15th International Workshop
on Spoken Language Translation

Bruges, Oct. 29 - 30, 2018

Proceedings

www.iwslt.org

Proceedings of the

International Workshop on Spoken Language Translation

October 29-30, 2018

Bruges, Belgium

Edited by

Marco Turchi

Jan Niehues

Marcello Frederico

Contents

Foreword

The International Workshop on Spoken Language Translation (IWSLT) is an annually scientific workshop, associated with an open evaluation campaign on Spoken Language Translation, where both scientific papers and system descriptions are presented.

Since 2004, the annual workshop has been held six times in Asia, five times in America, and three times in Europe. This year, the 15th International Workshop on Spoken Language Translation is hosted Europe, in awesome Bruges, Belgium from 29th to 30th October 2018. The reason was to co-locate IWSLT with other two important and related conferences: the Conference on Machine Translation (WMT) and the Conference on Empirical Methods for Natural Language Processing (EMNLP), both taking place in Brussels, after IWSLT.

The IWSLT workshop includes presentations of scientific papers in dedicated technical sessions, either in oral or poster form. Scientific papers cover theoretical and practical issues in the field of machine translation, spoken language translation, automatic speech recognition, text-to-speech synthesis. This year, we received 26 submissions of scientific papers, which were carefully peer-reviewed by members of the program committee. Out of them, 12 were selected for publication based on their technical merit and relevance to the workshop. The proceedings of IWSLT are published on the workshop website.

As in the past editions, IWSLT will assign a best student paper award for which this year we have selected two finalists that will give an oral presentation of their work.

The workshop will also include the presentation and discussion of the outcomes of the 2018 IWSLT evaluation campaign, which this year focused on two tasks: low resource machine translation of TED Talks from Basque to English, and speech translation of lectures from English to German.

The low resource translation task addressed a conventional bilingual text translation task, but given the difficulty of the proposed translation direction and the scarcity of available parallel data, we supported the development of deep learning models also leveraging parallel data from related languages.

The speech translation task also introduced several novel aspects with respect to the past, which pushed participants to go beyond the usual pipeline approach of cascading automatic speech recognition and machine translation. First, this year we only evaluated end-to-end performance, so that every participant had to generate German translations based on the English audio. Second, for participants who wanted to focus on one component of the pipeline, we provided baseline components for the other parts. Third, we introduced a special evaluation condition to test end-to-end models, namely deep learning models directly mapping source language speech into target language text. For this condition, we provided an aligned TED corpus of English audio and German texts.

For each proposed task, monolingual and bilingual language resources, as needed,

were provided to participants in order to train their systems. Another novelty we introduced this year was to extend the evaluation period from one week to one month, so that participants could better plan their effort. Blind test sets were released at the begin of July, and all translation outputs produced by the participants were evaluated using several automatic translation quality metrics. Finally, each participant was requested to submit a paper describing his system and the utilized resources. System papers went through a peer-review process only aiming at improving their overall quality.

The efforts of the organizers were definitely rewarded. The IWSLT evaluation campaign attracted this year 15 research teams, 8 taking part to the low-resource translation task, and 9 to the speech translation task. The presentations of the corresponding 15 system description papers will be as usual preceded by an overview talk of the evaluation campaign given by one of the organizers.

Before opening the workshop a few words of acknowledgment.
I would like to express my gratitude to the evaluation committee, Jan Niehues, Mauro Cettolo, Sebastian Stüker, Luisa Bentivogli, and Roldano Cattoni for preparing such a great playground for the research community. To the program chair Marco Turchi and all the program committee members for arranging an excellent workshop program. To Margit Rödder, for taking care of the website and all the local and financial arrangements.

Finally, I would like to thank our generous sponsors for supporting IWSLT. Our gold sponsors *Amazon Web Services* and *AppTek*, and, our silver sponsor *M*Modal*. This year sponsors will have the opportunity to give a short corporate presentation at our workshop, which I'm sure will enrich IWSLT with an industry perspective on spoken language translation and provide additional networking opportunities in our community.
I wish all the participants a fruitful and engaging IWSLT workshop!

Welcome to Bruges!

Marcello Federico
Workshop Chair IWSLT 2018

Organizers

Chairs

Marcello Federico (FBK, Italy /Amazon, USA): Workshop
Marco Turchi (FBK, Italy) : Program
Jan Niehues (KIT,Germany): Evaluation
Alex Waibel (CMU, USA/KIT, Germany): Steering Committee

Local and Financial Chair

Margit Röder (KIT, Germany)

Evaluation Technical Committee

Jan Niehues (KIT, Germany)
Mauro Cettolo (FBK, Italy)
Sebastian Stüker (KIT, Germany)
Luisa Bentivogli (FBK, Italy
Roldano Cattoni (FBK, Italy)
Marcello Federico (FBK, Italy/Amazon, USA)

Steering Committee

Marcello Federico, (FBK, Italy/Amazon, USA)
Masakiyo Fujimoto (NICT, Japan)
Will Lewis (Microsoft Research, USA)
Chi Mai Luong (IOIT, Vietnam)
Joseph Mariani (IMMI, France)
Satoshi Nakamura (NAIST, Japan)
Hermann Ney (RWTH, Germany)
Sebastian Stüker (KIT, Germany)
Alex Waibel (CMU, USA/KIT, Germany)
Francois Yvon (CNRS-LIMSI, France)

Program Committee

Marco Turchi, FBK (Italy), Program Chair
Duygu Ataman, FBK (Italy)
Loic Barrault, Université du Mans (France)
Laurent Besacier, LIG (France)

Francisco Casacuberta, UPV (Spain)
José Guilherme Carmago de Souza, eBay Inc. (USA)
Mauro Cettolo, Pervoice (Italy)
Boxing Chen, NRC-CNRC (Canada)
Mattia Di Gangi, FBK (Italy)
Orhan Firat, Google (USA)
Mark Fishel, University of Tartu (Estonia)
George Foster, Google (Canada)
Paco Guzman, Facebook (USA)
Barry Haddow, University of Edinburgh (UK)
Christian Hardmeier, University of Uppsala (UK)
Mathias Huck, LMU (Germany)
Marcin Junczys-Dowmunt, Microsoft (USA)
Kevin Kilgour, Google (Zürich)
Philip Kohen, John Hopkins University (USA)
Julia Kreutzer, Heildeberg University (Germany)
Roland Kuhn, NRC (Canada)
Yves Lepage, Waseda University (Japan)
Will Lewis, Microsoft (USA)
Qun Liu, DCU (Ireland)
Evgeny Matusov, Apptek (Germany/USA)
Surafel Melaku Lakew, FBK (Italy)
Markus Müller, KIT(Germany)
Graham Neubig, CMU (USA)
Michael Paul, ATR-Trek (Japan)
Stephan Peitz, Apple (USA)
Maja Popović, DCU (Ireland)
Matt Post, Amazon Research
Elizabeth Salesky, CMU (USA)
Carolina Scarton, University of Sheffield (UK)
Lane Schwartz, U. Illinois (USA)
Rico Senrich, University of Edinburgh (UK)
Matthias Sperber, KIT(Germany)
Felix Stahlberg, University of Cambridge (UK)
Ales Tamchyna, Charles University (Czech Republic)
Jörg Tiedemann, H. Helsinki (Finland)
Antonio Toral, University Groningen (The Netherlands)
Vincent Vandeghinste, KU Leuven (Belgium)
David Vilar, Amazon Research (Germany)
Andy Way, DCU (Ireland)
Marion Weller, Stuttgart University (Germany)
Philip Williams, University of Edinburgh (UK)

Acknowledgments

IWSLT 2018 is proud to present its gold sponsor

and its silver sponsor

Participants

	SLT		Low-resourced MT
	Baseline	End-to-End	
USTC-NEL (CHINA)	✔	✔	
ADAPT (IRELAND)			✔
University of Helsinki (FINLAND)			✔
MeMAD (FINLAND)	✔		
Prompsit (SPAIN)			✔
AppTek (GERMANY)	✔		
Sogou-TIIC (CHINA)	✔		
SRPOL-UEDIN (UK/POLAND)			✔
AFRL (USA)	✔		✔
KIT (GERMANY)	✔	✔	
Alibaba (CHINA)	✔		
CUNI (CZECH)			✔
FBK (ITALY)		✔	✔
JHU/KyotoU (USA/JAPAN)		✔	
SGNLP (KOREA)		✔	

Program

Monday, October 29th, 2018

08:30-09:00	**Workshop Registration**
09:00-09:15	**Welcome Remarks** *Marcello Federico (Workshop Chair)* Amazon-FBK, USA, Italy
09:15-10:05	**Invited talk: Multimodal Machine Translation** *Prof. Lucia Specia* University of Sheffield, UK
10:05-11:05	**INDUSTRIAL SESSION**
10:05-10:20	**Gold sponsor presentation: AppTek**
Coffee Break (10:20-10:50)	
10:50-11:05	**Silver sponsor presentation: M*Modal**
11:00-12:00	**EVALUATION CAMPAIGN**
11:05-12:00	**Report on the 15th IWSLT Evaluation Campaign, IWSLT 2018** *Jan Niehues, Roldano Cattoni, Sebastian Stüker, Mauro Cettolo, Marco Turchi, Marcello Federico* KIT & FBK, Germany, Italy
Lunch (12:00-13:30)	
13:30-15:00	**ORAL SESSION I**
13:30-14:00	**Samsung and University of Edinburgh's System for the IWSLT 2018 Low Resource MT Task** *Philip Williams, Marcin Chochowski, Pawel Przybysz, Rico Sennrich, Barry Haddow and Alexandra Birch* UEDIN & Samsung R&D Institute, UK, Poland
14:00-14:30	**The USTC-NEL Speech Translation system at IWSLT 2018** *Dan Liu, Junhua Liu, Wu Guo, Shifu Xiong, Zhiqiang Ma, Rui Song, Chongliang Wu and Quan Liu* USTC & IFLYTEK Co. LTD, China

14:30-15:00	**Unsupervised Parallel Sentence Extraction from Comparable Corpora** *Viktor Hangya, Fabienne Braune, Yuliya Kalasouskaya and Alexander Fraser* LMU & Volkswagen Data Lab Munich, Germany
Coffee Break (15:00-15:30)	
15:30-17:30	**POSTER SESSION**
	The JHU/KyotoU Speech Translation System for IWSLT 2018 *Hirofumi Inaguma, Xuan Zhang, Zhiqi Wang, Adithya Renduchintala, Shinji Watanabe and Kevin Duh* JHU & KyotoU, USA, Japan
	The ADAPT System Description for the IWSLT 2018 Basque to English Translation Task *Alberto Poncelas, Kepa Sarasola and Andy Way* DCU, Ireland
	The University of Helsinki submissions to the IWSLT 2018 low-resource translation task *Yves Scherrer* HelsinkiU, Finland
	Prompsit's Submission to IWSLT 2018 Low Resource Machine Translation Task *Víctor M. Sánchez-Cartagena* Prompsit, Spain
	The AFRL IWSLT 2018 Systems: What Worked, What Didn't *Brian Ore, Eric Hansen, Katherine Young, Grant Erdmann and Jeremy Gwinnup* AFRL, USA
	Alibaba Speech Translation Systems for IWSLT 2018 *Fei Huang, Nguyen Bach and Chongjia Ni* Alibaba, China
	The MeMAD Submission to the IWSLT 2018 Speech Translation Task *Umut Sulubacak, Aku Rouhe, Jörg Tiedemann, Stig-Arne Grönroos and Mikko Kurimo* HelsinkiU & AaltoU, Finland
	Neural Speech Translation at AppTek *Evgeny Matusov, Patrick Wilken, Parnia Bahar, Julian Schamper, Pavel Golik, Albert Zeyer, Joan Albert Silvestre-Cerda, Adria Martinez-Villaronga, Hendrik Pesch and Jan-Thorsten Peter* AppTek, Germany
	KIT's IWSLT 2018 SLT Translation System *Matthias Sperber, Ngoc Quan Pham, Thai Son Nguyen, Jan Niehues, Markus Müller, Thanh-Le Ha, Sebastian Stüker and Alex Waibel* KIT, Germany
	CUNI Basque to English Submission in IWSLT18 *Tom Kocmi, Dušan Variš and Ondřej Bojar* CUNI, Czech Republic
	Adapting Multilingual NMT to Extremely Low Resource Languages, FBK's Participation of the Basque-English Low Resource MT, IWSLT 2018 *Surafel M. Lakew and Marcello Federico* FBK, Italy

	The Sogou-TIIC Speech Translation System for IWSLT 2018 *Yuguang Wang, Liangliang Shi, Linyu Wei, Weifeng Zhu, Jinkun Chen, Zhichao Wang, Shixue Wen, Wei Chen, Yanfeng Wang and Jia Jia* Sogou & TIIC, China
	Fine-tuning on Clean Data for End-to-End Speech Translation: FBK @ IWSLT 2018 *Mattia Antonino Di Gangi, Roberto Dessì, Roldano Cattoni, Matteo Negri and Marco Turchi* FBK, Italy
	Using Spoken Word Posterior Features in Neural Machine Translation *Kaho Osamura, Takatomo Kano, Sakriani Sakti, Katsuhito Sudoh and Satoshi Nakamura* NIST & RIKEN, Japan
	Transfer Learning in Multilingual Neural Machine Translation with Dynamic Vocabulary *Surafel Melaku Lakew, Aliia Erofeeva, Matteo Negri, Marcello Federico and Marco Turchi* FBK & UniTN, Italy
	Data Selection with Feature Decay Algorithms Using an Approximated Target Side *Alberto Poncelas, Gideon Maillette de Buy Wenniger and Andy Way* DCU, Ireland
18:30-19:30	**GUIDED TOUR THROUGH BRUGES**
19:30-	**DINNER IN THE BREWERY "DE HALVE MAAN"**

Tuesday, October 30th, 2018

09:00-10:00	**ORAL SESSION II**
09:00-09:30	**Multi-Source Neural Machine Translation with Data Augmentation** *Yuta Nishimura, Katsuhito Sudoh, Graham Neubig and Satoshi Nakamura* NIST & CMU, Japan, USA
09:30-10:00	**Word Rewarding for Adequate Neural Machine Translation** *Yuto Takebayashi, Chenhui Chu, Yuki Arase and Masaaki Nagata* OsakaU & NTT Corporation, Japan
10:00-10:15	**INDUSTRIAL SESSION**
10:00-10:15	**Gold sponsor presentation: Amazon**
Coffee Break (10:15-10:45)	
10:45-12:15	**ORAL SESSION III**
10:45-11:15	**Analyzing Knowledge Distillation in Neural Machine Translation** *Dakun Zhang, Josep Crego and Jean Senellart* Systran, France
11:15-11:45	**Learning to Segment Inputs for NMT Shows Preference for Character-Level Processing** *Julia Kreutzer and Artem Sokolov* HeidelbergU & Amazon, Germany
11:45-12:15	**Machine Translation Human Evaluation: a comprehensive investigation of evaluation based on Post-Editing and its relation with Direct Assessment** *Luisa Bentivogli, Mauro Cettolo, Marcello Federico and Christian Federmann* FBK & Amazon & Microsoft, Italy, USA
Lunch (12:15-13:30)	
13:30-15:00	**ORAL SESSION IV**
13:30-14:00	**Multi-paraphrase Augmentation to Leverage Neural Caption Translation** *Johanes Effendi, Sakriani Sakti, Katsuhito Sudoh and Satoshi Nakamura* NIST & RIKEN, Japan
14:00-14:30	**A Machine Translation Approach for Modernizing Historical Documents Using Back Translation** *Miguel Domingo and Francisco Casacuberta* UPV, Spain
14:30-15:00	**Word-based Domain Adaptation for Neural Machine Translation** *Shen Yan, Shahram Khadivi, Leonard Dahlmann, Pavel Petrushkov and Sanjika Hewavitharana* eBay Inc, USA
Coffee Break (15:00-15:30)	
15:30-16:45	**PANEL SECTION**
14:00-15:00	**Panel: The future of machine and speech translation** *Panelists: Evgeny Matusov (AppTek), Will Lewis (Microsoft), Kevin Duh (JHU), Marcin Junczys-Dowmunt (Microsoft) and Lucia Specia (University of Sheffield)* Moderator: Sebastian Stüker (KIT)
16:45-17:00	**CLOSING REMARKS + ANNOUNCEMENTS**

Keynotes

Multimodal Machine Translation

Lucia Specia, Imperial College London/Sheffield University

Abstract

Humans interact with the world through multiple modalities (hearing, vision, etc.). This is also true when understanding and generating language. Computational models for language processing, however, are traditionally limited to exploring language only (spoken or written). In this talk I will cover recent work in the area of multimodal machine learning for machine translation, where vision is used as additional modality, and where the goal is to achieve structured language grounding. I will also provide an overview on approaches that explore multimodality for language grounding in other sequence to sequence models: automatic speech recognition and spoken language translation.

Evaluation Campaign

The IWSLT 2018 Evaluation Campaign

J. Niehues[(1)] R. Cattoni[(2)] S. Stüker[(1)] M. Cettolo[(2)] M. Turchi[(2)] M. Federico[(3)†]

[(1)] KIT - Adenauerring 2, 76131 Karlsruhe, Germany
[(2)] FBK - Via Sommarive 18, 38123 Trento, Italy
[(3)] Amazon AI - East Palo Alto, CA 94303, USA

Abstract

The International Workshop of Spoken Language Translation (IWSLT) 2018 Evaluation Campaign featured two tasks: low-resource machine translation and speech translation. In the first task, manually transcribed speech had to be translated from Basque to English. Since this translation direction is a under-resourced language pair, participants were encouraged to use additional parallel data from related languages. In the second task, participants had to translate English audio into German text with a full speech-translation system. In the baseline condition, participants were free to use composite architectures, while in the end-to-end condition they were restricted to use a single model for the task.

This year, eight research groups took part in the low-resource machine translation task and nine in the speech translation task.

1. Introduction

We report here on the outcomes of the 2018 evaluation campaign organized by the International Workshop of Spoken Language Translation (IWSLT). The IWSLT workshop started in 2004 [1] with the purpose of enabling the exchange of knowledge among researchers working on speech translation and creating an opportunity to develop and compare translation systems on a common test bed. The evaluation campaign built on one of the outcomes of the C-STAR (Consortium for Speech Translation Advanced Research) project, namely the BTEC (Basic Travel Expression Corpus) multilingual spoken language corpus [2], which initially served as a primary source of evaluation. Since its beginning, translation tasks of increasing difficulty were offered and new data sets covering a large number of language pairs were shared with the research community. In the fifteenth editions organized from 2004 to 2018, the campaign attracted around 70 different participating teams from all over the world.

Automatic spoken language translation is particularly challenging for a number of reasons. On one side, machine translation (MT) systems are required to deal with the specific features of spoken language. With respect to written language, speech is structurally less complex, formal and fluent. It is also characterized by shorter sentences with a lower amount of rephrasing but a higher pronoun density [3]. On the other side, speech translation [4] requires the integration of MT with automatic speech recognition, which brings with it the additional difficulty of translating content that may have been corrupted by speech recognition errors.

Along the years, three main evaluation tracks were progressively introduced, addressing all the core technologies involved in the spoken language translation task, namely:

- Automatic speech recognition (ASR), *i.e.* the conversion of a speech signal into a transcript
- Machine translation (MT), *i.e.* the translation of a polished transcript into another language
- Spoken language translation (SLT), *i.e.* the conversion and translation of a speech signal into a transcript in another language

In previous years, the ASR transcript was provided to the participants of the SLT task. Therefore, the SLT task main focus on investigating translation methods for automatic transcripts.

The recent development in deep learning lead to the usage of similar techniques in machine translation and automatic speech recognition. Furthermore, the success of sequence-to-sequence model allowed the development of end-to-end speech translation systems [5]. Therefore, in this years edition, we drooped the ASR task and included the transcription of the audio into the SLT task. Hence, for the first time, participants needed to develop a full speech translation pipeline and were supplied with audio-text parallel data.

The 2018 IWSLT evaluation focused on translating talks from two sources of data: translation of TED talks corpus [6] and, for the speech translation task, university lectures collected at KIT [7].

The TED translation task of IWSLT has become a seasoned task by now. Its introduction was motivated by its higher complexity with respect to the previous travel tasks, and by the availability of high quality data. In order to keep the tasks interesting and to follow current trends in research and industry, we expanded and developed the IWSLT tasks further. Motivated by last years success of the multi-lingual machine translation task, we created task on a low resources

† Work performed while the author was at FBK, Italy.

language pair. Furthermore, we developed the speech translation task further. Participants need to build a complete speech translation system and we encourage research on end-to-end models. Unlike in previous years, we also limited the scope of the evaluation to very few languages. The main reason for this was to avoid dispersion of participants in too many tasks.

The translation directions considered this year were English to German for SLT task were English to German and Basque to English for low-resource MT task.

For all tasks, permissible training data sets were specified and instructions for the submissions of test runs were given together with the detailed evaluation schedule.

All runs submitted by participants were evaluated with automatic metrics. In particular, for the low-resource MT task, an evaluation server was set up so that participants could autonomously score their runs on different dev and test sets. This year, 15 groups participated in the evaluation (see Table 1). In the following, we describe each task in more details and provide in an appendix a detailed report of their results.

2. Low Resource Machine Translation

2.1. Definition

The Low Resource Translation Task addresses a conventional bilingual text translation task in the domain of the TED talks. Participants were required to translate TED talks from Basque to English. Given the difficulty of the proposed translation direction and the scarcity of available parallel data, additional parallel data from related languages were prepared.

Concerning Basque-English data, training set included 64 TED talks with 5.6K parallel sentences (81K Basque and 109K English tokens). Development set contained 10 talks with 1.1K parallel sentences (17K Basque and 23K English tokens). The valuation set *tst2018* consisted of 10 talks with 1.1K parallel sentences (15K Basque and 20K English tokens).

