other recent metrics like CHRF and MPEDA, ahead of BLEU and TER baselines by up to 23% relative Pearson correlation in our experiments. It is also language independent and requires no additional resource or training. Possible improvements include better handling of shifts, as CHARCUT is currently unaware of shift distance; or again automatically correlate the minimum match size with the number of highlighted substrings in order to keep outputs readable even with very different input segments.

6. Availability

CHARCUT is open source and available at `https://github.com/alardill/CharCut`. It consists of a single Python script that computes scores and highlights differences (HTML outputs). Figure 5 shows a sample output.

7. References

[1] M. Stanojević and K. Sima'an, "BEER 1.1: ILLC UvA submission to metrics and tuning task," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 396–401. [Online]. Available: http://aclweb.org/anthology/W15-3050

[2] H. Yu, Q. Ma, X. Wu, and Q. Liu, "CASICT-DCU Participation in WMT2015 Metrics Task," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 417–421. [Online]. Available: http://aclweb.org/anthology/W15-3053

[3] R. Gupta, C. Orasan, and J. van Genabith, "Machine Translation Evaluation using Recurrent Neural Networks," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 380–384. [Online]. Available: http://aclweb.org/anthology/W15-3047

[4] M. Fomicheva, N. Bel, L. Specia, I. da Cunha, and A. Malinovskiy, "CobaltF: A Fluent Metric for MT Evaluation," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 483–490. [Online]. Available: http://www.aclweb.org/anthology/W/W16/W16-2339

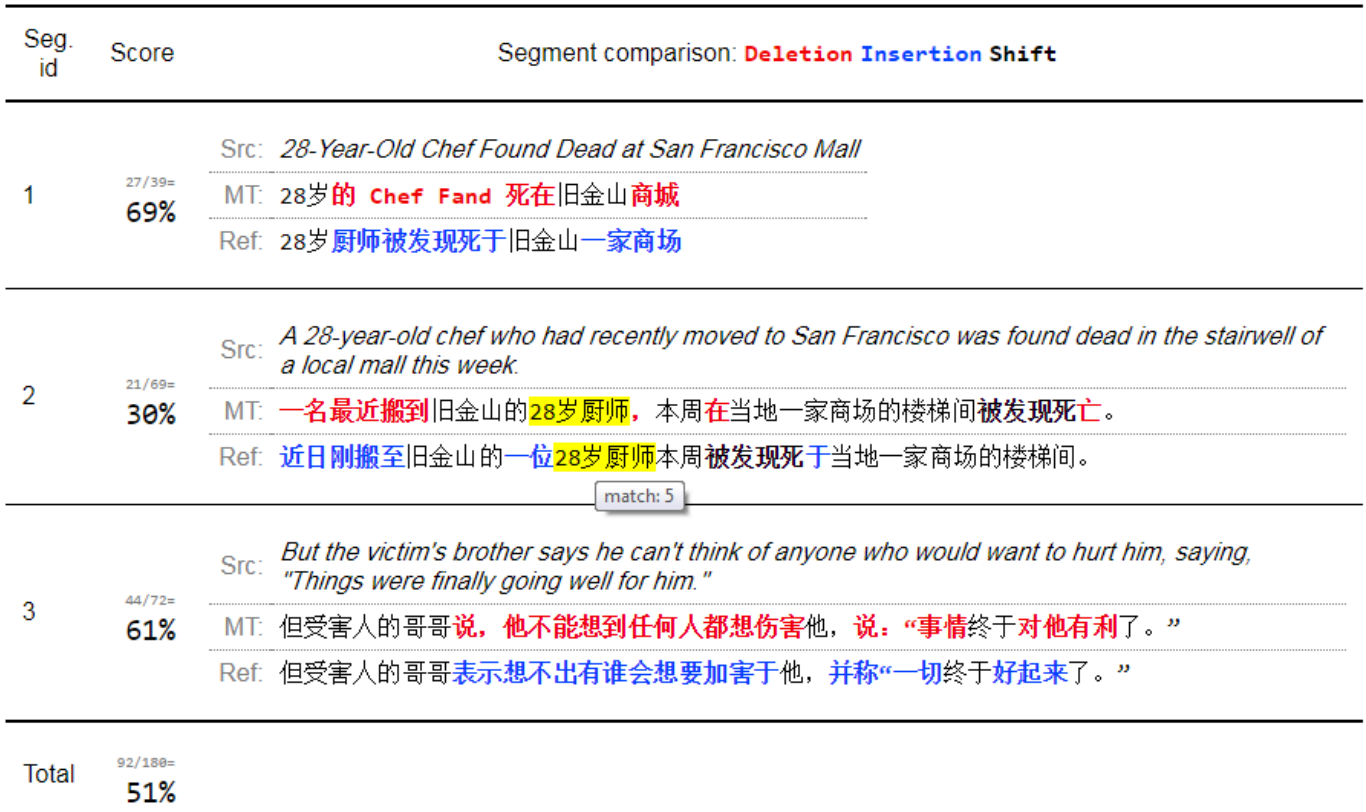


Figure 5: *Sample HTML output obtained on the first three English-Chinese segments of WMT17, using a 2-character minimum match size. The interface is kept slick on purpose and uses only classical colours: red for deletions, blue for insertions, plus bold for shifts (here, only one in second segment). The scores reflect directly the number of highlighted characters. The background of matching substrings turns yellow when pointed with the mouse.*