In-domain parallel training data included also talks from related languages: 73 talks for Basque-French, 74 for Basque-Spanish, 2595 for French-English, 2589 for Spanish-French and 2650 for Spanish-English. Moreover, an additional archive with the original xml files of all the TED talks available at April 2018 – excluding those in the tst2018 evaluation set – was provided. Finally, participants could download any data of the original TED talks from the TED website – excluding those in the tst2018 evaluation set.

Out-of-domain training data were restricted to parallel and monolingual corpora (including Basque data) provided by the OPUS[1] and WMT[2] organizations on their respective websites. Moreover, participants were allowed to utilize Basque-Spanish parallel and monolingual data from the Open Data Euskadi Repository kindly provided by the Vicomtech[3] research center.

In-domain training and development data were supplied through the website of the WIT3 ([6]), while out-of-domain training data were made available through the workshop's website.

2.2. Evaluation

Automatic translation of the test2018 tst2018 evaluation set were required to be in NIST XML format with case-sensitive, detokenized and punctuated texts. Translations quality was measured automatically by means of the three automatic standard metrics BLEU, NIST, and TER. Case sensitive scores were calculated with the software tools mteval-v13a.pl3 and tercom-0.7.254, by invocating:

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

It is worth noticing here that the two scoring scripts apply their own internal tokenization.

In order to allow participants to evaluate their progresses automatically and under identical conditions, an evaluation server was developed. Participants could submit the translation of the 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 core of the evaluation server is a shell script wrapping the mteval and tercom scorers. The REST service is implemented with a PHP script running over Apache HTTP Server, while the GUI on the web is written in HTML with AJAX code. The evaluation server was utilized also by the organizers for the automatic evaluation of the official submissions. After the evaluation period, the evaluation on the test2018 set was enabled to all participants as well.

2.3. Submissions

We received 15 submissions from 8 different participants (4 participants sent primary submissions only).

2.4. Results

The results on the tst2018 evaluation set for each participant are shown in Appendix A.1, sorted by the BLEU metric.

3. Speech Translation

3.1. Definition

In contrast to previous years, this year the participants needed to build the whole speech translation systems. The organizers did not provide any intermediate results as done in previous years. Instead, a baseline system was provided [8]. Participants were free to used parts of this system or purely rely on the own models.

[1] http://opus.nlpl.eu/

[2] http://www.statmt.org/wmt18/

[3] http://www.vicomtech.org

Table 1: List of Participants

ALIBABA	Machine Intelligence Technology Lab, Alibaba Group
APPTEC	Applications Technology (AppTek), Aachen, Germany
AFRL	Air Force Research Laboratory, United States of America
ADAPT	ADAPT Centre, Ireland
CUNI	Charles University - Institute of Formal and Applied Linguistics, Czechia
FBK	Fondazione Bruno Kessler, Italy
HY	University of Helsinky, Finland
JHU	Johns Hopkins University, Baltimore, USA
KIT	Karlsruhe Institute of Technology, Germany
MEMAD	Department of Digital Humanities / HELDIG University of Helsinki, Finland
	Department of Signal Processing and Acoustics Aalto University, Finland
PROMPSIT	Prompsit Language Engineering, Spain
SGNLP	NLP Laboratory in Sogang University, South Korea
SRPOL-UEDIN	Samsung R&D Institute Poland and University of Edinburgh, Poland/UK
TIIC	Voice Interaction Technology Center, Sogou Inc., Beijing, China
	Tiangong Institute for Intelligent Computing, Tsinghua University, Beijing, China
USTC-NEL	University of Science and Technology of China and IFLYTEK Co. LTD.

This year edition of the speech translation task contained two different conditions. In the first condition *Baseline*, the participants could use any architecture to generate the translations in the target language. The second condition *End-to-End* concentrated on end-to-end models. In this condition, participants need to train one large model to perform the whole process from source language audio to target language text.

In both tasks the same test data is used. The test data is English audio and needs to be translated into German. The test data consisted of two related types of data. One part of the training data are TED talks. These talks are well-prepared and address a broad audience. Therefore, they contain only very few disfluencies and contain only very few special terms. The second part of the test sets contain university talks and research presentations. Since the talks are targeted to a small target audience, the test sets contain more special terms.

For training the system, different data sources were provided to the participants. For training the ASR components, the TED LIUM corpus could be used [9]. For the training of the machine translation component, the data available form the WMT evaluation[4] was allowed. In addition, the organizers provide the WIT corpus []. Furthermore, for the first time, also a corpus to train the end-to-end corpus was provided. This corpus consists of English TED talks aligned with their German transcription[5].

3.2. Evaluation

Since the audio was not segmented by a human into sentence-like units, the generated translation were segmented into different sentences than reference transcript and translation. Therefore, in a first step of the evaluation we need 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 [10]. Two segmentation were generated, one used case information for the case-sensitive metrics and one using no case information for the case-insensitive metrics.

Using the resegmented input, we used 4 different metrics to evaluate the results. For BLEU [11] and TER[12], we calculated case-sensitive and case-insensitive scores. In addition, we calculated the BEER score [13] and the characTER [14].

3.3. Submissions

In total we received 27 submissions from 9 partners. We received 7 primary submissions in the baseline condition and 4 primary submissions in the end-to-end submission. Two participants submitted output to both conditions. The results of all primary submissions are summarized in Appendix A.1.

3.4. Results

The detailed results of the automatic evaluation in terms of BLEU, TER, BEER and characTER can be found in Appendix A.1.

4. Conclusions

We reported results of the 2018 IWSLT Evaluation Campaign which featured two tasks: the translation of TED talks from Basque to English and the speech translation task from English to German. In the second one, the test set contains TED talks as well as university lectures and research talks. In this task, two tracks were offered: a baseline condition and the end-to-end condition. In total, 14 international re-

[4] https://www.statmt.org/wmt18/

[5] http://i13pc106.ira.uka.de/ mmueller/iwslt-corpus.zip

search groups joined the evaluation campaign. For the first time, traditional pipeline approaches for speech translation were compared to end-to-end translation models.

5. Acknowledgements

6. References

[1] Y. Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, and J. Tsujii, "Overview of the IWSLT04 Evaluation Campaign," in *Proceedings of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2004, pp. 1–12.

[2] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world." in *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2002, pp. 147–152.

[3] N. Ruiz and M. Federico, "Complexity of spoken versus written language for machine translation," in *Proceedings of the 17th Annual Conference of the European Association for Machine Translation (EAMT)*, Dubrovnik, Croatia, 2014, pp. 173–180.

[4] F. Casacuberta, M. Federico, H. Ney, and E. Vidal, "Recent efforts in spoken language processing," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 80–88, May 2008.

[5] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," in *Interspeech 2017*, 08 2017, pp. 2625–2629.

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

[7] E. Cho, S. Fünfer, S. Stüker, and A. Waibel, "A corpus of spontaneous speech in lectures: The kit lecture corpus for spoken language processing and translation," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, N. C. C. Chair), K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Reykjavik, Iceland: European Language Resources Association (ELRA), may 2014.

[8] T. Zenkel, M. Sperber, J. Niehues, M. Müller, N.-Q. Pham, S. Stüker, and A. Waibel, "Open Source Toolkit for Speech to Text Translation," *The Prague Bulletin of Mathematical Linguistics*, vol. 111, pp. 125–135, October 2018. [Online]. Available: https://ufal.mff.cuni.cz/pbml/111/art-zenkel-et-al.pdf

[9] A. Rousseau, P. Deléglise, and Y. Esteve, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks," in *LREC*, 2014. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1104_Paper.pdf

[10] E. Matusov, G. Leusch, O. Bender, , and H. Ney, "Evaluating Machine Translation Output with Automatic Sentence Segmentation," in *Proceedings of the 2nd International Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, USA, 2005.

[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*. Association for Computational Linguistics, 2002. [Online]. Available: https://www.aclweb.org/anthology/P02-1040.pdf

[12] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of association for machine translation in the Americas*, 2006. [Online]. Available: https://www.cs.umd.edu/~snover/pub/amta06/ter_amta.pdf

[13] M. Stanojevic and K. Sima'an, "BEER: BEtter evaluation as ranking," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2014.

[14] 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: Volume 2, Shared Task Papers*. Association for Computational Linguistics, 2016. [Online]. Available: https://doi.org/10.18653%2Fv1%2Fw16-2342

Appendix A. Automatic Evaluation

A.1. Official Testset (*tst2018*)

· All the sentence IDs in the IWSLT 2018 testset were used to calculate the automatic scores for each run submission.
· MT systems are ordered according to the *BLEU* metrics.
· *WER*, *BLEU* and *TER* scores are given as percent figures (%).

Low Resource MT : Basque-English

System	BLEU	NIST	TER
SRPOL-UEDIN	26.21	6.51	59.49
HY	25.01	6.45	59.48
PROMPSIT	24.02	6.24	60.81
FBK	23.99	6.34	59.43
CUNI	22.86	6.10	60.31
ADAPT	13.89	4.46	69.98
AFRL	12.25	4.03	80.63
SGNLP	10.42	3.49	103.96

Speech Translation : English-German

System	BLEU	TER	BEER	characTER	BLEU(CI)	TER(CI)	#Words
Baseline condition							
TIIC	28.09	55.74	54.73	84.72	29.44	53.73	39611
USTC-NEL	26.47	58.03	52.69	92.24	27.86	55.98	38372
ALIBABA	22.36	63.03	51.77	69.26	24.23	60.22	39751
APPTEC	21.45	64.12	51.56	63.47	22.72	61.69	41210
KIT	19.44	67.94	50.61	58.16	20.78	65.52	42128
AFRL	17.24	69.10	49.23	64.27	18.37	66.78	41155
MEMAD	15.8	74.51	47.01	82.56	17.13	72.00	41848
End-to-End condition							
USTC-NEL	19.4	68.20	48.77	87.30	20.77	65.73	41372
FBK	10.24	78.20	40.68	129.47	11.16	76.38	36627
KIT	8.4	88.54	41.48	80.38	9.22	86.55	44155
JHU	5.45	89.59	35.46	99.89	6.09	88.20	40932

Speech Translation TED Only : English-German

System	BLEU	TER	BEER	characTER	BLEU(CI)	TER(CI)
TIIC	28.18	57.31	52.74	61.06	29.36	55.65
USTC-NEL	26.79	59.89	51.28	92.50	27.89	58.23
ALIBABA	22.77	63.66	50.62	65.54	24.57	60.96
APPTEC	21.05	66.31	49.96	60.96	22.17	64.20
KIT	18.84	69.05	48.73	57.97	20.02	66.92
AFRL	15.46	72.23	47.26	61.02	16.51	70.06
MEMAD	15.57	74.83	45.35	87.54	16.8	72.56
End-to-End condition						
USTC-NEL	18.32	70.50	46.65	88.73	19.58	68.36
FBK	9.75	77.57	38.98	150.35	10.57	75.95
KIT	7.99	86.68	39.55	86.36	8.82	84.76
JHU	4.51	85.84	32.71	112.77	4.97	84.63

Speech Translation Lecture Only : English-German

System	BLEU	TER	BEER	characTER	BLEU(CI)	TER(CI)
TIIC	27.55	54.25	57.43	117.57	29.06	51.89
USTC-NEL	25.95	56.24	54.59	91.89	27.6	53.82
ALIBABA	21.77	62.42	53.30	74.42	23.68	59.52
APPTEC	21.84	62.03	53.73	66.96	23.28	59.28
KIT	20.01	66.88	53.16	58.43	21.5	64.18
AFRL	18.94	66.12	51.92	68.77	20.13	63.65
MEMAD	16.01	74.20	49.25	75.65	17.44	71.46
End-to-End condition						
USTC-NEL	20.41	66.00	51.67	85.31	21.87	63.22
FBK	10.7	78.81	42.99	100.49	11.7	76.79
KIT	8.76	90.32	44.11	72.07	9.58	88.26
JHU	5.84	93.18	39.24	82.03	6.58	91.60

Unsupervised Parallel Sentence Extraction from Comparable Corpora

Viktor Hangya[1], *Fabienne Braune*[1,2], *Yuliya Kalasouskaya*[1], *Alexander Fraser*[1]

[1]Center for Information and Language Processing
LMU Munich, Germany
[2]Volkswagen Data Lab Munich, Germany
{hangyav, fraser}@cis.lmu.de
fabienne.braune@volkswagen.de

Abstract

Mining parallel sentences from comparable corpora is of great interest for many downstream tasks. In the BUCC 2017 shared task, systems performed well by training on gold standard parallel sentences. However, we often want to mine parallel sentences *without bilingual supervision*. We present a simple approach relying on bilingual word embeddings trained in an unsupervised fashion. We incorporate orthographic similarity in order to handle words with similar surface forms. In addition, we propose a dynamic threshold method to decide if a candidate sentence-pair is parallel which eliminates the need to fine tune a static value for different datasets. Since we do not employ any language specific engineering our approach is highly generic. We show that our approach is effective, on three language-pairs, without the use of any bilingual signal which is important because parallel sentence mining is most useful in low resource scenarios.

1. Introduction

The ability to extract parallel sentences from monolingual corpora is of great interest to the field and many approaches have been proposed [1, 2, 3, 4]. In this paper we explore ways to mine parallel sentences from monolingual data without bilingual supervision.

Our approach is based on bilingual word embeddings (BWEs) which represent words from different languages in the same vector space. While many authors leverage BWEs for parallel sentence extraction, previous work requires a strong bilingual signal to either (i) train the BWEs [5] (ii) train a classifier for sentence-pair extraction [6, 7, 8] or (iii) for feature engineering [9]. The disadvantage of these approaches is that the required bilingual signal is not available for many language pairs which is itself one of the reasons why parallel sentence extraction is important. In contrast to these approaches, our method does not need any bilingual signal. We create BWEs using post-hoc mapping [10] which allows us to leverage large amounts of (cheap) monolingual data to train good monolingual word embeddings (MWEs) which are then mapped into BWEs. We use the method proposed in [11] which combines adversarial training with post-hoc mapping [12] to learn BWEs without any bilingual signal. We show that high performance can be achieved using no parallel sentences nor any bilingual signal.

As a baseline system we produce sentence embeddings by averaging the word embeddings in the source language and target language sentences and compare them using cosine similarity. One difficult aspect of the task is that not all source sentences have a parallel target sentence, thus besides picking the most similar target sentence for a given source sentence it has to be decided if they are actually parallel. We propose a dynamic threshold method which calculates a minimum similarity value in an unsupervised fashion based on the input corpus.

Taking the average of the word embeddings in a sentence tends to give too much weight to irrelevant words [13]. Recently, various word-based sentence similarity metrics were introduced [14, 15]. The disadvantage of these methods is either that they are computationally expensive or that they do not handle irrelevant words. To overcome these issues, we propose a simple method which efficiently pairs source-target words while handling irrelevant words, thus making it feasible to process large datasets. In addition, we consider an important weakness of BWEs that was shown before [16], i.e., that they are poor at capturing the translations of named entities and rare words, showing that this is an important problem for parallel sentence extraction. We alleviate this by combining semantic similarities taken from BWEs with orthographic cues such as Levenshtein distance.

In summary, our contributions are: (i) We evaluate two approaches for parallel sentence extraction utilizing BWEs, based on sentence embeddings and word-by-word similarities respectively, which do not need any bilingual signal, in contrast with previous work. (ii) We introduce a dynamic threshold method for deciding whether a candidate sentence pair is parallel. (iii) We incorporate orthographic similarity to improve performance of parallel sentence extraction. (iv) We show the generality of our method on the German-English, French-English and Russian-English comparable corpora of the BUCC 2017 shared task [17].

2. Building Bilingual Word Embeddings

In this section we present two different scenarios to build BWEs. In particular, we use only monolingual datasets to train MWEs and we map them to the same bilingual space comparing two methods: the first only needs a small seed lexicon while the second does not rely on any bilingual signal.

2.1. Monolingual Word Embeddings

We train MWEs for all 4 languages in our test set. For this we used monolingual news crawls downloaded between 2011 and 2014 taken from the WMT 2014 shared task [18] containing around 80M, 117M, 31M and 45M sentences for English, German, French and Russian respectively. We used FastText skipgram [19] to train MWEs which computes a distributed representation of words using context and word structure information in the form of character n-grams. Settings used are: Embedding dimension 300; Minimum occurrence frequency 5; Window size 5; Character n-gram sizes between 3 and 6.

2.2. Bilingual Word Embeddings

Our approach to the task of parallel sentence extraction requires BWEs, which is a common vector space for words in two different languages. In previous research BWEs were created either from word-aligned, sentence-aligned or document-aligned parallel data [20, 21] or by using the cross-lingual reference to optimize two monolingual spaces, so called joint training [22, 23, 24]. Similarly to [9] we create BWEs using post-hoc mapping. First, we explain the basic idea of post-hoc mapping in the supervised setup and discuss the way how supervision is eliminated in the unsupervised method which our approach is based on.

Given two MWEs $\mathbb{R}^{d_S}$ and $\mathbb{R}^{d_T}$ post-hoc mapping is performed via a matrix $\mathbf{W} \in \mathbb{R}^{d_S \times d_T}$ which is learned using a bilingual seed lexicon. Each pair of words (s_i, t_i) in the lexicon, with $s_i \in V_s$ and $t_i \in V_t$, is projected into $\vec{x_i} \in \mathbb{R}^{d_S}$ and $\vec{y_i} \in \mathbb{R}^{d_T}$. W can be solved by learning a linear mapping [10]:

$$\mathbf{W}^* = \underset{\mathbf{W} \in \mathbb{R}^{\mathbf{d_S} \times \mathbf{d_T}}}{\arg\min} \; || \mathbf{XW} - \mathbf{Y} ||_{\mathbf{F}} \qquad (1)$$

where $\mathbf{X}$ and $\mathbf{Y}$ are obtained by concatenating all projections $\vec{x_i}$ and $\vec{y_i}$ of words in the seed lexicon. The authors of [12] showed that the mapping can be improved by enforcing an orthogonality constraint on $\mathbf{W}$ which can be achieved by solving the singular value decomposition of $\mathbf{YX}^T$. To achieve good performance a seed lexicon of around 5000 word-pairs is used.

$\mathbf{W}$ can also be solved without any explicitly bilingual signal. The system of [11] uses adversarial training i.e. a generator and discriminator framework to achieve this. The aim of the discriminator is to distinguish mapped source language embeddings $\mathbf{W}\mathcal{X}$ and target language embeddings $\mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are sets of embeddings of words coming from the source and target language. In contrast, the goal of the generator is to learn $\mathbf{W}$ such that it prevents the discriminator from making accurate predictions. After training, $\mathbf{W}$ is used to automatically extract a seed lexicon of best candidate word pairs which is used to perform post-hoc mapping with [12].

We use [11] in our fully unsupervised setup. As a contrastive experiment we report results with [12] using a seed lexicon of 5000 word pairs, which was used as a baseline in [11] as well.

3. Sentence Extraction

We evaluate our model on the shared task data provided by the BUCC Workshop at ACL 2017. We evaluate our system on De-En, Fr-En and Ru-En language pairs. The dataset consists of comparable monolingual corpora (Wikipedia dumps) where the BUCC organizers inserted truly parallel sentences (taken from News Commentary) into the monolingual data for each language pair [17]. Our task is to recover the truly parallel sentences, while minimizing false alarms.

3.1. Sentence Embeddings

We use a basic sentence embedding approach as a baseline. BWEs are used to embed sentences in both languages into the same space. Each sentence embedding is computed by dimension-wise averaging of the embeddings of words in the given sentence (contained in the BWEs) followed by $l2$ normalization. Once source and target sentences are embedded, their similarity can be efficiently computed via cosine similarity [25]. To overcome the issue of giving too much weight to semantically poor words, which decreases precision and mistakenly selects non-parallel sentences, we remove stopwords [26], digits and punctuation from texts before calculating sentence embeddings. Consider this erroneous example, which shows how weighting stop words like *a, in, by* too highly causes an erroneous match:

> **De:** *Inzwischen sterben mehr Frauen* ***an*** *Gebärmutterhalskrebs – alle zwei Minuten* ***eine*** *– als* ***bei einer*** *Entbindung.*
>
> **Gloss:** *Meanwhile die more women* ***from (literal: in)*** *ovarian cancer – every two minutes* ***one (literal: a)*** *– than* ***at (literal: by) a*** *birth.*
>
> **En:** *For women* ***in*** *the developing world,* ***by*** *contrast, dying* ***in*** *childbirth is simply* ***a*** *fact of life.*

3.2. Dynamic thresholding

To decide whether a candidate sentence pair, i.e., source sentence and its most similar target sentence, is parallel we introduce a method which calculates a minimum similarity value that the candidate has to meet. We calculate this threshold value for each test set with a simple formula:

$$th = \bar{S} + \lambda * std(S) \quad (2)$$

where S is a set containing the similarity values between each source sentence in the test set and its most similar target candidate, $\bar{S}$ and $std(S)$ are its mean and standard deviation. We set $\lambda = 2.0$ based on the De-En development set which worked optimally for the other setups as well. The advantage of this method is that it performed well on all our datasets, while fine tuning a static threshold value on the development sets did not achieve good results (see §4) due to the difference of development and test data. Note also that λ could be quickly and easily adjusted by the user in order to balance between quality and quantity for downstream tasks (in practice inspection of only a few samples is sufficient).

3.3. Bilingual dictionaries

Averaging word embeddings in a sentence tends to give too much weight to irrelevant words. It was shown that hub words, which are similar to a high proportion of other words, have negative effects on performance of embedding based methods [13]. *Word Mover's Distance* was introduced [14], which is based on the minimum distance that the words in one text need to "travel" to reach the words in the other text, to overcome such issues. On the other hand, the approach is computationally intensive which is not desirable in the case of parallel sentence extraction due to the high number of candidate sentence pairs. Furthermore, it was show that WMD performs similarly to maximum alignment based methods on monolingual sentence similarity tasks while the latter is computationally less intensive [15]. We propose an efficient hub word aware maximum alignment approach based on bilingual dictionaries and show that it is more effective than simple sentence embeddings. In this method, we perform bilingual lexicon induction on the trained BWEs to generate large n-best dictionaries, which we then use to mine parallel sentences.

3.3.1. Bilingual lexicon induction

Given a BWE representing two languages V_s and V_t, an n-best list of translations for each word $s \in V_s$ can be induced by taking the n words $t_i \in V_t$ whose representation $\vec{x_t}$ in the BWE is closest to the representation $\vec{x_s}$ according to cosine similarity.

From the source side of the comparable data we compute a list containing the 200,000 most frequent words. For each word in the list, we retrieve the 100-best translations using bilingual lexicon induction on the BWEs. Each translation is given a weight by using cosine similarity computed with the BWE.

3.3.2. Sentence extraction

Given a candidate pair of source and target sentences S and T, the similarity score is calculated by iterating over the words in S from left to right and pair each word s, in a greedy fashion, with the word $t \in T$ that has the highest similarity based on our dictionary. During iteration, we ignore all t which have been already paired to overcome the hubness problem, i.e. by preventing the pairing of multiple source words to the same target word. Then, the averaged word-pair similarity gives the final score. We apply the same stopword filtering as before and use dynamic thresholding for the final decision. Although, we kept our method simple for computational reasons we use pre-filtering as in previous work [6]. For each source sentence we only consider the 100 most similar target sentences as candidates based on sentence embedding similarities. Given a BWE model our method requires around 2.5 hours to process sentences from the De-En test set (164 billion sentence pairs) on a single thread.

3.3.3. Orthographic similarity

As it was shown in previous work the performance of bilingual lexicon induction can be significantly improved by using orthographic cues, especially for rare words. We extend this idea to the sentence level by using a dictionary containing orthographically similar source-target language word pairs and their similarity[1]. We define orthographic similarity as one minus normalized Levenshtein distance. We use this orthographic dictionary with BWE based dictionary when mining parallel sentences by using the bigger value from the two dictionaries. If the given word pair is not in a dictionary we consider their similarity as 0.0.

4. Results

As we mentioned earlier we evaluate our system on the De-En, Fr-En and Ru-En data of the BUCC 2017 shared task [17]. We show results based on BWEs created fully unsupervised with the method of [11] (**unsup**) and the lightly-supervised system of [12] (**lisup**) on the released training sets as in [8]. Our systems only rely on news crawl monolingual data and a small seed lexicon in case of the latter thus we did not use the training set in earlier steps. We will show the performance of our final system and results of previous supervised systems on the official test set at the end[2]. As baseline we use the sentence embedding system with stopword filtering and dynamic thresholding. We report precision, recall and F1-scores.

From table 1 it can be seen that the dictionary based approaches significantly outperform the baseline system for each language pair. Our systems perform best on Fr-En and lowest on Ru-En which strongly correlates with the performance of the mapping approaches, that was shown in [11], on bilingual lexicon induction. Even though the baseline performs the weakest it is competitive on French-English with similar systems [6]. We also ran our baseline system on De-En with *lisup* and static threshold value instead of dynamic.

[1] For speedup we only consider word pairs that have at least 0.8 similarity

[2] We evaluated on the test set by sending our predictions to the shared task organizers.

		De-En			Fr-En			Ru-En		
		P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
lisup	sentence-embedding	27.86	18.01	21.88	29.22	14.38	19.28	6.92	4.42	5.39
	BWE dict.	23.05	42.29	29.83	38.16	52.19	44.08	16.80	24.77	20.02
	BWE+ORT dict.	24.19	45.11	**31.49**	39.00	52.64	44.80	16.32	24.05	19.45
unsup	sentence-embedding	26.53	16.40	20.27	28.99	14.07	18.94	6.54	3.84	4.84
	BWE dict.	22.67	41.90	29.42	37.97	52.30	44.00	17.31	24.97	**20.44**
	BWE+ORT dict.	23.71	44.57	30.96	39.02	52.61	**44.81**	16.75	24.20	19.80

Table 1: Results of our proposed systems on the BUCC 2017 shared task's training set for the 3 language-pairs. Baseline is the sentence embedding based model with stopword filtering and dynamic tresholding. We underline the best F1 scores for a language-pair and BWE method and use **bolding** for the best overall F1 score for a given language-pair.

By fine tuning the value on the development set (achieving high score) we got only 2.69% F1-score on the training set showing the importance of dynamic thresholding. Furthermore, in our preliminary experiments we used a shuffled parallel dataset of parliament proceedings and news articles for BWEs as in previous work [6]. It showed that having strongly comparable data could give 5% performance gain for our setups in average. On the other hand, having access to such data is unrealistic in real life scenarios, so we don't use this data further in our work.

Comparing the dictionary based approaches with and without the orthographic dictionary it can be seen that the orthographic information helped the most for De-En and also increased performance for Fr-En. In the following example the incorrect En sentence is about the same topic but orthography was needed to extract the sentence with correct entities:

De: ***Microsoft*** *hat* ***Nokia Milliarden*** *von Dollar versprochen, wenn es seine* ***Smartphones*** *exklusiv mit* ***Windows Phone*** *ausstattet.*

En-: *In Q1 2008* ***Samsung*** *shipped 46.3* ***million mobile handsets*** *1Q 2008.*

En+: ***Microsoft*** *promised to pay* ***billions*** *of dollars for* ***Nokia*** *to use* ***Windows Phone*** *exclusively.*

On the other hand, it did not help for Ru-En because of their different character sets. We manually analyzed the results and saw that the use of orthographic information gave higher similarity scores to sentence-pairs that contained named entities with the same orthography. These pairs were correctly mined without orthography thus no performance increase was caused. On the other hand, higher similarity scores caused higher dynamic threshold value thus losing some correctly mined pairs. This phenomenon can be fixed by better fine tuning λ for this setup.

Our *lisup* and *unsup* systems are on par with each other. Regarding F1 scores, the seed lexicon caused higher performance only for De-En while the unsupervised method performed better for the rest of the language pairs. In figure 1 we show precision-recall curves comparing the two systems on the three language pairs. This also shows that their performance is similar. There is a bigger gap between the systems

		P (%)	R (%)	F1 (%)
De-En	[27]	88	80	84
	lisup BWE+ORT dict.	24	45	32
	unsup BWE+ORT dict.	24	45	31
Fr-En	[27]	80	79	79
	lisup BWE+ORT dict.	39	53	45
	unsup BWE+ORT dict.	39	53	45
Ru-En	lisup BWE+ORT dict.	16	24	19
	unsup BWE+ORT dict.	17	24	20

Table 2: Results on the test set. We show the best performing supervised system of the shared task [27].

in the case of Ru-En in higher precision ranges in favor of the unsupervised system. Overall, these results show that good performance can be achieved in a fully unsupervised manner, i.e., using only monolingual data for training BWEs and using only these for mining parallel sentence-pairs.

We show negative examples which our unsupervised system with orthographic information made on the De-En set in table 3. Examples 1-3 are incorrectly mined sentence pairs. In the first case the meaning of the mined pair is very similar although it is not parallel while high named entity content causes the error in the next two. Although names in example 2 are not orthographically similar they are close in embedding space which causes the error. Similarly, cardinal directions are different in example 3 but in general they appear in similar contexts thus get represented similarly in the word embedding space. In contrast, examples 4-6 have not been mined by our system. The first two pairs have extra information on the source side, although they are parallel, which caused error for our system. Example 6 contains the compound noun *Entwicklungsländern* (developing countries) which is not handled by our system.

Finally, we show results on the official shared task test sets in table 2[3]. For comparison we also include the results of the best performing supervised system on De-En and Fr-En [27]. There were no submissions for Ru-En. It can be seen that the performance of our systems on the test set are very close to the performance on the training set which we presented in table 1. This shows that our dynamic threshold approach, with λ tuned on the De-En development set, is gen-

[3]Results are rounded for consistency with the shared task paper.

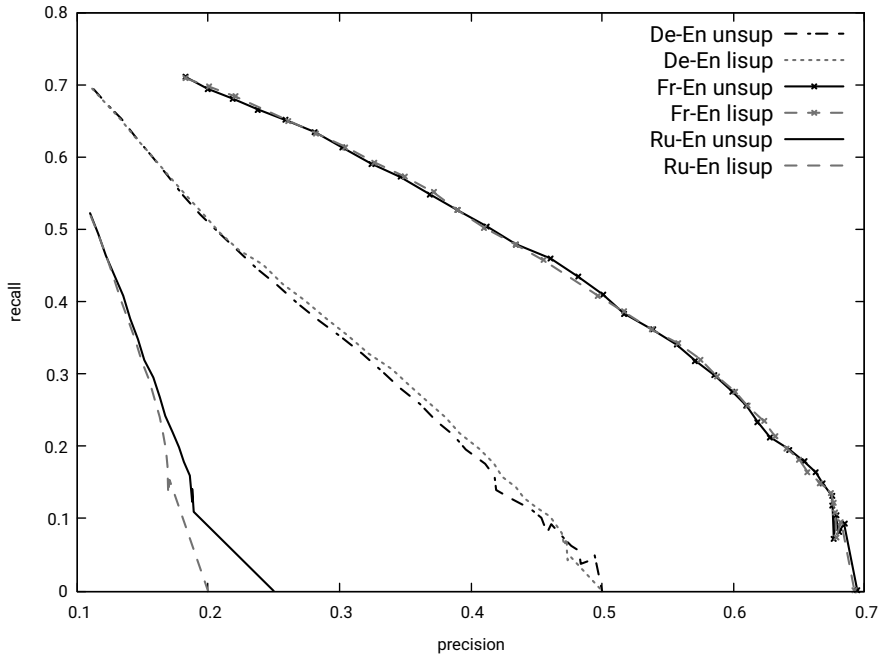


Figure 1: Precision-recall curves comparing unsup and lisup systems on the three language pairs.

1.	Das Werk wurde außerdem mit vier Academy Awards (Oscars) prämiert, darunter der Trophäe für den besten fremdsprachigen Film. In 2011, it was awarded the Academy Award for Best Documentary Feature at the 83rd Academy Awards. *The work has also received four Academy Awards (Oscars), including the Best Foreign Language Film Trophy.*
2.	Am 7. Juli 1957 wurde Angelas Bruder Marcus, am 19. August 1964 ihre Schwester Irene geboren. In April 1976 a daughter, Josina, was born, and in December 1978 a son, Malengane. *On July 7, 1957 Angela's brother Marcus was born, on August 19, 1964 her sister Irene.*
3.	Im Osten Serbien, im Südosten Montenegro, sowie im Norden, Westen und Südwesten Kroatien. It was in the modern Vojvodina (in northern Serbia), northern Croatia and western Hungary. *In east Serbia, southeast Montenegro, as well as in the north, west and southwest Croatia.*
4.	Aber die meisten Frauen, die Hillary Clinton wählen sollen, sind nicht Unternehmensjuristinnen oder Staatssekretärinnen. But most of the women sought as voters are not corporate attorneys or secretaries of state. *But most women who are to vote for Hillary Clinton are not corporate lawyers or state secretaries.*
5.	Durch diese Kürzungen ging jedoch die Produktion weiter zurück und die wirtschaftliche Misere verschlimmerte sich nur noch mehr. As they cut, output fell further and economic misery deepened. *As a result of these cuts, however, production continued to decline and the economic misery deepened.*
6.	Für Frauen in den Entwicklungsländern dagegen ist es ganz normal, bei der Entbindung sterben zu können. For women in the developing world, by contrast, dying in childbirth is simply a fact of life.

Table 3: Samples from the manual analysis. 1-3 are incorrectly mined examples (translation of De sentences where differing from En pair shown in italic) while 4-6 are the missed parallel sentences.

eral enough to work well on multiple languages and datasets. Interestingly, the supervised system performed better on De-En comparing with Fr-En while our approach reached higher F1 scores on the latter. One reason for this could be the better mapping quality of the word embedding space for Fr-En which was show in [11]. Our results with the fully unsupervised system are lower comparing to the supervised method since the latter has access to a large parallel corpus during training. Using parallel data supervised systems can learn features which help to decide if a sentence pair is parallel, e.g. word order of a source side phrase in the target side. In contrast, in the unsupervised case, we can only rely on word similarity information which can cause errors when syntax is the deciding factor in the case of a sentence-pair with similar words. On the other hand, our approach performed well on this task and will serve as a strong baseline for future unsupervised methods. With our dynamic thresholding method it is also easy to calculate a good initial threshold value which can be changed manually by the user in order to balance between quantity and quality of the mined sentence pairs.

5. Conclusion

In this work we introduced our first steps for the task of unsupervised parallel sentence extraction. We showed the performance of a simple sentence embedding system based on unsupervised BWEs and proposed a novel technique for dynamically setting the decision threshold. We improved upon this baseline system by proposing a simple word pair similarity based method which is efficient for large corpora. Furthermore, we addressed the shortcomings of BWEs when applying them for parallel sentence mining by using orthographic similarity. We showed that our system works well for various language pairs where BWEs could be built by achieving good results on De-En, Fr-En and Ru-En. In addition, we showed that unsupervised BWEs perform as well as BWEs based on a small seed lexicon. The goal of this short work is to provide a strong baseline for the unsupervised parallel sentence extraction task, and we are hoping to encourage more research on this important problem.

6. Acknowledgments

We thank Helmut Schmid for helpful discussions and comments. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement № 640550).

7. References

[1] D. S. Munteanu, A. Fraser, and D. Marcu, "Improved Machine Translation Performance via Parallel Sentence Extraction from Comparable Corpora," in *Proc. NAACL-HLT*, 2004.

[2] J. R. Smith, C. Quirk, and K. Toutanova, "Extracting Parallel Sentences from Comparable Corpora Using Document Level Alignment," in *Proc. NAACL-HLT*, 2010.

[3] C. Chu, R. Dabre, and S. Kurohashi, "Parallel Sentence Extraction from Comparable Corpora with Neural Network Features," in *Proc. LREC*, 2016.

[4] K. Krstovski and D. A. Smith, "Bootstrapping Translation Detection and Sentence Extraction from Comparable Corpora," in *Proc. NAACL-HLT*, 2016.

[5] J. Grover and P. Mitra, "Bilingual Word Embeddings with Bucketed CNN for Parallel Sentence Extraction," in *Proc. ACL, Student Research Workshop*, 2017.

[6] F. Grégoire and P. Langlais, "BUCC 2017 Shared Task: a First Attempt Toward a Deep Learning Framework for Identifying Parallel Sentences in Comparable Corpora," in *Proc. 10th Workshop on Building and Using Comparable Corpora*, 2017.

[7] H. Bouamor and H. Sajjad, "H2@ BUCC18: Parallel Sentence Extraction from Comparable Corpora Using Multilingual Sentence Embeddings," in *Proc. Workshop on Building and Using Comparable Corpora*, 2018.

[8] H. Schwenk, "Filtering and Mining Parallel Data in a Joint Multilingual Space," in *Proc. ACL*, 2018.

[9] B. Marie and A. Fujita, "Efficient Extraction of Pseudo-Parallel Sentences from Raw Monolingual Data Using Word Embeddings," in *Proc. ACL*, 2017.

[10] T. Mikolov, Q. V. Le, and I. Sutskever, "Exploiting Similarities among Languages for Machine Translation," *CoRR*, vol. abs/1309.4168, 2013.

[11] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, "Word Translation Without Parallel Data," *CoRR*, vol. abs/1710.04087, 2017.

[12] C. Xing, D. Wang, C. Liu, and Y. Lin, "Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation," in *Proc. NAACL-HLT*, 2015.

[13] G. Dinu, A. Lazaridou, and M. Baroni, "Improving Zero-Shot Learning by Mitigating the Hubness Problem," in *Proc. workshop track at international conference on learning representation*, 2015.

[14] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From Word Embeddings to Document Distances," in *Proc. ICML*, 2015.

[15] T. Kajiwara and M. Komachi, "Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings," in *Proc. COLING*, 2016.

[16] F. Braune, V. Hangya, T. Eder, and A. Fraser, "Evaluating bilingual word embeddings on the long tail," in *Proc. NACL-HLT*, 2018.

[17] P. Zweigenbaum, S. Sharoff, and R. Rapp, "Overview of the Second BUCC Shared Task: Spotting Parallel Sentences in Comparable Corpora," in *Proc. BUCC*, 2017.

[18] 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 *Proc. 9th Workshop on Statistical Machine Translation*, 2014.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *CoRR*, vol. abs/1607.04606, 2016.

[20] W. Y. Zou, R. Socher, D. M. Cer, and C. D. Manning, "Bilingual Word Embeddings for Phrase-Based Machine Translation." in *Proc. EMNLP*, 2013.

[21] K. M. Hermann and P. Blunsom, "Multilingual Distributed Representations without Word Alignment," in *Proc. ICLR*, 2014.

[22] S. Gouws, Y. Bengio, and G. Corrado, "BilBOWA: Fast Bilingual Distributed Representations without Word Alignments," in *Proc. ICML*, 2015.

[23] A. Klementiev, I. Titov, and B. Bhattarai, "Inducing Crosslingual Distributed Representations of Words," in *Proc. COLING*, 2012.

[24] H. Soyer, P. Stenetorp, and A. Aizawa, "Leveraging Monolingual Data for Crosslingual Compositional Word Representations," *CoRR*, vol. abs/1412.6334, 2014.

[25] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *CoRR*, vol. abs/1702.08734, 2017.

[26] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit," in *Proc. ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, 2002.

[27] A. Azpeitia, T. Etchegoyhen, and E. Martínez, "Weighted Set-Theoretic Alignment of Comparable Sentences," in *Proc. 10th Workshop on Building and Using Comparable Corpora*, 2017.

Word Rewarding for Adequate Neural Machine Translation

Yuto Takebayashi[†], *Chu Chenhui*[‡], *Yuki Arase*[†], *Masaaki Nagata*[*]

[†]Graduate School of Information Science and Technology, Osaka University
[‡]Institute for Datability Science, Osaka University
[*]NTT Communication Science Laboratories, NTT Corporation
{takebayashi.yuto, arase}@ist.osaka-u.ac.jp, chu@ids.osaka-u.ac.jp, nagata.masaaki@lab.ntt.co.jp

Abstract

To improve the translation adequacy in neural machine translation (NMT), we propose a rewarding model with target word prediction using bilingual dictionaries inspired by the success of decoder constraints in statistical machine translation. In particular, the model first predicts a set of target words promising for translation; then boosts the probabilities of the predicted words to give them better chances to be output. Our rewarding model minimally interacts with the decoder so that it can be easily applied to the decoder of an existing NMT system. Extensive evaluation under both resource-rich and resource-poor settings shows that (1) BLEU score improves more than 10 points with oracle prediction, (2) BLEU score improves about 1.0 point with target word prediction using bilingual dictionaries created either manually or automatically, (3) hyper-parameters of our model are relatively easy to optimize, and (4) under-generation problem can be alleviated in exchange for increasing over-generated words.

1. Introduction

Neural machine translation (NMT) [1, 2, 3] has dramatically improved machine translation quality compared to statistical machine translation (SMT). However, current NMT systems still suffer from the *adequacy* problem due to inappropriate lexical choice, under-generation, and over-generation [4]. In SMT, bilingual dictionaries have been used to improve adequacy in translation as decoder constraints. Typical example is the XML markup function implemented on MOSES [5].

Inspired by the decoding constraints for SMT, we propose a rewarding model using bilingual dictionaries to address the adequacy problem in NMT. Our model *rewards* target words that are promising to be used in correct translations by boosting their probabilities to be output by a decoder. It predicts such target words using bilingual dictionaries that are created manually or automatically. By applying byte pair encoding (BPE) [6] to dictionaries, our model can benefit from both BPE and dictionaries.

While previous studies incorporate bilingual dictionaries into NMT for translation of rare words [7, 8] and domain-specific words [9], we do so to improve the adequacy of NMT. Hence, dictionaries are made use of translating not only specific types of words but also all words. In addition, these are methodologically different; our model simply biases the trained decoder while previous models change the inside NMT architectures and require training of the entire systems. Due to this design, our model is easy to add to trained NMT systems and compatible with BPE.

Extensive evaluation on Japanese-to-English and English-to-Japanese translation has been conducted using two datasets; IWSLT (TED Talk) [10], spoken language domain with a small set of bilingual sentences (223k), and ASPEC [11], a scientific domain with a large set of bilingual sentences (3M). We refer to the former as a *resource-poor domain* and the latter as a *resource-rich domain,* hereafter. The results show that the rewarding model with oracle prediction of target words, where all and only target words in references are predicted, BLEU score improves more than 10 points on average in both of the resource-poor and resource-rich domains. When using bilingual dictionaries created manually or automatically in the rewarding model to predict target words, BLEU scores improve about 1.0 point on average in both domains.

Detailed analysis of our model reveals that it is relatively insensitive to settings of its hyper-parameters and easy to optimize. In addition, it is shown that our model decreases the number of under-generated words while tends to increase the number of over-generated words.