[5] M. Stanojević, A. Kamran, P. Koehn, and O. Bojar, "Results of the WMT15 Metrics Shared Task," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 256–273. [Online]. Available: http://aclweb.org/anthology/W15-3031

[6] O. Bojar, Y. Graham, A. Kamran, and M. Stanojević, "Results of the WMT16 Metrics Shared Task," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 199–231. [Online]. Available: http://www.aclweb.org/anthology/W/W16/W16-2302

[7] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," in *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. [Online]. Available: http://www.aclweb.org/anthology/P02-1040

[8] 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*. Cambridge, Massachusetts, USA: Association for Computational Linguistics, August 2006, pp. 223–231. [Online]. Available: http://www.mt-archive.info/AMTA-2006-Snover.pdf

[9] Étienne Denoual and Y. Lepage, "BLEU in Characters: Towards Automatic MT Evaluation in Languages without Word Delimiters," in *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP)*, Jeju Island, Republic of Korea, October 2005, pp. 79–84. [Online]. Available: http://www.aclweb.org/anthology/I/I05/I05-2014.pdf

[10] M. Li, C. Zong, and H. T. Ng, "Automatic Evaluation of Chinese Translation Output: Word-Level or Character-Level?" in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 159–164. [Online]. Available: http://www.aclweb.org/anthology/P11-2028

[11] M. Popović, "chrF: character n-gram F-score for automatic MT evaluation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 392–395. [Online]. Available: http://aclweb.org/anthology/W15-3049

[12] W. Wang, J.-T. Peter, H. Rosendahl, and H. Ney, "CharacTer: Translation Edit Rate on Character Level," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 505–510. [Online]. Available: http://www.aclweb.org/anthology/W/W16/W16-2342

[13] R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with Subword Units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 1715–1725. [Online]. Available: http://www.aclweb.org/anthology/P16-1162

[14] M. Federico, N. Bertoldi, M. Cettolo, M. Negri, M. Turchi, M. Trombetti, A. Cattelan, A. Farina, D. Lupinetti, A. Martines, A. Massidda, H. Schwenk, L. Barrault, F. Blain, P. Koehn, C. Buck, and U. Germann, "THE MATECAT TOOL," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, August 2014, pp. 129–132. [Online]. Available: http://www.aclweb.org/anthology/C14-2028

[15] S. O'Brien, "Towards predicting post-editing productivity," *Machine Translation*, vol. 25, no. 3, p. 197, 2011. [Online]. Available: http://dx.doi.org/10.1007/s10590-011-9096-7

[16] J. Giménez and L. Màrquez, "Asiya: An Open Toolkit for Automatic Machine Translation (Meta-)Evaluation," *The Prague Bulletin of Mathematical Linguistics*, no. 94, pp. 77–86, 2010. [Online]. Available: http://ufal.mff.cuni.cz/pbml/94/art-gimenez-marques-evaluation.pdf

[17] O. Klejch, E. Avramidis, A. Burchardt, and M. Popel, "MT-ComparEval: Graphical evaluation interface for Machine Translation development," *The Prague Bulletin of Mathematical Linguistics*, no. 104, pp. 63–74, 2015. [Online]. Available: http://ufal.mff.cuni.cz/pbml/104/art-klejch-et-al.pdf

[18] 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*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 228–231. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0734

[19] M. Turchi, M. Negri, and M. Federico, "MT Quality Estimation for Computer-assisted Translation: Does it Really Help?" in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 530–535. [Online]. Available: http://www.aclweb.org/anthology/P15-2087

[20] C. Friedman and R. Sideli, "Tolerating spelling errors during patient validation," *Computers and Biomedical Research*, vol. 25, no. 5, pp. 486–509, 1992.

[21] G. Recchia and M. Louwerse, "A Comparison of String Similarity Measures for Toponym Matching," in *Proceedings of The First ACM SIGSPATIAL International Workshop on Computational Models of Place*, Orlando, USA, November 2013, pp. 54–61. [Online]. Available: http://doi.acm.org/10.1145/2534848.2534850

[22] A. L. Delcher, S. Kasif, R. D. Fleischmann, J. Peterson, O. White, and S. L. Salzberg, "Alignment of whole genomes," *Nucleic Acids Research*, vol. 27, no. 11, pp. 2369–2376, 1999. [Online]. Available: http://dx.doi.org/10.1093/nar/27.11.2369

[23] A. Birch, O. Abend, O. Bojar, and B. Haddow, "HUME: Human UCCA-Based Evaluation of Machine Translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 1264–1274. [Online]. Available: https://aclweb.org/anthology/D16-1134

[24] L. Zhang, Z. Weng, W. Xiao, J. Wan, Z. Chen, Y. Tan, M. Li, and M. Wang, "Extract Domain-specific Paraphrase from Monolingual Corpus for Automatic Evaluation of Machine Translation," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 511–517. [Online]. Available: http://www.aclweb.org/anthology/W/W16/W16-2343

AUTHOR INDEX

iwslt.org