2. Neural Machine Translation

The encoder-decoder model with attention [3, 12] is one of the most popular architectures in NMT. It takes an input sentence $X = \{x_1, ..., x_n\}$ and generates its translation $Y = \{y_1, ..., y_m\}$ as:

$$p(Y|X;\theta) = \prod_{j=1}^{m} p(y_j|y_{<j}, X;\theta),$$

where θ is a set of parameters and $y_{<j} = \{y_1, \cdots, y_{j-1}\}$. Given a parallel corpus $C = \{(X, Y)\}$, the training objective minimizes the cross-entropy loss with regard to θ:

$$L_\theta = \sum_{(X,Y)\in C} -\log p(Y|X).$$

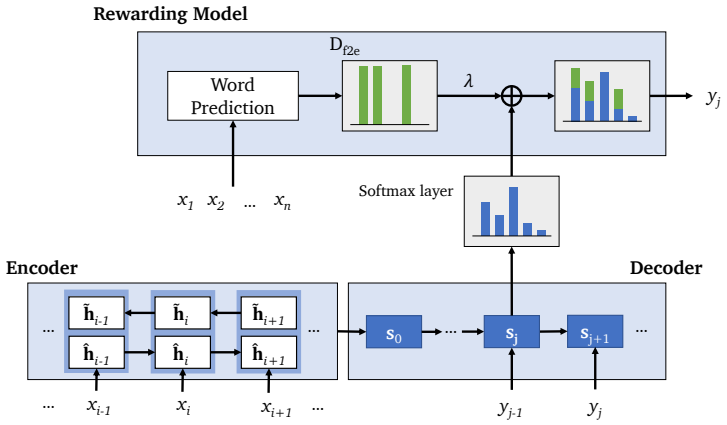


Figure 1: Rewarding model at decoding step j: predicted target words D_{f2e} are rewarded to have better chances to be output at each decoding time step. Note that the attention model is omitted for clarity.

The model consists of three parts, namely, an encoder, a decoder, and an attention model. The encoder has an embedding layer and an recurrent neural network (RNN) layer. The former converts words into their continuous space representations. Taking these embeddings, the RNN layer then computes a state that represents the input sequence till the current time step. Specifically, we use the bi-directional long short-term memory (LSTM) [13] that encodes the source sentence by forward and backward directions. At time step i, the state is represented by concatenating the forward hidden state $\hat{\mathbf{h}}_i$ and the backward one $\tilde{\mathbf{h}}_i$ as $\mathbf{h}_i = [\hat{\mathbf{h}}_i; \tilde{\mathbf{h}}_i]$. In this manner, X can be represented as $\mathbf{h} = \{\mathbf{h}_1, ..., \mathbf{h}_n\}$.

The decoder remembers all the history of translation and its softmax layer computes the posterior probability $p(y_j|y_{<j}, X)$ of a word y_j to output as translation. In order to focus on specific parts of the input sentence necessary for translation, the attention model is incorporated. We use the global attention mechanism proposed in [12].

3. Rewarding Model

On top of a decoder, our model rewards predicted words so that they have better chances to be output as translations as shown in Figure 1. Specifically, it first predicts a set of target words D_{f2e} that are promising to be used in translations using bilingual dictionaries. Then, our model *rewards* a target word if it is contained in D_{f2e} by adding weight to the posterior probability:

$$Q(y_j|y_{<j}, X) = \log p(y_j|y_{<j}, X) + \lambda r_{y_j}, \quad (1)$$

where λ is the weight of reward that will be tuned using a development set. This means that our model boosts the probabilities of predicted words that might have been slipped away during beam search in the conventional decoder. In [14], a similar rewarding model is proposed, but rewards are based on remaining sequence lengths.

We use a simple binary rewarding in this paper:

$$r_{y_j} = \begin{cases} 1 & (y_j \in D_{f2e}), \\ 0 & (\text{otherwise}). \end{cases} \quad (2)$$

We also tried to model the rewarding function using lexical translation probabilities that can be estimated for automatically created dictionaries. However, preliminary experiments empirically showed that this simple form of rewarding worked best. This may be because these probabilities are modeled in completely different ways, *i.e.*, $p(y_j|y_{<j}, X)$ in Equation (1) is conditioned on the entire source sentence while lexical translation probabilities are conditioned on source words. Further investigation is our future work.

Finally, a target word is output as:

$$y_j = \arg\max_{y_j} Q(y_j|y_{<j}, X).$$

Accurate prediction of D_{f2e} is crucial for our rewarding model. In the next section, we discuss practical implementations to obtain D_{f2e} from dictionaries.

4. Target Word Prediction with Dictionaries

In this study, we look up bilingual dictionaries created manually or automatically as word prediction, which allows to make our model minimally interact with the original NMT system. We will consider a sophisticated prediction model using an information in the encoder in future [15].

4.1. Prediction with Manually Created Dictionary

Thanks to the accumulated efforts by the academia and industry, bilingual dictionaries have been manually created for language pairs of English and Japanese. Such manual bilingual dictionaries provide reliable translation knowledge, although their coverage is limited. One disadvantage of manual dictionaries is that conjugation and derivative forms are generally not provided in such dictionaries. As a simple way to predict the target word set, we look up source words in a manual bilingual dictionary.

4.2. Prediction with Automatically Created Dictionary

Previous studies have proposed methods to automatically construct bilingual dictionaries. Especially, word alignment techniques for SMT [16, 5] allow us to construct a dictionary directly from a parallel corpus. Similar alignment may be possible using the attention model in NMT, however, reliability is not assured because the attention model is rather soft as a constraint [17, 18].

The biggest advantage of using word alignment for dictionary construction is that the domain of the dictionary matches that of translation targets. In addition, conjugations are available in the dictionary. A disadvantage is that alignment errors may decrease the quality of the dictionary.

We apply the GIZA++ toolkit[1] that is an implementation of the IBM alignment models [16] on a parallel corpus to automatically create a bilingual dictionary. To control the precision and recall of target word prediction, we introduce a threshold δ, which is tuned on development data. Target words with lower translation probability than δ are discarded.

4.3. Exact and Partial Matching with BPE

Conducting translation on sub-words is effective to address the unknown word problem [19]. We apply BPE [6] to dictionaries for word prediction to make our rewarding model compatible to BPE-based NMT. For both the dictionary entries and source sentences, we first apply a BPE model trained on a parallel corpus and then match the entries in dictionaries and source sentences.

We use two types of matching methods between an input sentence and dictionary entries: *exact match* and *partial match*. The former is precision-oriented and the latter is recall-oriented. After applying BPE, a dictionary headword (lemma) consists of multiple sub-words; a lemma w is denoted as $w = w_1, \ldots, w_k$. *Exact match* regards w as matched to a source sentence X if and only if: $w_1, \ldots, w_k \in X$, *s.t.*, for $\forall i \in \{1, \ldots, k-1\}, w_i = x_j \Leftrightarrow w_{i+1} = x_{j+1}$. On the other hand, *partial match* regards w as matched to X if $w_i \in X$ for $\exists w_i \in w$. In both matching methods, translations of w are added to the target word set as predictions. Obviously, target word predictions by *partial match* subsumes those by *exact match*.

5. Experiment Settings

To investigate the effects of our model, we conducted Japanese-to-English and English-to-Japanese translation experiments on resource-poor and resource-rich domains.

5.1. Translation Tasks

The resource-poor task used the IWSLT 2017 Japanese-English task from the WIT project [10]. The IWSLT task provides 223k parallel sentences for training. We used the dev 2010 and test 2010 sets for development and testing, containing 871 and 1,549 sentences, respectively.

The resource-rich task used the Japanese-English paper excerpt corpus (ASPEC)[2] [11], which is one subtask of the workshop on Asian translation (WAT)[3] [20]. For training, we used the first 2M parallel sentence pairs among the entire 3M pairs sentences following [21], because the remaining 1M sentences were noisy. The ASPEC task provides 1,790, and 1,812 sentences for development and testing, respectively. We conducted both Japanese-to-English and English-to-Japanese translation experiments on these two tasks, referred to as *IWSLT-JE*, *IWSLT-EJ*, *ASPEC-JE*, and *ASPEC-EJ* for short, hereafter.

5.2. NMT and Rewarding Model

We used the mlpnlp-nmt system[4] that is an LSTM based encoder-decoder NMT model with attention, which achieved the best translation performance in human evaluations for both the ASPEC-JE and ASPEC-EJ tasks at WAT 2017 [20].[5] We implemented our rewarding model on top of the mlpnlp-nmt system (our implementation will be public upon acceptance of the paper). We followed the hyper-parameter settings of [21]. The sizes of the source and target side embeddings, the LSTM hidden states, the attention hidden states were all set to 512. We used 2-layer LSTMs for both the encoder and decoder with beam size of 5. Stochastic gradient descent was used as the learning algorithm, with an initial learning rate of 1.0, gradient clipping of 5.0, and a dropout rate of 30% for the inter-layer dropout. The mini batch size was 128. The training epochs for IWSLT-JE, IWSLT-EJ, ASPEC-JE, and ASPEC-EJ were all set to 20, and we chose the model with the best development BLEU score among all the epochs as the baseline systems.[6]

For the rewarding models, λ in Equation (1) was tuned on the development sets from 0.1 to 1.0 by 0.1 interval. The threshold δ that prunes the automatically constructed dictionaries in Section 4.2 was tuned on 0, 0.0001, 0.001, 0.01 and 0.1. We selected the best combination among all combinations of δ and λ on the development set for each model.

We investigate the upper-bound performance of our rewarding model using oracle target word prediction. On this oracle model, predicted target words are all and only words in a reference translation, *i.e.*, precision and recall of prediction are both 100%. The best weight of λ was searched from 0.1 increasing the value by 0.1 until we observed a decrease in BLEU scores.

As preprocessing for the parallel corpora and bilingual dictionaries, we segmented Japanese sentences/entries using MeCab,[7] and tokenized and truecased the English sen-

[1] http://code.google.com/p/giza-pp

[2] http://lotus.kuee.kyoto-u.ac.jp/ASPEC/

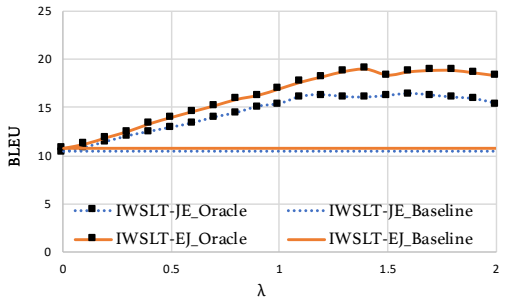
[3] http://orchid.kuee.kyoto-u.ac.jp/WAT/

[4] https://github.com/mlpnlp/mlpnlp-nmt/

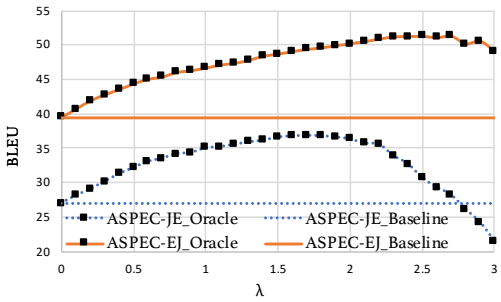
[5] Experiments on other NMT models as future work.

[6] Epoch #11, #20, #13 and #13 for IWSLT-JE, IWSLT-EJ, ASPEC-JE, and ASPEC-EJ, respectively.

[7] https://github.com/taku910/mecab



(a) IWSLT



(b) ASPEC

Figure 2: BLEU scores by the oracle rewarding model when changing the λ on the development set. BLEU scores dramatically improved on ASPEC task; 9.8 and 11.8 point improvements on ASPEC-JE and EJ, respectively.

tences/entries with the *truecase.perl* script in Moses[8] for both translation tasks. We further split the words into subwords using joint BPE [6] with $32,000$ merge operations. The vocabulary sizes of the IWSLT-JE task were $21,534$ and $18,022$, respectively. The vocabulary sizes of ASPEC-JE task were $28,852$ and $22,340$, respectively.

5.3. Bilingual Dictionaries

As the manual dictionary, we used EDR,[9] which is the publicly available English and Japanese bilingual dictionary.[10] The numbers of English-to-Japanese and Japanese-to-English entry pairs are 676k and $1,052$k, respectively. In EDR, only lemmas are provided and thus inflected forms of English verbs are unavailable. To address this issue, inflected forms of the EDR lemmas are extracted from the English dictionary of XTAG project,[11] which is used as the English morphological analysis dictionary for TreeTagger.[12] All the possible inflected forms are added into our dictionary.

For dictionary look-up, a source sentence is first lemmatized and matched with the dictionary. We used MeCab for Japanese and TreeTagger for English to lemmatize words.

To automatically construct bilingual dictionaries,[13] we used the GIZA++ toolkit on the training corpus in both English-to-Japanese and Japanese-to-English directions.[14] We applied the "grow-diag-final-and" heuristic and obtained lexical translation probabilities using Moses. We then prune translation pairs with low probabilities by δ.

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

[9]http://www2.nict.go.jp/ipp/EDR/ENG/indexTop.html?

[10]https://www.nict.go.jp/en/about/

[11]https://www.cis.upenn.edu/∼xtag/

[12]http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

[13]Note that the corpora used for building the dictionaries are the same as the one used for training each NMT systems. Other resources have not been used to create automatic dictionaries.

[14]Note that GIZA++ was applied on the parallel corpora without BPE, which was only used for look up a source word in a dictionary.

6. Results

We first investigate the effect of λ using the development sets on both the oracle target word sets and our word prediction methods. Next, we evaluate the translation quality on the test sets using the optimized λ. Finally, we conduct detailed analysis of translation results by our rewarding model.

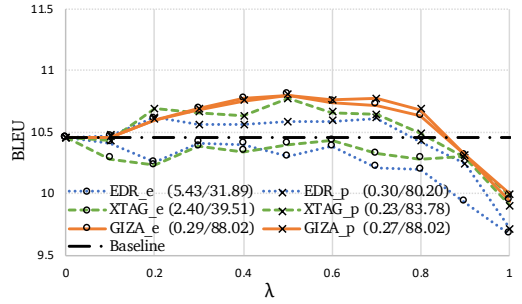
Throughout the section, the BLEU-4 score was used as the evaluation metric, which was computed using the *multi-bleu.perl* script in Moses on tokenized and truecased English and word-segmented Japanese sentences, respectively. The significance tests were performed using the bootstrap resampling [22] at $p < 0.01$.

6.1. Effects of λ

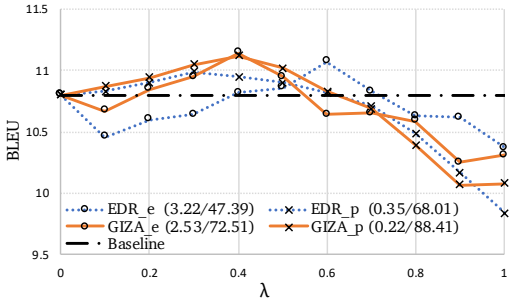
Figure 2 shows the BLEU scores by the oracle word rewarding on the development sets of the IWSLT-JE, IWSLT-EJ, ASPEC-JE, and ASPEC-EJ tasks. The BLEU scores significantly improved according to the λ. The best settings of λ improves 6.00, 8.25, 9.80, and 11.77 BLEU scores on the IWSLT-JE, IWSLT-EJ, ASPEC-JE, and ASPEC-EJ tasks from each baseline system, respectively.

Figure 3 shows the BLEU scores with respect to the λ and precision/recall of word prediction on our model with word prediction using manually or automatically created dictionaries. EDR indicates the models predicting target words using EDR. XTAG indicates the models using EDR extended with XTAG, which are only for the Japanese-to-English direction. GIZA indicates the models that predict target words using automatically constructed dictionary by GIZA++. The suffixes *e* and *p* in the legends indicate *exact match* and *partial match*, respectively.

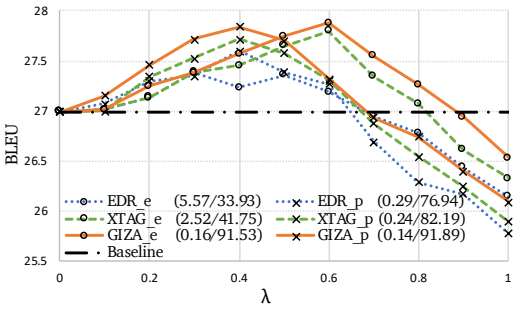
The results show that BLEU scores depend on precision and recall of target word prediction by different dictionaries. The weights of λ that achieved the best BLEU scores varied from 0.1 to 1.0. Notice that these weights are much smaller than the oracle prediction, which are 0.5, 0.4, 0.4, and 0.5 for IWSLT-JE, IWSLT-EJ, ASPEC-JE, and ASPEC-EJ on GIZA *partial-match*, respectively. This is because predicted words are less reliable and too much rewarding degrades the trans-



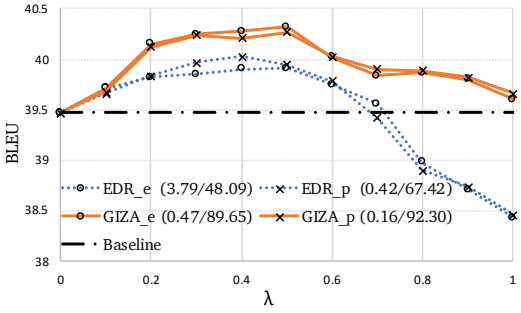
(a) IWSLT-JE



(b) IWSLT-EJ



(c) ASPEC-JE



(d) ASPEC-EJ

Figure 3: BLEU scores by our rewarding models with word prediction using bilingual dictionaries when changing the λ on the development sets. The gentle convex curves of BLEU scores show that the weight of λ is tunable by a simple grid search.

lation quality. The gentle convex curves of BLEU scores also show that λ is easily tunable using a simple grid search.

6.2. Word Prediction and Translation Results

Table 1 shows the comparison of BLEU scores on the test sets of the baseline and the rewarding models. We also report the results that use a merged dictionary. We chose the XTAG partial and GIZA partial for Japanese-to-English, EDR partial and GIZA partial for English-to-Japanese for merging because of their individual good performance. We tuned the λ for merged dictionary using the development set.

We can see that compared to the baselines, most of our methods significantly improve BLEU scores. Overall, a word prediction method with high recall shows a larger improvement in BLEU score as consistently shown by comparing exact matching *v.s.* partial matching, as well as comparing EDR *v.s.* XTAG, EDR or XTAG *v.s.* GIZA, and GIZA *v.s.* merged dictionary. However, there is still a gap between rewarding by our target word prediction and rewarding by oracle prediction. Our GIZA and merged dictionary models achieve a high recall of about 90% but a very low precision of 0.1%. Improving the precision for word prediction while keeping a recall high is our future work.

The baselines on ASPEC-JE and ASPEC-EJ are our reproduction of the state-of-the-art at WAT competition as single models, which are reported as achieved 27.62 and 39.71 BLEU scores in the paper. Compared to these scores, our rewarding model improved 0.67 and 0.36 points, respectively.

6.3. Under and Over Generation

We investigated the rate of under-generation and over-generation that are the major adequacy problems in NMT [23] using Translation Edit Rate (TER) [24]. TER aligns a reference and translation result. We counted the number of *Deletion* and *Insertion* regarding these are caused by under and over generation, respectively. This is an approximation to detect under and over generations, but we consider it is useful as an automatic and handy evaluation metric.

Table 2 shows the average numbers of under and over generations per sentence. The under-generation decreases on all the rewarding models in exchange of increasing over-generation. The rewarding model with oracle target word prediction reduces under generation about 1.2 word on average. This result shows that our rewarding model is also effective for alleviating the under-generation problem. The over-generation can be reduced by adding global constraint to the rewarding model, which prohibits rewarding the same predicted target. This is our future work.

Example translations of the baseline and our rewarding model (GIZA partial match) are shown in the following. The phrase of “congenital immunity” and “cancer of” were successfully translated by our model.

Source IL - 1 2 の 癌 に対する 抵抗 性 (先天 免疫) の 生物 反応 について も 考察 し た

Reference biological response of the resistance (*congenital immunity*) to *cancer of* IL - 12 was also examined .

		IWSLT				ASPEC			
		JE		EJ		JE		EJ	
		BLEU	(Pre. / Rec.)	BLEU	(Pre. / Rec.)	BLEU	(Pre. / Rec.)	BLEU	(Pre. / Rec.)
Baseline		9.97	(- / -)	10.26	(- / -)	27.21	(- / -)	39.50	(- / -)
EDR	exact	9.99	(5.19 / 30.58)	**10.75**	(3.14 / 46.08)	**27.82**	(5.57 / 34.10)	39.74	(3.70 / 48.33)
	partial	10.06	(0.27 / 79.50)	**10.73**	(0.33 / 67.09)	**27.94**	(0.28 / 77.15)	**40.05**	(0.42 / 67.37)
XTAG	exact	9.94	(2.25 / 38.10)	-	(- / -)	**27.73**	(2.50 / 41.97)	-	(- / -)
	partial	**10.30**	(0.20 / 82.88)	-	(- / -)	**28.00**	(0.23 / 82.42)	-	(- / -)
GIZA	exact	**10.36**	(0.27 / 85.98)	**10.88**	(2.56 / 72.92)	**28.29**	(0.15 / 91.48)	**39.96**	(0.46 / 89.73)
	partial	**10.32**	(0.25 / 87.61)	**10.83**	(0.21 / 87.38)	**28.28**	(0.13 / 91.80)	**40.07**	(0.15 / 91.65)
Merged dictionary		**10.33**	(0.16 / 89.25)	**10.81**	(0.17 / 88.45)	**28.29**	(0.14 / 91.77)	**40.05**	(0.17 / 92.12)
Oracle		**17.68**	(100 / 100)	**20.26**	(100 / 100)	**37.13**	(100 / 100)	**52.22**	(100 / 100)

Table 1: Comparison of BLEU scores on the test sets (The scores in bold indicate that the results are significantly better than the baseline at $p < 0.01$). The best improvement in BLEU score is 1.08 point when using GIZA *exact-match* in ASPEC-JE.

	under-generation				over-generation			
	IWSLT		ASPEC		IWSLT		ASPEC	
	JE	EJ	JE	EJ	JE	EJ	JE	EJ
Baseline	3.58	3.25	3.37	3.36	**1.64**	**2.04**	**2.27**	**1.69**
EDR_e	3.53	2.89	3.07	2.94	1.70	2.40	2.51	2.13
EDR_p	3.44	3.13	2.85	2.90	1.69	2.18	2.70	2.09
XTAG_e	3.48	-	2.92	-	1.90	-	2.64	-
XTAG_p	3.14	-	2.92	-	2.36	-	2.64	-
GIZA_e	3.18	2.90	2.75	**2.78**	2.33	2.58	2.67	2.15
GIZA_p	3.20	**2.86**	2.75	**2.78**	2.34	2.52	2.66	2.15
Oracle	**2.40**	**2.86**	**2.71**	3.01	4.26	2.80	3.04	3.06

Table 2: Numbers of under/over-generated words per sentence estimated by TER (The scores in bold indicate the best scores).

Baseline the biological response of the resistance to IL - 12 is also discussed .

Our Model the biological response of the resistance (*congenital immunity*) to the *cancer of* IL - 12 is also discussed .

7. Related Work

Our rewarding model can be viewed as a constraint on the decoder to output desired target words. There have been studies that aim to output predetermined words or phrases in neural language generation. For this purpose, the grid beam search in NMT is proposed [25] and the SMT lattice is combined into NMT [26]. In neural conversation generation, Wen et al. (2015) input a vector representing which information should be generated to an encoder [27], and a decoder is designed to explicitly control generation of emotional words [28].

Compared with these previous studies, one benefit of our rewarding model is that the predicted words are used as soft constraints on outputs with minimal interaction to the decoder. The most relevant study from the methodological point of view is [14] that also proposes a rewarding model in a decoder of NMT to improve the translation quality in general, such as remaining sequence lengths to output. We focus on the adequacy problem in NMT and combine word prediction with bilingual dictionaries. Some studies tackle the adequacy problem in NMT, but they require an independent SMT system [29, 30] or modification of the decoder [31]. Different from these, ours is simple and a cost-effective solution for the adequacy problem.

The under and over-generation problems have been recognized not only in NMT, but in other applications that use the encoder-decoder model for natural language generation. Different solutions have been proposed. First, a coverage vector is introduced in NMT [23, 32, 33] that tracks which source words have been translated by the attention mechanism. A sparse and constrained attention has been proposed [34], while word prediction, which are also used to reduce computational cost of softmax function at the decoder [35, 36], has been proposed to solve the under-generation problem. The decoder in [37] encourages to output predicted target words by initializing the decoder through word prediction, and the model in [38] predicts target words and their expected frequencies to resolve the under and over generation problems in NMT-based summarization.

8. Conclusion

We proposed a rewarding model with word prediction to boost the translation probabilities of the predicted target words that should be in correct translations. Our model allows incorporating bilingual dictionaries on a BPE-based NMT system. Extensive evaluation on both resource-poor and resource-rich domains showed its effectiveness.

As future work, first, we plan to improve the precision of word prediction preserving the recall at high. Second, we plan to improve our rewarding model to effectively incorporate translation probabilities and extend the model to reward not only words but also phrases. We will also consider a global constraint by predicting not only target words but their frequencies, and adjust rewards when a word has been used in translation. Finally, more experiments on datasets of various domains and language pairs will be conducted to investigate the generality of our approach.

9. References

[1] 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

[2] 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*. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969173

[3] 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.

[4] P. Koehn and R. Knowles, "Six challenges for neural machine translation," in *Proceedings of the First Workshop on Neural Machine Translation*. Vancouver: Association for Computational Linguistics, August 2017, pp. 28–39. [Online]. Available: http://www.aclweb.org/anthology/W17-3204

[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] 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

[7] P. Arthur, G. Neubig, and S. Nakamura, "Incorporating discrete translation lexicons into neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 1557–1567. [Online]. Available: https://aclweb.org/anthology/D16-1162

[8] J. Zhang and C. Zong, "Bridging neural machine translation and bilingual dictionaries," *CoRR*, vol. abs/1610.07272, 2016. [Online]. Available: http://arxiv.org/abs/1610.07272

[9] M. Arcan and P. Buitelaar, "Translating domain-specific expressions in knowledge bases with neural machine translation," *CoRR*, vol. abs/1709.02184, 2017. [Online]. Available: http://arxiv.org/abs/1709.02184

[10] 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.

[11] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara, "Aspec: Asian scientific paper excerpt corpus," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA), May 2016.

[12] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 1412–1421. [Online]. Available: http://aclweb.org/anthology/D15-1166

[13] 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

[14] J. Li, W. Monroe, and D. Jurafsky, "Learning to decode for future success," *CoRR*, vol. abs/1701.06549, 2017. [Online]. Available: http://arxiv.org/abs/1701.06549

[15] S. Ma, X. SUN, Y. Wang, and J. Lin, "Bag-of-words as target for neural machine translation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 332–338. [Online]. Available: http://www.aclweb.org/anthology/P18-2053

[16] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–312, 1993.

[17] L. Liu, M. Utiyama, A. Finch, and E. Sumita, "Neural machine translation with supervised attention," in *Proceedings of the 26th International Conference on Computational Linguistics (COLING2016)*, Osaka, Japan, December 2016, pp. 3093–3102.

[18] H. Mi, Z. Wang, and A. Ittycheriah, "Supervised attentions for neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 2283–2288. [Online]. Available: https://aclweb.org/anthology/D16-1249

[19] T. Luong, I. Sutskever, Q. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 11–19. [Online]. Available: http://www.aclweb.org/anthology/P15-1002

[20] T. Nakazawa, S. Higashiyama, C. Ding, H. Mino, I. Goto, H. Kazawa, Y. Oda, G. Neubig, and S. Kurohashi, "Overview of the 4th workshop on asian translation," in *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, November 2017, pp. 1–54. [Online]. Available: http://www.aclweb.org/anthology/W17-5701

[21] M. Morishita, J. Suzuki, and M. Nagata, "Ntt neural machine translation systems at wat 2017," in *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, November 2017, pp. 89–94. [Online]. Available: http://www.aclweb.org/anthology/W17-5706

[22] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 388–395.

[23] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," 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. 76–85. [Online]. Available: http://www.aclweb.org/anthology/P16-1008

[24] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of Association for Machine Translation in the Americas*, 2006.

[25] C. Hokamp and Q. Liu, "Lexically constrained decoding for sequence generation using grid beam search," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1535–1546. [Online]. Available: http://aclweb.org/anthology/P17-1141

[26] F. Stahlberg, A. de Gispert, E. Hasler, and B. Byrne, "Neural machine translation by minimising the bayes-risk with respect to syntactic translation lattices," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, April 2017, pp. 362–368. [Online]. Available: http://www.aclweb.org/anthology/E17-2058

[27] T.-H. Wen, M. Gasic, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young, "Semantically conditioned lstm-based natural language generation for spoken dialogue systems," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 1711–1721. [Online]. Available: http://aclweb.org/anthology/D15-1199

[28] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu, "Emotional chatting machine: Emotional conversation generation with internal and external memory," 2018.

[29] J. Zhang, M. Utiyama, E. Sumita, G. Neubig, and S. Nakamura, "Improving neural machine translation through phrase-based forced decoding," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, November 2017, pp. 152–162. [Online]. Available: http://www.aclweb.org/anthology/I17-1016

[30] L. Zhou, W. Hu, J. Zhang, and C. Zong, "Neural system combination for machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 378–384. [Online]. Available: http://aclweb.org/anthology/P17-2060

[31] 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. [Online]. Available: https://transacl.org/ojs/index.php/tacl/article/view/948

[32] F. Meng, Z. Lu, H. Li, and Q. Liu, "Interactive attention for neural machine translation," in *Proceedings of the 26th International Conference on Computational Linguistics (COLING2016)*, Osaka, Japan, December 2016, pp. 2174–2185.

[33] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[34] C. Malaviya, P. Ferreira, and A. F. T. Martins, "Sparse and constrained attention for neural machine translation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2018, pp. 370–376. [Online]. Available: http://aclweb.org/anthology/P18-2059

[35] X. Shi and K. Knight, "Speeding up neural machine translation decoding by shrinking run-time vocabulary," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 574–579. [Online]. Available: http://aclweb.org/anthology/P17-2091

[36] B. Sankaran, M. Freitag, and Y. Al-Onaizan, "Attention-based vocabulary selection for NMT decoding," *CoRR*, vol. abs/1706.03824, 2017. [Online]. Available: http://arxiv.org/abs/1706.03824

[37] R. Weng, S. Huang, Z. Zheng, X.-Y. DAI, and J. CHEN, "Neural machine translation with word predictions," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 136–145. [Online]. Available: https://www.aclweb.org/anthology/D17-1013

[38] J. Suzuki and M. Nagata, "Cutting-off redundant repeating generations for neural abstractive summarization," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, April 2017, pp. 291–297. [Online]. Available: http://www.aclweb.org/anthology/E17-2047

Analyzing Knowledge Distillation in Neural Machine Translation

Dakun Zhang, Josep Crego, Jean Senellart

SYSTRAN / 5 rue Feydeau, 75002 Paris, France

firstname.lastname@systrangroup.com

Abstract

Knowledge distillation has recently been successfully applied to neural machine translation. It allows for building shrunk networks while the resulting systems retain most of the quality of the original model. Despite the fact that many authors report on the benefits of knowledge distillation, few have discussed the actual reasons why it works, especially in the context of neural MT. In this paper, we conduct several experiments aimed at understanding why and how distillation impacts accuracy on an English-German translation task. We show that translation complexity is actually reduced when building a distilled/synthesised bi-text when compared to the reference bi-text. We further remove noisy data from synthesised translations and merge filtered synthesised data together with original reference, thus achieving additional gains in terms of accuracy.

1. Introduction

Neural machine translation (NMT) achieves state-of-the-art results in several translation tasks and for multiple language pairs [1, 2]. Equivalent to its phrase-based predecessor, neural networks directly learn from parallel bi-texts, consisting of large amounts of human created sentences with their corresponding translations. Therefore, the quality of an MT engine is heavily dependent on the amount and quality of parallel sentences.

Several techniques aimed at boosting the quality [3, 4] and quantity [5] of training data are successfully applied to neural MT. Parallel to these techniques, knowledge distillation [6] has attracted the focus of many researchers given its simplicity and the quality of the results. However, despite the fact that a growing number of private entities have begun to include distillation into their NMT systems [7, 8, 9] and that knowledge distillation has demonstrated its performance for multiple tasks [10, 11, 12], none of them give a detailed analysis of the reasons why it works.

In most cases, the availability of parallel corpora is a prerequisite to build Neural MT systems. The process of compiling parallel bi-texts is usually composed of several steps: crawling, filtering, cleaning, etc. As a result, parallel corpora usually contain parallel sentences that are often not as parallel as one might assume. And even for parallel sentences that truly convey the same meaning, in some cases translations follow a more or less word-for-word pattern (more literal translations). While in many other cases, translations show greater latitude of expression (more flexible translations) with higher degrees of variability, which humans often judge as good. However, machine translations are usually "closer" in terms of syntactic structure and present lower levels of variability when considering word choice. It is rather an intuitive idea that feeding more "literal" translations to a neural MT network should facilitate the training process compared to training with less literal translations (original bi-text).

In this paper, we report on the results of experiments where we automatically distill a human translation bi-text which is then used to train neural translation engines. Thus, aiming at boosting the learning ability of neural translation models. We show that the resulting models perform even better than a neural translation engine trained on the original reference dataset.

Our contribution is as follows:

- We analyse the reason why and how distillation works for neural machine translation.
- We analyse in detail the difference between original reference translations and synthesised translations.
- We further filter out noise from synthetic data and measure the impact on using both synthetic and reference translations.

The remainder of this paper is structured as follows. Section 2 briefly surveys previous work. Section 3 outlines our neural MT engine and details the distillation approach presented in this paper. Sections 4, 5 and 6 report training configurations and experimental results with detailed analysis. Section 7 draws conclusions and outlines future work.

2. Related Work

Sequential knowledge distillation for neural machine translation was first detailed by [6]. The authors trained a smaller student network to perform better by learning from a larger teacher network allowing more compact neural MT models to be built. [7] followed this idea and proposed a similar language simplification method based on distillation. They reported improvements on English to German and English to French translations. [8] further demonstrated distillation experiments from both an ensemble teacher model and a single

model. After that, they improved training efficiency and performance by removing noisy sentences from the training corpus. As a comparison, we performed detailed experiments and analysed the reasons why and how distillation works for neural machine translation.

Other than neural MT, [10] and [11] show that distillation also works well when transferring knowledge from a network ensemble or from a large highly regularised model into a smaller, distilled network for image classification and speech recognition. [12] applied distillation based on a search based structured prediction on dependency parsing and machine translation. These works demonstrate that knowledge distillation is adapted to many different tasks. In this work, we focus on the influence of knowledge distillation to neural machine translation and suggest directions for further improvements.

3. Neural Machine Translation

We train two types of NMT systems in this work, an RNN-based model and a Transformer-based model. The RNN model follows the architecture presented in [13]. It is implemented as an encoder-decoder network with multiple layers of an RNN with Long Short-Term Memory hidden units [14]. The Transformer model follows the work in [15]. It encodes the representation of sentences in a way a self attention only and is reported as the current state-of-the-art in many machine translation tasks [1, 9].

For the RNN model, the encoder is a bidirectional neural network that reads an input sequence $s = (s_1, ..., s_J)$ and calculates a forward sequence of hidden states $(\overrightarrow{h_1}, ..., \overrightarrow{h_J})$, and a backward sequence $(\overleftarrow{h_1}, ..., \overleftarrow{h_J})$. The decoder is an RNN that predicts a target sequence $t = (t_1, ..., t_I)$, being J and I respectively the source and target sentence lengths. Each word t_i is predicted based on a recurrent hidden state h_i, the previously predicted word t_{i-1}, and a context vector c_i. We employ the attentional architecture from [16] and use the implementation of *OpenNMT*[1]. Additional details are given in [17].

Unlike the RNN model, the Transformer model directly models the representations of each sentence with a self-attention mechanism. Hence, it reduces the number of operations related between tokens in different positions, especially for distant positions, in input and output sequence. The Transformer model stacks a so-called multi-head self attention layer and a position-wise, fully connected layer for non-linear conversions on both the encoder and decoder side. On the decoder side, it uses masked self-attention to prevent positions to attend to unseen positions.

The notion of time step is encoded automatically in the sequence in the RNN model. Whereas the Transformer model uses positional embedding to record the position information of each word in the sequence. In addition, the Transformer model is easy to be parallelized for the MLE training process across multiple GPUs. This allows the benefit of accelerating the training speed compared with the RNN model. In this work, we use the implementation of *OpenNMT-tf*[2] to train our Transformer based systems.

3.1. Knowledge Distillation

Knowledge distillation is a method to train different deep neural networks on the same data. Information learned from a large teacher model with the original reference data can be learned quite well with a smaller student model with the synthesised data [10]. Thus, a compact smaller model is generated and used to replace the larger model, especially in some resource limited devices.

For machine translation, we follow the approach described by [6]. The machine translation model is trained to minimise the Kullback-Leibler divergence, either between the model distribution and ground-truth distribution $\mathcal{L}_{NLL}$, or between the model distribution and synthesised data distribution $\mathcal{L}_{KD}$, which is from the teacher system, or an interpolation of both:

$$\mathcal{L} = (1 - \alpha) \cdot \mathcal{L}_{NLL} + \alpha \cdot \mathcal{L}_{KD}$$

In [6], three distillation methods are proposed by tuning the weight (α) in sequence-level loss. We simplify this process and perform experiments with $\alpha = 1$ in this work. First, we train a teacher system with the original source/target data. Second, we train another student system with the source/synthesised target data. The synthesised target language data is generated by running beam search (with beam size 5) over the training set with the teacher system (forward translation),

The objective during the training of the student system is the same as the teacher system. The only difference is the student system's objective is not to maximise log likelihood toward the ground-truth reference, but toward a generalised "soft" target, which is from the teacher system. From this point of view, the student system is directed by how the teacher system acts. Hence, in general a stronger teacher system is preferred. E.g. an ensemble teacher system is used in [8].

4. Experimental Conditions

4.1. Data

Experiments are performed using a preprocessed and tokenised version of WMT English-German translations[3]. The training set contains $4.5M$ sentence pairs. We use *newstest2013* as validation set and both *newstest2014* and *newstest2015* as test sets. We applied joint byte-pair encoding (BPE) [18] with $32K$ merge operations. The actual training vocabulary size is of $34K$ tokens after BPE tokenization. We

[1] https://github.com/OpenNMT/OpenNMT

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

[3] The corpus is already tokenised and can be downloaded from https://nlp.stanford.edu/projects/nmt

tfm	*L*arge	N=6, d=512, d_{ff}=4096, h=8
	*M*iddle	N=6, d=512, d_{ff}=2048, h=8
	*S*mall	N=4, d=256, d_{ff}=2048, h=8
rnn	*L*arge	Bi-LSTM, 4x1024, emb=512
	*M*iddle	Bi-LSTM, 2x1024, emb=512
	*S*mall	Bi-LSTM, 2x512, emb=512

Table 1: Configurations of the networks used in this paper. In the remainder of this paper we use respectively *tfm.L*, *tfm.M*, *tfm.S*, *rnn.L*, *rnn.M* and *rnn.S* to represent systems trained with different configurations.

limit sentence length to 100 in both source and target sides (excluding 0.31% of the training corpus). After decoding, we remove BPE joiners and evaluate the tokenised output with *multi-bleu.perl* [19].

4.2. Network Configuration

In this paper, we employs several neural MT models based on the Transformer [15] and RNN [13] models. Three different systems are used for each architecture, which differ in network size. Details of the system configurations are given in Table 1.

For RNN based systems, we use stochastic gradient descent, a mini-batch size of 64 in segments with dropout probability set to 0.3. We train our models during 18 epochs and evaluate the performance of the last epoch. Initial learning rate is set to 1.0 and we start decaying after epoch 10 by a fixed decay rate of 0.7. In decoding, we use a beam size of 5.

In the case of Transformer systems, we use Lazy Adam optimiser, which starts the learning rate at 1.0. We train the systems with a batch size of $8,192$ in tokens and save checkpoints in every $5,000$ steps. We terminate training after $400K$ iterations and average the last 8 checkpoints to get the final evaluation.

5. Analysis of Synthetic Translations

Aiming for a better understanding of the translated languages, we first conduct an elementary human analysis of the German hypotheses (synthesised translations) produced by the best performing network. We observe that in many cases, automatic translations produced by our neural MT systems consist of paraphrases of the reference translations. While both, reference and automatic translations, preserve the same meaning and are grammatically correct, automatic translations are closer in terms of syntactic structure to the source sentences than reference translations, which seams a key factor to train machine translation systems.

Examples in Table 2 illustrate this fact. In the first example, the English and German synthetic sentences follow a very similar structure. While in the German reference translation, the sentence: *you are sure to find the nightclub you like* is expressed by *clubbers (Disco-Gänger) are guaranteed to get their money (common garantiert auf ihre Kosten)*. In the second example, we can see a very similar situation. The verb *verwendet* is shifted to the end of the sentence when comparing the structure of the English and the German synthetic sentences. In contrast, the reference German translation employs a greater latitude of expression.

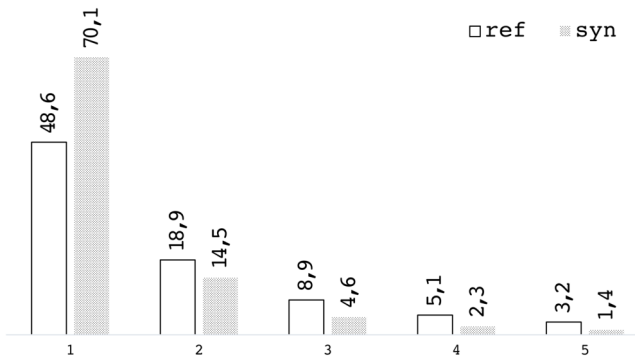


Figure 1: Histogram indicating the percentage (%) of source words aligned to n (x-axis) distinct target words in the training set.

Next, we conduct several experiments in order to confirm the hypothesis that automatic translations are closer to the input sentence than reference translations. We compare reference German translations (*ref*) to automatic translations (*syn*) produced by our neural MT network over the entire training set. In our experiments, we employ word alignments computed using `fast_align`[4].

5.1. Translation Fertility

First, we measure translation fertility. We identify the number of different target words aligned to each source word in the training corpus. We regard this number as the "fertility" between parallel sentences. Figure 1 shows a histogram indicating the percentage of source words aligned to n distinct target words.

As it can be seen, English words are in average related to less German words in the case of the automatic translations (*syn*) than for reference translations (*ref*). 70.1% of English tokens are aligned to a single German token in the case of automatic translations while this number is reduced to 48.6% in the case of reference translations. As expected, the opposite situation is also observed when considering target words aligned to multiple source words. In this case, reference translations show always a higher percentage of tokens.

5.2. Translation Distortion

In addition, we compare the translation distortion in order to validate the closeness (similarity) of syntactic structures. The translation distortion is calculated by the number of crossed alignments on automatic (*syn*) and reference (*ref*) translations. Given a sentence pair with its set of alignments, we

[4] `https://github.com/clab/fast_align`

Src:	[In Cala Ratjada]$_1$ [you are sure to find]$_2$ [a nightclub]$_3$ [you like]$_4$.
Ref:	**Disco-Gänger kommen** [in Cala Ratjada]$_1$ **garantiert auf ihre Kosten**.
Syn:	[In Cala Ratjada]$_1$ [finden Sie sicher]$_2$ [einen Nachtclub]$_3$, [den Sie mögen]$_4$.
Src:	[Your personal information]$_1$ [will only be]$_2$ [used]$_3$ [to process your booking]$_4$.
Ref:	**Sie** [werden nur]$_2$ **in dem Umfang weitergegeben , wie es** [für eine Buchung]$_4$ **notwendig ist**.
Syn:	[Ihre persönlichen Daten]$_1$ [werden nur zur]$_2$ [Bearbeitung Ihrer Buchung]$_4$ [verwendet]$_3$.

Table 2: Examples of English-to-German translation. Subscripts in these examples indicate the alignment between multi-words expressions.

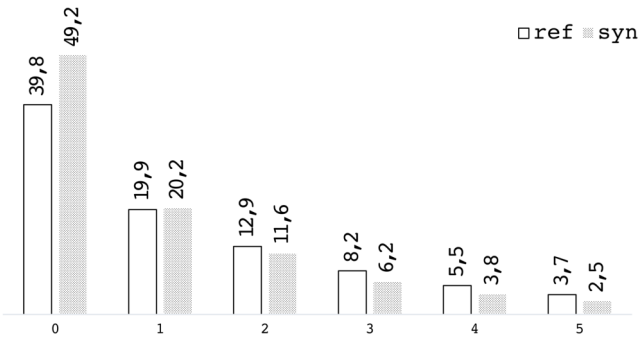


Figure 2: Difference in number of crossed alignments (in percentage (%)) between reference and synthesised translations. 0 means no crossed alignment, i.e. monotonic, between source and target sentences.

compute for each source word s_i the number of alignment crossings between the given source word and the rest of the source words. We consider that two alignments (i, j) and (i', j') are crossed if $(i-i')*(j-j') < 0$. Figure 2 illustrates the difference in number of crossed alignments between reference and synthetic translations.

As it can be seen, automatic (*syn*) translations show a higher number of words with no crossed alignments (49.2%) than reference (*ref*) translations (39.8%). In contrast, when considering larger numbers of crossings, the reference data set shows higher ratios than the automatic data set. This shows the synthesised target is much "closer" to the source in grammatical order compared with original reference.

Note that automatic translations carry important levels of noise (translation errors) that cannot be neglected from the view of a human. However, since it's the generalised output from the teacher system, it is indeed compatible to the machines. The next section evaluates the suitability of automatic translations as a training set for our neural MT systems compared to reference translations.

6. Results

6.1. Basic systems

We first summarise translation accuracy results (BLEU scores) of our 6 basic systems learned over the reference training set. As shown in Table 3, systems implementing the Transformer architecture outperform RNN networks. The best score is achieved by *tfm.L*, which is 27.87 for *newstest2014* and 30.04 for *newstest2015*. The smallest transformer network (*tfm.S*) clearly outperforms the largest RNN network (*rnn.L*) in about 0.5 BLEU points for *newstest2014*. We choose the best two systems *tfm.L* and *tfm.M* as the candidates for teacher systems.

Config	*newstest2014*	*newstest2015*
*tfm.L**	**27.87**	**30.04**
*tfm.M**	27.59	29.73
tfm.S	24.60	27.58
rnn.L	24.11	26.62
rnn.M	24.11	26.74
rnn.S	22.94	25.85

Table 3: BLEU scores on systems trained over the original dataset. Systems with * are candidates for teacher systems.

According to BLEU scores, similar performance is obtained by large and middle size versions of the Transformer and RNN models. However, the smallest systems show a clear drop in performance for both architectures. We argue that, when the network is big enough, the performance relies more on the amount of training data. That is, if the training data is fixed, there is a proper size of the neural network to learn the information embedded inside this training data. An even larger network could achieve a better performance, but not significantly.

6.2. Comparison between different teachers

For the distillation based method, the teacher system is important because its output will be used as the student's training reference. As shown in [8], a student system trained with an ensemble teacher usually performs better than that trained with a single teacher system. In our experiments, we train student systems based on a single teacher system. Table 3 shows that system *tfm.L* achieves similar accuracy compared with system *tfm.M* with original dataset. This means that there is no major difference between *tfm.L* and *tfm.M*, and these two systems can both be used as teacher systems. Therefore, in this section, we compare the performance between student systems trained on different teachers. We show that a strong teacher will lead to better students. Results are

Student System	Teacher System	*newstest 2014*	*newstest 2015*
tfm.S	*tfm.L*	26.07	28.96
	tfm.M	25.33	27.79
rnn.S	*tfm.L*	24.84	27.99
	tfm.M	24.22	27.10

Table 4: Results of comparison between different teacher systems.

shown in Table 4.

Both student systems outperform basic systems trained with original data (in Table 3). For *newstest2014*, systems trained with original reference data achieve 24.60 for model *tfm.S* and 22.94 for model *rnn.S*. When systems are trained with synthesised data from teacher system *tfm.L*, the performance improves to 26.07 (+1.47) and 24.84 (+1.90) respectively. This proves the distillation method works for neural machine translation.

We also notice that the difference between the two teacher systems *tfm.L* and *tfm.M* trained with original data is 0.28 for *newstest2014*. However, the difference between same student model trained with these two teachers increases to 0.74 for *tfm.S* and 0.62 for *rnn.S*. We argue that this is because of noisy data present in the synthesised target side. We found from the training synthesised data that some target sentences are exactly the same regardless of the source sentences. We further checked the original reference target sentence and confirmed that it is because the original bi-text is not parallel (noise in the original training data). During the training of the teacher system, neural models tend to normalise all these "bad" instances into a uniformed one by minimising the log likelihood. However, during the training of the student system, such noise is amplified and leads to the larger gap between the different systems. We therefore analyse in detail the influence of noise in the next section.

6.3. Influence of synthesised data noise

Based on Table 4, we can conclude that a stronger teacher is usually beneficial to train student systems. Similarly, we compared two similar student systems *rnn.M* and *rnn.S* based on teacher *tfm.L*. We found *rnn.M* can achieve 26.22 in BLEU score in *newstest2014*, which is +1.38 BLEU points higher than *rnn.S*. We therefore choose *tfm.L* as our teacher system and *tfm.S* and *rnn.M* as our default student systems in the following experiments. In this section, we train *tfm.S* and *rnn.M* with a different proportion of the synthesised data to see the influence of data noise for student systems.

As we showed in section 5, the synthesised data is the translation of the whole training set by a teacher model. Sequences in these generated hypotheses contain noise as they are from machine translated results. Noise includes ungrammatical sentences, wrong words selection, word ordering problems, etc. Also there are inconsistent target sentences with the source in semantic and under/over translation[5] problems in the synthesised data. We regard all these problems as noise because they are not correct translations.

	synthesised data	*newstest 2014*	*newstest 2015*
tfm.S	100%	26.07	28.96
	95%	**26.24**	**29.42**
	90%	26.20	29.21
rnn.M	100%	26.22	28.87
	95%	26.11	28.92
	90%	25.98	28.80

Table 5: Impact on different amounts of synthesised data for student systems by removing noisy data from the output of the teacher system *tfm.L*.

We use an embedding based method proposed by [20] to calculate the similarity between source and target sentences. In [20], a sentence embedding was first built based on word similarity, relying on a neural architecture, which is able to identify several types of cross-lingual divergences. The resulting embeddings are then used to measure semantic equivalence between sentences[6]. In our case, the target sentences are synthesised data from a teacher system. We filter out sentence pairs which are not similar based on the similarity score and train the student system with the remaining data. Table 5 shows the results from distilled *tfm.S* and *rnn.M* systems.

We compare two different student systems, Transformer based *tfm.S* and RNN based *rnn.M*. For *rnn.M* system, we can not see any gains by removing different proportions of noisy data. While for *tfm.S* system, when we remove 5% noisy data, we found an increase from 26.07 to 26.24 (+0.17) in BLEU score for *newstest2014*, which is also the best performance we have achieved until now.

We argue that the Transformer based system is more sensitive to the noisy data compared with the RNN based system. When a little amount noisy data (e.g. 5%) is removed, the performance improves because the remaining 95% is enough to train a good system. Along with the further reduction to 90%, both the Transformer based model and the RNN based model starts to decrease because there are fewer instances used for training. This also shows that the size of training data is another crucial factor to the final accuracy.

Another interesting phenomenon is that the differences between student systems with different architecture are not so big compared with systems trained with an original reference. In this experiment, *rnn.M* performs well compared with *tfm.S* given synthesised training data, while it is not the case for them to be trained with original data. We speculate that this is because the diversity in distilled bi-text is much

[5]During translating, under translation is when the words/phrases in the source sentence are missing (not translated) in the target side. Over translation is when there are duplicated translations for the same source words/phrases present on the target side.

[6]https://github.com/jmcrego/similarity

	merged data (hyp+ref)	*newstest 2014*	*newstest 2015*
tfm.S	95%+5%	**26.27**	29.09
	90%+10%	26.20	**29.11**
	80%+20%	25.76	28.88
	50%+50%	25.58	28.54
rnn.M	95%+5%	25.94	28.83
	90%+10%	25.52	28.76
	80%+20%	25.66	28.44
	50%+50%	25.04	27.60

Table 6: Results of merged corpus with synthesised data and original data. 95%+5% means the training data is composed of 95% data from the synthesised target translations (hypothesis) according to the similarity and additional 5% data from original target data (reference).

less than in the original reference. Systems with different architecture show different sensitivity of data diversity. That is also to say, the distilled bi-text is consistent and compact, which is much suitable for training machine translation systems.

6.4. Replacing noisy data with the original reference

Previous experiments show that systems trained with synthesised data usually perform better than systems trained with original reference data. At the same time, when we remove some noisy data from the synthesised training set, there is further improvement for the student system. In this section, we test experiments with merged synthesised data and the original reference as the training corpora to see which part is more crucial for the final performance.

First, we use a similarity score between source and target sentences calculated beforehand to rank the synthesised data. We select top $X\%$ "similar" data and for the remaining $(1 - X\%)$ data, we replace the target side with the original references to merge into a new data set. Table 6 shows the evaluation results on two student systems *tfm.S* and *rnn.M*.

Results show that synthesised data greatly contribute to the final accuracy. Along with the increase of data from the synthesised target side, the performance increases as well for both *tfm.S* and *rnn.M*. However, when comparing with systems trained with 100% synthesised data or systems trained with 95% synthesised data, the performance is different between the Transformer and RNN based models.

For *tfm.S*, when training with merged 95% data, the system reaches its highest performance in *newstest2014*. As for *rnn.M*, on the contrary, the performance starts to drop a little. It is even worse than training with the filtered 95% synthesised data. We argue this inconsistency stems from the architecture differences. The performance of the RNN based model is difficult to be improved. However, the Transformer-based model, due to its sensitivity to data diversity, can perform quite well as long as the training data is well controlled.

	2X data (hyp+ref)	*newstest 2014*	*newstest 2015*
tfm.S	100%+100%	25.95	28.74
rnn.M	100%+100%	25.78	28.85

Table 7: Results of the concatenated synthesised data and original reference. Twice the training cost is needed as the corpus is doubled.

6.5. Doubled training data

Lastly, we combine the synthesised data with the original reference. This will double the training data size and lead to twice the training cost. However, based on the results shown in Table 7, we found that even though the data size was doubled, we could not achieve further improvement.

We analyse that this is because the synthesised data is generated from the teacher system. All the information embedded in the synthesised data is already in the original data. In other words, adding such synthesised data is somewhat equivalent to adding the same original data. It is similar to training the system with the same data but with a 2X data size. Furthermore, considering noise existed in the synthesised data, systems trained with this 2X data is even worse than the original doubled data.

7. Conclusions

We have presented distillation experiments for neural machine translation. Results indicate the suitability of using synthetic translations to train neural MT systems. Higher accuracy results are obtained by the systems when trained using synthetic data. We show data noise present in both the original translation references and synthesised translations is a key factor that influence the final performance.

Meanwhile, the Transformer-based and RNN-based systems perform differently given different amounts of synthesised and/or merged data. We further prove that much "closer" translations contribute the most to the system's accuracy and that is also the reason why distillation works for neural machine translation.

In conclusion, we summarise that for neural machine translation:

- Having a stronger teacher system usually helps the student systems.
- Removing noise from the synthesised data of teacher systems also helps.
- Replacing noisy data with the original reference data can get further improvements.

A clear drawback of distillation-based methods is the efficiency of the training process. Student systems must be trained after the teacher system. In addition, we must also consider the cost of translating the entire training set. As

such, one solution is to integrate this procedure during the training process. Since data noise is one of the key factors during training, we believe that identifying noisy instances during training may alleviate the time problem. We leave that for future work.

8. Acknowledgements

We would like to thank all the anonymous reviewers for their insightful comments.

9. References

[1] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *CoRR*, vol. abs/1803.02155, 2018. [Online]. Available: http://arxiv.org/abs/1803.02155

[2] 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, August 2016.

[3] Y. Vyas, X. Niu, and M. Carpuat, "Identifying semantic divergences in parallel text without annotations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018, pp. 1503–1515. [Online]. Available: http://aclweb.org/anthology/N18-1136

[4] H. Schwenk, "Filtering and mining parallel data in a joint multilingual space," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2018, pp. 228–234. [Online]. Available: http://aclweb.org/anthology/P18-2037

[5] 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)*. Association for Computational Linguistics, 2016, pp. 86–96. [Online]. Available: http://www.aclweb.org/anthology/P16-1009

[6] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, November 2016.

[7] J. M. Crego and J. Senellart, "Neural machine translation from simplified translations," *CoRR*, vol. abs/1612.06139, 2016. [Online]. Available: http://arxiv.org/abs/1612.06139

[8] M. Freitag, Y. Al-Onaizan, and B. Sankaran, "Ensemble distillation for neural machine translation," *CoRR*, vol. abs/1702.01802, 2017. [Online]. Available: http://arxiv.org/abs/1702.01802

[9] A. Birch, A. Finch, M.-T. Luong, G. Neubig, and Y. Oda, "Findings of the second workshop on neural machine translation and generation," *arXiv preprint arXiv:1806.02940*, 2018.

[10] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[11] J. H. Wong and M. J. Gales, "Sequence student-teacher training of deep neural networks," 2016.

[12] Y. Liu, W. Che, H. Zhao, B. Qin, and T. Liu, "Distilling knowledge for search-based structured prediction," *CoRR*, vol. abs/1805.11224, 2018. [Online]. Available: http://arxiv.org/abs/1805.11224

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014, demoed at NIPS 2014: http://lisa.iro.umontreal.ca/mt-demo/. [Online]. Available: http://arxiv.org/abs/1409.0473

[14] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *CoRR*, vol. abs/1409.2329, 2014. [Online]. Available: http://arxiv.org/abs/1409.2329

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[16] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 1412–1421. [Online]. Available: http://aclweb.org/anthology/D15-1166

[17] J. Crego, J. Kim, G. Klein, A. Rebollo, K. Yang, J. Senellart, E. Akhanov, P. Brunelle, A. Coquard, Y. Deng, S. Enoue, C. Geiss, J. Johanson, A. Khalsa, R. Khiari, B. Ko, C. Kobus, J. Lorieux, L. Martins, D. Nguyen, A. Priori, T. Riccardi, N. Segal, C. Servan, C. Tiquet, B. Wang, J. Yang, D. Zhang, J. Zhou, and P. Zoldan, "Systran's pure neural machine translation systems," *CoRR*, vol. abs/1610.05540, 2016. [Online]. Available: http://arxiv.org/abs/1610.05540

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

[19] 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.

[20] M. Q. Pham, J. Crego, J. Senellart, and F. Yvon, "Fixing translation divergences in parallel corpora for neural mt," *Conference on Empirical Methods in Natural Language Processing*, 2018.

Word-based Domain Adaptation for Neural Machine Translation

Shen Yan, *Leonard Dahlmann*, *Pavel Petrushkov*, *Sanjika Hewavitharana*, *Shahram Khadivi*

eBay Inc.

{shenyan,fdahlmann,ppetrushkov,shewavitharana,skhadivi}@ebay.com

Abstract

In this paper, we empirically investigate applying word-level weights to adapt neural machine translation to e-commerce domains, where small e-commerce datasets and large out-of-domain datasets are available. In order to mine in-domain like words in the out-of-domain datasets, we compute word weights by using a domain-specific and a non-domain-specific language model followed by smoothing and binary quantization. The baseline model is trained on mixed in-domain and out-of-domain datasets. Experimental results on En $\rightarrow$ Zh e-commerce domain translation show that compared to continuing training without word weights, it improves MT quality by up to 3.11% BLEU absolute and 1.59% TER. We have also trained models using fine-tuning on the in-domain data. Pre-training a model with word weights improves fine-tuning up to 1.24% BLEU absolute and 1.64% TER, respectively.

1. Introduction

Domain adaptation (DA) techniques in machine translation (MT) have been widely studied. For statistical machine translation (SMT), several DA methods have been proposed to overcome the lack of domain-specific data. For example, self-training [1, 2] uses a MT system trained on general corpus to translate in-domain monolingual data as additional training sentences. Topic-based DA [3, 4] employs topic-based translation models to adapt for different scenarios. Data selection approaches [5, 6, 7, 8] first score the out-of-domain data using language model trained on both domain-specific and non-domain-specific monolingual corpora, then rank and select the out-of-domain data that are similar to in-domain data. Instance weighting methods [9, 10] score each sentence/domain using statistical rules, then train the MT models by giving sentence/domain-level scores.

Neural machine translation (NMT) has become state-of-the-art in recent years [11, 12, 13, 14, 15]. There are several research works on NMT domain adaptation. For example, back-translation methods [16] use a NMT model trained on the reverse direction to translate domain-specific monolingual data as additional training sentences. Fast DA approaches [13, 17] train a base model using mixed in-domain and out-of-domain datasets, then fine-tuning on in-domain datasets. Mixed fine-tuning [18] combines fine-tuning and multi-domain NMT. Similar to instance weighting in SMT, sentence/domain weighting methods [19, 20] can also be used for NMT domain adaptation based on different objectives. DA with meta information [21] is proposed to train topic-aware models using domain-specific tags for the decoder. Chunk weighting method [22] describes a way of selecting and integrating positive partial feedback from model-generated sentences into NMT training.

In this paper, we propose word-level weighting for NMT domain adaptation. We compute the word weights in out-of-domain datasets based on the logarithm difference of probability according to a domain-specific language model and non-domain-specific language model followed by smoothing and binary quantization. This gives the in-domain words in out-of-domain sentences higher weights and biases the NMT model to generate more in-domain-like words. Thus, the work presented in this paper can be viewed as a generalization of instance weighting. To remove noise in the word weights, we study the effectiveness of using smoothing methods. Specifically, a weighted moving average filter is proposed to apply smoothing to the computed word scores with its nearby words.

Experiments on En $\rightarrow$ Zh e-commerce domain translations tasks show that: 1) Domain adapted model with smoothed word weights gains significant improvement over non-smoothed weights; 2) Continuing training the model with computed word weights improves translation results significantly compared to continuing training without word weights; and 3) Compared to directly fine-tuning on in-domain datasets, fine-tuning after pre-training with word weights results in translation performance improvement on the in-domain e-commerce test set.

The rest of the paper is structured as follows. The approach and model we use is described in Section 2, where we first recap the NMT objective and then present the details of the proposed word-level weighting approach. Experimental results and discussions are presented in Section 3 and Section 4, followed by conclusions and outlook in Section 5.

2. Approach

We present word weighting objective on NMT before discussing how to generate the weights.

2.1. Objective

In this work we use attention-based neural machine translation model [11, 12, 14] for experiments. Given a parallel bilingual dataset *D*, the NMT model is trained to maximize the conditional likelihood *L* of a target sequence $y_1^T : y_1, \ldots, y_T$ given a source sequence $x_1^N : x_1, \ldots, x_N$:

$$L = \sum_{(x_1^N, y_1^T) \in D} \sum_{t=1}^{T} \log p(y_t | y_1^{t-1}, x_1^N) \tag{1}$$

Training objective (1) can be simply modified to word-level loss L_w with word weights w_t:

$$L_w = \sum_{(x_1^N, y_1^T, w_1^T) \in D} \sum_{t=1}^{T} w_t \log p(y_t | y_1^{t-1}, x_1^N) \tag{2}$$

The word weights w_t for a target sequence y_1^T can be 0 or 1. We set w_t = 1 for all in-domain sentences. For out-of-domain sentences, w_t = 1 means the word in the out-of-domain sentence is related to in-domain datasets (selected), w_t = 0 means it is not.

Our training objective (2) can be seen as a generalization of the original training objective (1) and instance weighting methods [19, 20]. The original loss (1) sets w_t = 1 for every word in all sentences. The instance-level loss can be expressed as giving a target sentence, $w_t = w \; \forall t$, where w is the weight for the sentence or the domain. Our training objective is similar to [22], however, instead of generating chunk-based user feedback for model predictions, we compute the word weights using language models trained on real target data.

2.2. Approaches to the objective

To compute discriminative word weights, we first follow the data selection methods in SMT [5]. To state this formally, let I be the domain-specific corpus, O be the non-domain-specific corpus, and y_t be the word in out-of-domain sentences at target position t. We denote by $P_I(y_t | y_{t-n}^{t-1})$ the per-word probability conditioned on previous $n-1$ words, according to a language model trained on I. Similarly, we denote by $P_O(y_t | y_{t-n}^{t-1})$ the per-word probability conditioned on previous $n-1$ words according to a language model trained on O. We can estimate $P_I(y_t | y_{t-n}^{t-1})$ and $P_O(y_t | y_{t-n}^{t-1})$ by training language models on I and O, separately. Therefore, the word scores s_t can be computed in the log domain:

$$s_t = \log P_I(y_t | y_{t-n}^{t-1}) - \log P_O(y_t | y_{t-n}^{t-1}) \tag{3}$$

Since the value of s_t is strongly correlated with the neighborhood words, it is worth investigating smoothing of the word scores before binary thresholding to remove the noise. Hence, a weighted moving average kernel:

$$\hat{s_t} = \sum_{k=\lfloor \frac{-L}{2} \rfloor}^{\lfloor \frac{L}{2} \rfloor} c_k s_{t+k} \tag{4}$$

is then applied to smooth word score s_t at each target position t. Here L is the kernel size and c_k are values of the kernel for $k \in [\frac{-L}{2}, \frac{L}{2}]$. In our experiments, we heuristically set the values of the kernel based on mean average with $c_k = c = \frac{1}{L}$ or gaussian distribution with $c_k = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-k^2}{2\sigma^2}}$, where we set σ to be the global variance of the word scores.

The special case of sentence-level weights can be expressed as $\hat{s_t} = \hat{s} \; \forall t$, where $\hat{s}$ is the averaged smoothed word scores for the target sentence y_1^T. In this case, the training objective (2) becomes equivalent to sentence weighting method from [20] with appropriately modified scoring function.

After smoothing the word scores, we finally binarize the smoothed word scores based on a threshold T:

$$w_t = \begin{cases} 1, & \text{if } \hat{s_t} \geq T \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

In our experiments we set the threshold $T = 0.5$ and only keep the words above the threshold. This means we select a word if $w_t = 1$ and do not select it if $w_t = 0$. Considering word weights w_t are gathered in a binary form during continuing training, the selected words would be good candidates that we want to extract from out-of-domain corpus O. In fact, word weights w_t are precomputed offline and used during the training. It can be set to any real value, depending on the way of thresholding.

2.3. Chunk-based weighting

Considering that the selected words in a target sentence might still be noisy and we select single random words, we alternatively experimented with selecting only the part (chunk) in the target sentence that has the longest consecutive weights (LCW) with $w_t = 1$. For each target sentence, we pick only one chunk and set all other weights to zero. See Figure 1 for an example. Then, because the surrounding context is also selected, the chunk is less likely to be noise. If there are multiple such chunks with the same length in the sentence, we simply randomly sample one of them. We found that the chunk-based approach in practice performs slightly better than word-level weighting.

3. Experiments

In this section, we conduct a series of experiments to study how well NMT performs when word-level weights are given for out-of-domain training data. We also study the effectiveness of the smoothing methods.

3.1. Datasets and data processing

We report the results on our in-house English-to-Chinese e-commerce item descriptions dataset. Item descriptions are

provided by private sellers and like any user-generated content, may contain ungrammatical sentences, spelling errors, and other type of noise. We first segmented the Chinese sentences with Stanford Chinese word segmentation tool [23] and tokenized English sentences with the scripts provided in Moses [24]. On both languages, we use subword units based on byte-pair encoding (BPE) [25] with 42,000 subword symbols learned separately for each language. For En-Zh we have 0.53M in-domain e-commerce sentence pairs and 5.15M sampled out-of-domain sentence pairs (UN, subtitles, TAUS data collections, etc.) that have significant n-gram overlap with the item description data. We validate our models on an in-house development set consisting of 3173 item descriptions, and evaluate on an in-house test set of 739 item descriptions using case-insensitive character-level BLEU [26] and TER [27] with in-house tools. For development and test sets, a single reference translation is used. Statistics of the data sets are reported in Table 1.

To compute our word weights we train a domain-specific 4-gram language model and a non-domain specific 4-gram language model using KenLM [28]. For the domain-specific language model, we collected domain-specific monolingual data from an e-commerce website, resulting in the number of 15M sentences. For the non-domain-specific language model, we use sampled LDC Chinese Gigaword (LDC Catalog No.: LDC2003T09) with 36M sentences. It should be noted that we train our language models on the word-level. In order to score a BPE-level corpus with such a language model, we score its words and copy this score for each of the subword units. After the word scores are computed, we then smooth them via a guassian distributed kernel with window size $L = 5$. We choose window size $L = 5$ considering that the language model is trained based on sequences of four words. We observed similar results with different window sizes, which is discussed in Section 4. Finally, we binarize the smoothed word scores into binary word weights by setting the threshold $T = 0.5$. The computed word weights are applied to the target side of out-of-domain sentences during the phase of continuing training. In order to get better translation results, we first trained the baseline model with mixed in-domain and out-of-domain data according to training objective 1, where no weights are used. We start our experiments by continuing training from this baseline model.

We implemented our NMT model using Tensorflow [29] library. The encoder is a bidirectional LSTM with size of 512 and the decoder is a LSTM with 2 layers of same size. All the weight parameters are initialized uniformly in $[-0.1, 0.1]$. We set dropout on RNN inputs with dropping probability 0.2. We train the networks with batch size 120 using SGD with initial learning rate 1.0 and gradually decaying to 0.1 after the initial 2 epochs.

3.2. Results

Statistics of the out-of-domain sentences/tokens selection after applying different types of weights are summarized in Table 2. Before the selection, the number of out-of-domain sentences is 5.15M and the number of tokens is 93.4M. When sentence-level weights are used, the sentences with $w_t = 0$ are ignored, resulting in the number of remaining sentences/tokens around 2.63M and 26.3M, respectively. When word-level weights are used, there are $1,279,927$ sentences where all word weights in the sentences are equal to zero. After removing these sentences, around 3.87M sentences are preserved and the number of selected tokens with word weights w_t=1 is around 36.6M. Given computed word weights, we alternatively choose only the chunk with the longest consecutive weights (LCW) where $w_t = 1$, resulting in chunk-level weights with the selected number of tokens further reduced to 25.8M.

We train a baseline NMT model on mixed in-domain and out-of-domain data with objective defined as Eq. 1 for 6 epochs. The data is mixed completely (mixed 0.53M in-domain e-commerce and 5.15M sampled out-of-domain sentence pairs) while training the baseline model. The baseline model initialized by a mix of in-domain/out-of-domain data can be regarded as a kind of "warm start". We have also tried training a baseline with out-of-domain data only and observed slightly worse result after fine-tuning on in-domain data (0.5 BLEU). Hence, we use the baseline model trained on a mix of in-domain/out-of-domain data in the following experiments. Given the baseline model, we then directly fine-tune on in-domain data for another 10 epochs or first continue training on the mixed data with sentence/chunk/word weights for 3 epochs and then fine-tune on in-domain data for 10 epochs. The model is saved after each epoch. We take the model which gives the best result on our development set for evaluation. Note that we always set word weights $w_t = 1$ for our in-domain dataset.

In Table 3, we show the effect of different types of weights on translation performance. First, the baseline trained on mixed in-domain and out-of-domain datasets gives 24.37% BLEU and 61.66% TER, respectively. Directly fine-tuning on in-domain dataset already improves the model due to the bias of the model towards in-domain data.

Continuing training on mixed datasets with previous objective defined in Eq. 1 shows insignificant changes in terms of BLEU and TER. However, introducing sentence-level weights improves the model from 24.37% to 25.79% BLEU and 61.66% to 60.82% TER, respectively. Compared to continuing training without weights, sentence-level weights are generated as described in Section 2.2, where w_t $\forall t$ are set to the same sentence weight $w \in \{0, 1\}$. We set the threshold equal to 0.5 and keep the sentences with weights above the threshold. The result from sentence-level feedback suggests that mining good out-of-domain sentences which are similar to in-domain datasets and dissimilar to out-of-domain datasets benefits model translation towards in-domain-like sentences even without fine-tuning on in-domain datasets.

The use of word-level weights improves the baseline model even better, from 24.37% to 26.14% BLEU and

Data set		e-commerce + out-of-domain	
Language		English	Chinese
Training	Sentences	5,689,989	
	Running words	97,266,344	96,480,106
	BPE vocabulary	33,484	45,867
Dev	Sentences	3173 (item descriptions)	
	Running words	51,130	48,900
Test	Sentences	739 (item descriptions)	
	Running words	19,034	18,262

Table 1: Corpus statistics for the e-commerce English→Chinese MT tasks.

Corpus	Sent. count	Token count
ood. sentences	5,153,191	93,427,867
+sent. weights	2,633,109	26,275,096
+word weights	3,873,264	36,617,395
+chunk weights	3,873,264	25,813,480

Table 2: Out-of-domain training corpus statistics. *ood. sentences* indicates the number of sentences/tokens in the out-of-domain corpus. *+sent. weights* indicates the number of selected out-of-domain sentences where the weights of the sentences are equal to 1. *+word weights* and *+chunk weights* indicate the statistics of selected out-of-domain sentences/tokens after applying word weights generation and LCW methods as described in Section 2.2 and 2.3.

61.66% to 60.34% TER, respectively. In this approach, the number of selected tokens is drastically reduced to 36.6M from 93.4M tokens, nearly 61% drop in number of tokens with improved translation performance. Word-level weights also outperform sentence-level weights by 0.35% in BLEU score and 0.48% in TER. It can be explained by the fact that each word in the sentences are given its own similarity to the in-domain datasets. Considering sentence-level weights set all words in a sentence with the same weight, even though part of the words in the sentences might not be related to the in-domain corpus, word-level weights are more accurate and effective.

Finally, chunk-level weights are generated from our word-level weights based on LCW. Here we aim to train the domain-adapted model from more consecutive segments rather than single selected words. On top of word-level weights, it improves by another 0.28% BLEU absolute and 0.24% TER, respectively. Out-of-domain sentences can be split into chunks which can be related to the in-domain and can be translated independently in terms of the context. The selection of the consecutive chunk with in-domain-like context can positively affect the training towards domain-adapted model. By focusing on in-domain related and out-of-domain unrelated part, word/chunk-level weights can effectively reduce the unnecessary noise in the out-of-domain training data. Compared to continuing training without word weights, we are able to further reduce the corpus by 72% tokens (25.8M vs. 93.4M selected tokens), resulting in an improvement of 2.11% BLEU absolute and 1.59% TER, respectively. It should also be noted that with similar number of tokens (25.8M vs. 26.3M), chunk-level weights outperforms sentence-level weights by 0.63% BLEU absolute and 0.72% TER.

Next, we further fine-tune the model with chunk-level weights and obtain further improvements of 0.88% BLEU absolute and 1.81% TER. Compared to directly fine-tuning on the baseline, continuing training the model with chunk-level weights and then fine-tuning improves translation results from 26.06% to 27.30% BLEU and 59.93% to 58.29% TER, respectively.

Results from the study on the effect of using different smoothing methods are shown in Table 4. The word weights generated without using smoothing methods, where $\hat{s}_t = s_t$, lead to poor translation quality of 21.38% from 24.37% BLEU and 66.25% from 61.66% TER, respectively. We need to smooth the word scores before thresholding because the values of $\log P_I(y_t|y_{t-n}^{t-1}) - \log P_O(y_t|y_{t-n}^{t-1})$ are noisy. If there are selected isolated words like ',' which have higher scores than the surrounding text, it may cause rare vocabulary problem after training.

The results from word weights computed from mean averaged filter and normal distributed filter are relatively close, 25.99% vs. 26.14% BLEU and 60.70% vs. 60.34% TER, respectively. These results are obtained via a filter with window size $L = 5$. In practice, we also tried setting window size $L = 3$ and $L = 7$, but didn't observe different results. We found that the surrounding word scores have to be considered for smoothing in order to make the word weights w_t less noisy as well as more precisely representing the similarity to the in-domain/out-of-domain.

Additionally, we also experimented with randomly selecting words in the out-of-domain sentences with binary mask. However, we observed a drop in the translation accuracy.

3.3. Examples

In Table 5, we show an example for which the system trained with word weights produces a better translation. The English sentence is "non-spill spout with patented valve". The

		Item descriptions	
No.	System description	BLEU [%]	TER [%]
1	Baseline	24.37	61.66
2	1 + continue training without word weights	24.31	61.69
3	1 + continue training with sentence weights	25.79	60.82
4	1 + continue training with word weights	26.14	60.34
5	1 + continue training with chunk weights	26.42	60.10
6	1 + fine-tuning on in-domain	26.06	59.93
7	5 + fine-tuning on in-domain	27.30	58.29

Table 3: E-commerce English → Chinese BLEU results on test set. *Baseline* is trained on mixed in-domain and out-of-domain data. *No. 2* is continuing training from baseline with objective defined as Eq. 1. *No. 3* is continuing training from baseline with sentence-level weights and *No. 4* is with word weights, as defined in Section 2.2. *No. 5* refers to assigning w_t using LCW method described in Section 2.3. *No. 6* is equivalent to directly fine-tuning on in-domain datasets starting from the baseline model and *No. 7* is equivalent to fine-tuning on in-domain datasets after *No. 5* is finished.

System	BLEU [%]	TER [%]
Baseline	24.37	61.66
+w.w. without smooth.	21.38	66.25
+w.w. (mean smooth.)	25.99	60.70
+w.w. (gauss. smooth.)	26.14	60.34

Table 4: Study on the effect of different smoothing methods for word weights generation. *Baseline* is the same as before. *w.w. without smoothing* means the word weights (w.w.) are computed without smoothing in the log domain. *w.w. (mean smooth.)* indicates smoothing the word scores via using a mean average filter before thresholding and *w.w. (gauss. smooth.)* indicates using a normal distributed filter before thresholding. The approaches regarding different smoothing methods are described in Section 2.2.

word "spout" is rare in our data, appearing in the out-of-domain training sentences only once. The Chinese side of this training example can be seen in Figure 1 together with the weights assigned to the individual words by our method. When smoothing is applied, isolated Chinese words such as "空气" ("air") are removed. With the longest consecutive words (LCW) method, the only remaining chunk is "防/溢出/喷口/内" ("inside the non-spills spout"), which is related to our in-domain data. The system with word weights is then trained only on this chunk on the target side, while the baseline model is trained on the entire sentence and generates inappropriate translations.

4. Discussions

The domain adaptation techniques (sentence-level/chunk-level/word-level) introduced in this paper are all derived from word weights generation. They aim to select out-of-domain sentences/chunks/words which are more related to in-domain corpus and unrelated to out-of-domain corpus. The word weights are computed prior to system tuning via the logarithm difference of LM probability scoring and are then used for tuning the sequence-to-sequence model. By measuring domain similarity with external criteria such as LM, this kind of out-of-domain data selection is able to highlight the in-domain-related and out-of-domain-unrelated parts and leads to less variation and errors in our e-commerce domain adaptation. In addition, the selected out-of-domain segments have to be smoothed in order to reduce noise.

5. Conclusions

In this work, we generate word-level weights by calculating the logarithm difference of the probability of two external language models for domain adaptation. This approach better selects the out-of-domain segments related to e-commerce domain, and requires fewer tokens for training. We experimented with continuing training models with sentence/chunk/word weights and show that they all give translation improvement in terms of BLEU and TER compared to continuing training without word weights. Experiments on our in-house English-Chinese datasets also show that continuing training with word weights then fine-tuning improves results over directly fine-tuning on baseline model.

In future, with the computed word weights as the initial parameters, we want to devise strategies to make online domain adaptation possible by dynamically updating word weights during training, which could in turn lead the in-domain data translation to better match its references.

6. References

[1] N. Ueffing and H. Ney, "Word-level confidence estimation for machine translation using phrase-based translation models," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, ser. HLT '05. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 763–770. [Online]. Available: https://doi.org/10.3115/1220575.1220671

System	Translation
Baseline	带专利阀的防溢出溅漏
+ word weights	带专利阀的防溢出喷口
Reference	带有专利阀门的防溢口

Table 5: Translations of ”non-spill spout with patented valve” produced by the baseline NMT system and the system trained with word weights. The last row shows the reference translation.

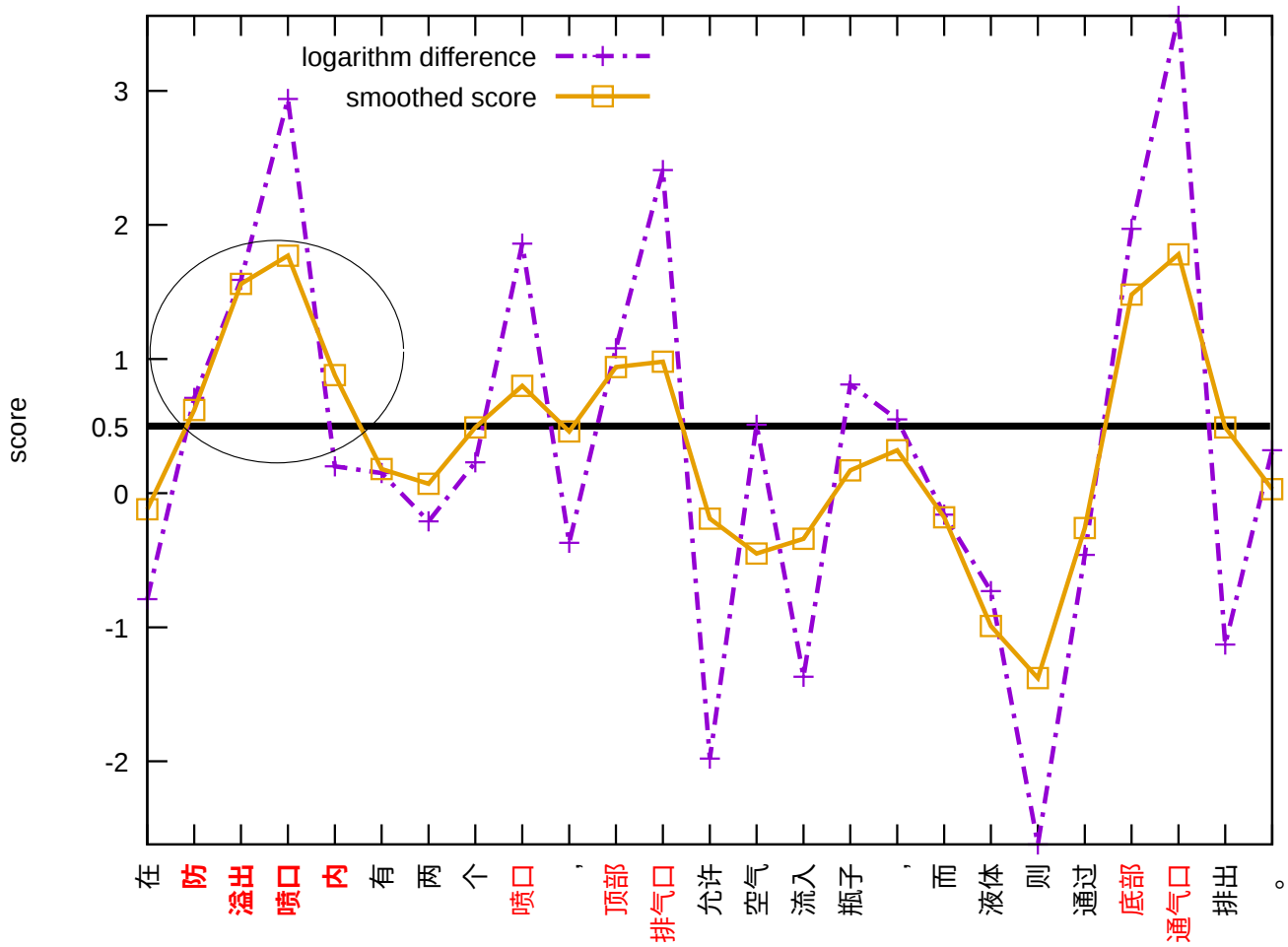


Figure 1: Our approach for word-based domain adaptation, with the target sentence on the bottom (the source sentence is ”Inside the non-spills spout there are two vents, where the top vent allows air to flow into the bottle while liquid smoothly pours out through the bottom vent.”). Above are displayed the word scores and the smoothed word scores. The black bold line indicates the threshold and the circle indicates which chunk is selected by LCW. After smoothing, the isolated random words are removed and the red words on the bottom are selected. The red bold words on the bottom are preserved after LCW.

[2] H. Schwenk, "Investigations on large-scale lightly-supervised training for statistical machine translation," in *IWSLT*, 2008.

[3] Y.-C. Tam, I. R. Lane, and T. Schultz, "Bilingual-lsa based lm adaptation for spoken language translation," in *ACL*, 2007.

[4] S. Hewavitharana, D. Mehay, S. Ananthakrishnan, and P. Natarajan, "Incremental topic-based translation model adaptation for conversational spoken language translation," in *ACL*, 2013.

[5] 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

[6] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 355–362. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145474

[7] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation," in *ACL*, 2013.

[8] N. D. H. S. S. J. A. A. S. Vogel, "Using joint models for domain adaptation in statistical machine translation," in *MT Summit*, 2015.

[9] S. Matsoukas, A.-V. I. Rosti, and B. Zhang, "Discriminative corpus weight estimation for machine translation," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, ser. EMNLP '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 708–717. [Online]. Available: http://dl.acm.org/citation.cfm?id=1699571.1699605

[10] G. Foster, C. Goutte, and R. Kuhn, "Discriminative instance weighting for domain adaptation in statistical machine translation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 451–459. [Online]. Available: http://dl.acm.org/citation.cfm?id=1870658.1870702

[11] 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 - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969173

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR*, 2015.

[13] 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 : December 3-4, 2015, Da Nang, Vietnam / Edited by Marcello Federico, Sebastian Stüker, Jan Niehues*. International Workshop on Spoken Language Translation, Da Nang (Vietnam), 3 Dec 2015 - 4 Dec 2015, Dec 2015.

[14] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[16] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *ACL*, 2016.

[17] M. Freitag and Y. Al-Onaizan, "Fast domain adaptation for neural machine translation," *CoRR*, vol. abs/1612.06897, 2016.

[18] C. Chu, R. Dabre, and S. Kurohashi, "An empirical comparison of domain adaptation methods for neural machine translation," in *ACL*, 2017.

[19] 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*. Association for Computational Linguistics, 2017, pp. 40–46. [Online]. Available: http://aclweb.org/anthology/W17-3205

[20] R. Wang, M. Utiyama, L. Liu, K. Chen, and E. Sumita, "Instance weighting for neural machine translation domain adaptation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 1482–1488. [Online]. Available: http://aclweb.org/anthology/D17-1155

[21] S. Khadivi, P. Wilken, L. Dahlmann, and E. Matusov, "Neural and statistical methods for leveraging meta-information in machine translation," in *MT Summit*, 2017.

[22] P. Petrushkov, S. Khadivi, and E. Matusov, "Learning from chunk-based feedback in neural machine translation," in *ACL*, 2018.

[23] P.-C. Chang, M. Galley, and C. D. Manning, "Optimizing chinese word segmentation for machine translation performance," in *Proceedings of the Third Workshop on Statistical Machine Translation*, ser. StatMT '08. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 224–232. [Online]. Available: http://dl.acm.org/citation.cfm?id=1626394.1626430

[24] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 177–180. [Online]. Available: http://dl.acm.org/citation.cfm?id=1557769.1557821

[25] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *CoRR*, vol. abs/1508.07909, 2016.

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

[27] M. Snover, B. J. Dorr, R. F. Schwartz, and L. Micciulla, "A study of translation edit rate with targeted human annotation," 2006.

[28] K. Heafield, "Kenlm: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, ser. WMT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 187–197. [Online]. Available: http://dl.acm.org/citation.cfm?id=2132960.2132986

[29] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'16. Berkeley, CA, USA: USENIX Association, 2016, pp. 265–283. [Online]. Available: http://dl.acm.org/citation.cfm?id=3026877.3026899

A Machine Translation Approach for Modernizing Historical Documents Using Backtranslation

Miguel Domingo, Francisco Casacuberta

PRHLT Research Center
Universitat Politècnica de València
midobal@prhlt.upv.es, fcn@prhlt.upv.es

Abstract

Human language evolves with the passage of time. This makes historical documents to be hard to comprehend by contemporary people and, thus, limits their accessibility to scholars specialized in the time period in which a certain document was written. Modernization aims at breaking this language barrier and increase the accessibility of historical documents to a broader audience. To do so, it generates a new version of a historical document, written in the modern version of the document's original language. In this work, we propose several machine translation approaches for modernizing historical documents. We tested these approaches in different scenarios, obtaining very encouraging results.

1. Introduction

Historical documents are an important part of our cultural heritage. With the aim of their preservation, there is an increase need of digitalazing—creating a digital text version which can be searched and automatically processed— of historical documents [1]. However, there is an additional difficulty created by their linguistic properties: On the one hand, human language evolves with the passage of time. On the other hand, due to a lack of a spelling convention, orthography changes depending on the author and time period in which a given document was written. These problems make historical documents harder to read, and makes it harder to digitalize them (since their digital text version needs to be searched and automatically processed).

The orthography problem has been well researched in the literature [2, 3, 4, 5, 6, 7]. The proposed solution that aims to solve this problem is known as spelling normalization, and its goal is to adapt the document's spelling to modern standards in order to achieve an orthography consistency and increase the document's readability. However, while is true that spelling normalization makes historical documents easier to read, they are still hard to comprehend by contemporary people. This problem limits the accessibility of historical documents to scholars specialized in the time period in which the document was written.

Modernization aims at breaking the language barrier, generating a new version of a historical document in the modern version of the language in which the document was originally written (see Fig. 1 for an example). Therefore, not only the orthography is updated. The lexicon and grammar are also modified in order to match the modern use of the document's language. The main drawback of this solution is that part of the document's original intention could be lost in the process (e.g., part of the rhyme in Fig. 1 is lost for the sake of clarity). Nonetheless, the document's clarity is increased and, thus, its accessibility to a broader audience.

To the best of our knowledge, modernization of historical documents has been less researched in the literature. A shared task was organized in order to translate historical text to contemporary language [9]. The shared task's main goal was spelling normalization. However, they also tackle modernization using a set of rules. Finally, there was an approach to modernize historical documents using Statistical Machine Translation (SMT) [10]. In this work, we tackle modernization by using an SMT and Neural Machine Translation (NMT) approach. Additionally, since a frequent problem when working with historical documents is the scarce availability of parallel training data [5], we created two small parallel corpora (see Section 3.1) and generated synthetic data using backtranslation [11]. Our main contributions are the followings:

- First use, to the best of our knowledge, of NMT and backtranslation for historical documents mod-

Shall I compare thee to a summer's day?	Shall I compare you to a summer day?
Thou art more lovely and more temperate:	You're lovelier and milder.
Rough winds do shake the darling buds of May,	Rough winds shake the pretty buds of May,
And summer's lease hath all too short a date:	and summer doesn't last nearly long enough.
Sometime too hot the eye of heaven shines,	Sometimes the sun shines too hot,
And often is his gold complexion dimm'd;	and often its golden face is darkened by clouds.
And every fair from fair sometime declines,	And everything beautiful stops being beautiful,
By chance or nature's changing course untrimm'd;	either by accident or simply in the course of nature.
But thy eternal summer shall not fade	But your eternal summer will never fade,
Nor lose possession of that fair thou ow'st;	nor will you lose possession of your beauty,
Nor shall Death brag thou wander'st in his shade,	nor shall death brag that you are wandering in the underworld,
When in eternal lines to time thou grow'st;	once you're captured in my eternal verses.
So long as men can breathe or eyes can see,	As long as men are alive and have eyes with which to see,
So long lives this, and this gives life to thee.	this poem will live and keep you alive.

Figure 1: Example of modernizing a historical document. The original text is *Shakespeare Sonnet 18*. The modernized version of the Sonnet was obtained from [8].

ernization.

- Comparison of approaches based on SMT and NMT.
- Experimented with three historical corpora—two of which were created for this work—from three different time periods and two different languages.

The rest of this document is structure as follows: Section 2 introduces the different Machine Translation (MT) approaches used in our work. Then, Section 3 describes the experiments conducted in order to assess our proposal. After that, Section 4 presents and discusses the results of those experiments. Finally, in Section 5, conclusions are drawn.

2. Machine Translation

In this section, we present the machine translation approaches used in our work.

2.1. Statistical Machine Translation

Given a source sentence $\mathbf{x}$, SMT aims at finding its best translation $\hat{\mathbf{y}}$ [12]:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} Pr(\mathbf{y} \mid \mathbf{x}) \quad (1)$$

For years, the prevailing approach to compute this expression have been phrase-based models [13]. These models rely on a log-linear combination of different models [14]: namely, phrase-based alignment models, reordering models and language models; among others [15, 16]. However, more recently, this approach has shifted into neural models (see Section 2.2).

2.2. Neural Machine Translation

NMT is the neural approach to compute Eq. (1). Frequently, it relies on a Recurrent Neural Network (RNN) encoder-decoder framework. In this framework, the source sentence is projected into a distributed representation at the encoding step. At the decoding step, the decoder generates its translation word by word [17].

The system's input is a sequence of words in the source language. Each source word is linearly projected to a fixed-sized real-valued vector through an embedding matrix. These word embeddings are feed into a bidirectional [18] Long Short-Term Memory (LSTM) [19] network, resulting in a sequence of annotations produced by concatenating the hidden states from the forward and backward layers.

The model features an attention mechanism [20], which allows the decoder to focus on parts of the input sequence, computing a weighted mean of annotations sequence. A soft alignment model computes these weights by weighting each annotation with the previous decoding state.

The decoder is another LSTM network, conditioned by the representation computed by the attention model and the last word generated. Finally, a deep output layer [21] computes a distribution over the target language vocabulary.

The model is trained by means of stochastic gradient descend, applied jointly to maximize the log-likelihood over a bilingual parallel corpus. At decoding time, the model approximates the most likely target sentence with beam-search [17].

2.3. Backtranslation

Backtranslation [11] has become the norm when building state-of-the-art NMT systems, especially in resource-poor scenarios [22]. It is a useful technique to increase the training data by creating synthetic text from monolingual data. Given a monolingual corpus in the target language, and an MT system trained to translate from the target language to the source language, the synthetic data is generated by translating the monolingual corpus with the MT system. After that, the synthetic data is used as the source part of the corpus, and the monolingual data as the target part. Finally, this new corpus is mixed with the available training data in order to train a new MT system.

In this work, to generate the synthetic data, we translate the monolingual data using an ad-hoc SMT system trained with the corpus' training partition. Additionally, since the datasets are considerable small, prior to mixing the synthetic corpus with the training partition, we replicate several times the training data in order to match the size of the synthetic data and avoid overfitting [23]. Finally, we trained an NMT system with this new corpus.

3. Experimental Framework

In this section, we present the corpora and metrics, and describe the MT systems used during the experimental session.

3.1. Corpora

The first corpus used to assess our proposal was the **Dutch Bible** [9]. This corpus consists in a collection of different versions of the Dutch Bible. More precisely, a version from 1637, another from 1657, another from 1888 and another from 2010. All versions contain the same text except for the 2010 version, which is missing the last books. Moreover, the authors mentioned that the translation from this last version is not very reliable. Additionally, due to Dutch not evolving significantly during this period, 1637 and 1657 versions are fairly similar. For this reason, we decided to only use the 1637 version—considering this as the original document—and the 1888 version—considering 19^{th} century Dutch as *modern Dutch*.

To create the synthetic corpus (see Section 2.3), we collected all 19^{th} century Dutch books available at the *Digitale Bibliotheek voor de Nederlandse letteren*[1] and used them as monolingual data.

The second corpus we used was **El Quijote**. We built this corpus using a version [24] of the original 17^{th} century Spanish novel by Miguel de Cervantes, and a 21^{st} century version modernized by Andrés Trapiello [25]. The first step was to split each document into sentences. Since the 17^{th} century version was faithful to the original manuscript (in which each document line is formed by a very few words), we replaced line breaks by spaces to create a single sentence, and removed empty lines. For consistency, we did the same to the 21^{st} century version. After that, we split each document into sentences by adding line breaks to relevant punctuation (i.e., dots, quotation marks, admiration marks, etc). Then, to ensure consistency, we checked special symbols (e.g., quotation marks) and made sure that the same character was used in both versions. Finally, in order to create a parallel corpus, we aligned both documents using `Hunalign` [26].

Since the content of this corpus was a novel, we decided that, to create the synthetic corpus, it would be best to use monolingual data coming from Spanish literature. For this reason and, considering that Spanish hasn't changed significantly over the last decades, we decided to collect free-of-right late 20^{th} century Spanish novels from *Project Gutenberg*[2].

Finally, as a third corpus, we selected **El Conde Lucanor**. We built this corpus using a version of the original 14^{th} century Spanish novel by Don Juan Manuel, and a 21^{st} century version modernized by Luis López Nieves [27]. To create the parallel version, we followed the same steps than with *El Quijote*. However, unlike with *El Quijote*, the resulting corpus was too small to be able to use for training an MT system. Therefore, we decided to use it only as a test. Unable to find a suitable training corpus, we decided to test *El Conde Lucanor* using the systems created for *El Quijote*—despite the fact that the original documents were written three centuries apart from one another.

Table 1 presents the corpora statistics.

3.2. Metrics

In order to asses our proposal, we made use of the following well-known metrics:

- **BiLingual Evaluation Understudy (BLEU)** [28]: computes the geometric average of the modified

[1] `http://dbnl.nl/`

[2] `https://www.gutenberg.org/`

		Dutch Bible	El Quijote	El Conde Lucanor
Train	$\|S\|$	35.2K	10K	-
	$\|T\|$	870.4/862.4K	283.3/283.2K	-
	$\|V\|$	53.8/42.8K	31.7/31.3K	-
Development	$\|S\|$	2000	2000	-
	$\|T\|$	56.4/54.8K	53.2/53.2K	-
	$\|V\|$	9.1/7.8K	10.7/10.6K	-
Test	$\|S\|$	5000	2000	2252
	$\|T\|$	145.8/140.8K	41.8/42.0K	62.0/56.7K
	$\|V\|$	10.5/9.0K	8.9/9.0K	7.4/8.6K
Monolingual	$\|S\|$	4.1M	567.0K	-
	$\|T\|$	88.3M	9.5M	-
	$\|V\|$	2.0M	470.4K	-

Table 1: Corpora statistics. $|S|$ stands for number of sentences, $|T|$ for number of tokens and $|V|$ for size of the vocabulary. *Monolingual* refers to the monolingual data used to create the synthetic data. M denotes million and K thousand.

n-gram precision, multiplied by a brevity factor that penalizes short sentences.

- **Translation Error Rate (TER)** [29]: computes the number of word edit operations (insertion, substitution, deletion and swapping), normalized by the number of words in the final translation.

Confidence intervals ($p = 0.05$) were computed for all metrics by means of bootstrap resampling [30].

3.3. MT Systems

We trained the SMT systems with `Moses` [31], following the standard procedure: we estimated a 5-gram language model—smoothed with the improved KneserNey method—using `SRILM` [32], and optimized optimized the weights of the log-lineal model with MERT [33]. Additionally, we lowercased and tokenized the corpora using the standard scripts and, later, truecased the translated text using `Moses`' truecaser.

To train the NMT systems, we used `OpenNMT` [34]. We used LSTM units, following the findings from [35]. We set the size of the word embedding and LSTM units to 1024. We used *Adam* [36] with a learning rate of 0.0002 [37]. The beam size was set to 6. Finally, the corpora were lowercased and tokenized—and, later, truecased and detokenized—using `OpenNMT`'s tools.

In order to reduce the vocabulary, we applied Byte Pair Encoding (BPE) [38] to both SMT and NMT systems. We trained the models with a joint vocabulary of 32000 BPE units.

4. Results

In this section, we present and discuss the results of the experiments conducted in order to assess our proposal. Table 2 presents the experimental results.

Dutch Bible contained an additional baseline which consisted in generating a modernized version of the text by applying a set of rules to the original document [9]. This second baseline improved significantly (close to 40 BLEU points and 30 TER points) the standard baseline of considering the original document as the modernized version. However, the SMT approach improved those results even more (near 30 BLEU points and 15 TER points of improvement with respect to the second baseline, and 70 BLEU points and 50 TER points with respect to the standard baseline). The NMT approach yielded better results than the standard baseline (and improvement of around 25 BLEU points and 5 TER points), but worse results than the second baseline and the SMT approach. Most likely, this is due to the training corpus being too small, which is a well-known problem in NMT. Finally, the backtranslation approach yielded the worst results. These results represent an improvement over the standard baseline in term of BLEU (around 4 points), and a deterioration in terms of TER (around 8 points). Most likely, this is due to the monolingual data used for backtranslation not being similar enough to the training data.

The experiments using *El Quijote* behaved similarly—taking into account that the only available baseline is the standard one—to *Dutch Bible*: The SMT approached yielded the best results (an improvement of

System	Dutch Bible		El Quijote		El Conde Lucanor	
	BLEU	TER	BLEU	TER	BLEU	TER
Baseline	13.5 ± 0.3	57.0 ± 0.3	36.5 ± 0.8	43.3 ± 1.1	5.8 ± 0.3	89.6 ± 1.0
Baseline_2	50.8 ± 0.4	26.5 ± 0.3	-	-	-	-
SMT	$\mathbf{80.1 \pm 0.5}$	$\mathbf{9.9 \pm 0.3}$	$\mathbf{58.9 \pm 1.0}$	$\mathbf{29.4 \pm 1.2}$	$\mathbf{8.4 \pm 0.3}$	$\mathbf{83.8 \pm 1.0}$
NMT	38.0 ± 0.6	51.7 ± 2.2	37.4 ± 1.2	51.5 ± 2.0	2.7 ± 0.2	99.5 ± 2.0
$\text{NMT}_{\text{Synthetic}}$	17.4 ± 0.5	65.6 ± 1.7	45.2 ± 1.3	50.6 ± 3.5	3.1 ± 0.2	165.1 ± 8.2

Table 2: Experimental results. *Baseline* system corresponds to considering the original document as the modernized version. *Baseline*$_2$ came with the *Dutch Bible* and is a modernized version of the text generated by applying a set of rules to the original document [9]. *SMT* and *NMT* are the SMT and NMT approaches respectively. $\text{NMT}_{\text{Synthetic}}$ is the NMT system trained with the synthetic data generated through backtranslation. Best results are denoted in **bold**.

close to 22 BLEU points and 14 TER points). The results yielded by the NMT approach were not significantly different to the baseline in terms of BLEU, and represented close to a 10 points deterioration in terms of TER. In this case, however, the backtranslation approach yielded nearly a 10 points improvement in terms of BLEU, and the same TER results as the NMT approach.

Not being able to obtain enough suitable training data for *El Conde Lucanor*, we used the same systems than for *El Quijote*. However, these documents were written three centuries apart from one another (*El Conde Lucanor* is written in 14^{th} century Spanish and *El Quijote* in 17^{th} century Spanish). Therefore, the obtained results contained a low translation quality. Nonetheless, it is worth noting that the SMT approach yielded improvements over the baseline (around 3 BLEU points and 6 TER points). However, the NMT and backtranslating approached yielded a deterioration of 3 BLEU points (in both cases) and 10 and 75 TER points respectively.

In general, SMT yielded the best results in all cases. NMT was able to improve *Dutch Bible*'s baseline, yielding similar results to *El Quijote*'s baseline and worse results than *El Conde Lucanor*'s baseline. Finally, despite being successfully used in resources-poor scenarios, backtranslation was only able to improve results for the experiment using *El Quijote*, and these results were worse than the ones yielded by the SMT approach.

4.1. Qualitative Analysis

Table 2 shows some examples of sentences modernized using the different MT approaches.

The first example is a sentence from *El Quijote*. The hypothesis generated by the SMT approach is very closed to the reference. The main differences are a change in the order of actions (the original sentence says *Y dejando de comer, se levantó*, which is changed by the hypothesis into *Y levantándose, dejó de comer*) and some changes in the conjugation of verbs (e.g., *dejando* is changed into *dejó*). However, the main goal of modernization is not to generate a perfect equivalent version, but to make the document easier to comprehend—making the overall meaning more important than the exact choice of words. While sentences like these are penalized by the automatic metrics, they accomplish modernization's goal.

The hypothesis generated by the NMT approach follows the same structure than the SMT hypothesis (it makes the same reordering and conjugation changes) but contains non-existent words (e.g., *ancen*) and has some errors (e.g, *a los pies* in stead of *puesto a caballo*). Therefore, some parts are easier to comprehend than in the original version, but the meaning of the sentence is not clear.

Finally, the hypothesis generated by the backpropagation approach is almost the same as the one generated by the SMT approach (the only change is *fierded* in stead of *fiereza*). While this hypothesis is less correct than the SMT one, they are both equally easy to comprehend.

The second example is from *El Conde Lucanor*, whose experiments were conducted using the systems trained with *El Quijote*. As a result, all the hypothesis are hard to comprehend. While the automatic metrics heavily penalize the backtranslation approach, in this case, is the one which is closer to modern Spanish. Moreover, it is the only hypothesis which preserves the name of the main characters (*Lucanor* and *Patronio*). Looking through the whole texts, the SMT and NMT hypothesis frequently changed the characters named into non-existent words, while the backtranslation approach rarely modified them. Finally, having trained the systems with *El Quijote* has a visible effect in the SMT and

	El Quijote
Original:	Y, leuantandose, dexó de comer, y fue a quitar la cubierta de la primera imagen, que mostro ser la de San Iorge puesto a cauallo, con vna serpiente enroscada a los pies, y la lança atrauessada por la boca, con la fiereça que suele pintarse.
Modernized:	Y dejando de comer, se levantó y fue a quitar la cubierta de la primera imagen, que resultó ser la de san Jorge a caballo, con una serpiente enroscada a los pies y la lanza atravesándole la boca, con la fiereza que suele pintarse.
SMT:	Y levantándose, dejó de comer, y fue a quitar la cubierta de la primera imagen, que mostró ser la de San Jorge puesto a caballo, con una serpiente enroscada a los pies, y la lanza atravesada por la boca, con la fiereza que suele pintarse.
NMT:	Y levantándose, dejó de comer, y fue a quitar la cubierta de la primera ancen, que mostró ser la de San Marorge a los pies y la lanza ahabesada por la boca;
NMT$_{\text{Synthetic}}$:	Y levantándose, dejó de comer, y fue a quitar la cubierta de la primera imagen, que mostró ser la de San Jorge puesto a caballo, con una serpiente enroscada a los pies, y la lanza atravesada por la boca, con la fierded que suele pintarse.
	El Conde Lucanor
Original:	-Señor conde Lucanor -dixo Patronio-, vien entiendo que el mío consejo non vos faze grant mengua, pero vuestra voluntad es que vos diga lo que en esto entiendo, et vos conseje sobre ello, fazerlo he luego.
Modernized:	-Señor Conde Lucanor -dijo Patronio-, bien sé que mi consejo no os hace mucha falta, pero, como confiáis en mí,
SMT:	-– Señor conde Lucanor -dijo Patroniorosa, vien entiendo que el mío consejo non vos face grant mengua, pero vuestra voluntad es que vos diga lo que en esto entiendo, et vos aconseje en ello, ferlo he luego .
NMT:	Señor conde Olcanor dijo dijo Pacasos –dijo en entiendo que el mío consejo non os fazo felimengua y vuestra merced es que vos diga lo que en esto entiendo.
NMT$_{\text{Synthetic}}$:	-Señor conde Lucanor -dijo Patronio, vien entiendo que el mío consejo non es face grant mengua, pero vuestra voluntad es que vos diga lo que en esto entiendo, et vos conseje sobre ello, también yo he dicho.

Table 3: Examples of modernizing a sentence using the different MT approaches. *SMT* and *NMT* are the SMT and NMT approaches respectively. *NMT*$_{\text{Synthetic}}$ is the NMT system trained with the synthetic data generated through backtranslation.

NMT approaches (with verbs conjugations such as *ferlo*, or expression such as *vuestra merced*). This effect, however, is not so visible in the backtranslation hypothesis. All in all, neither hypothesis accomplishes the goal of improving the comprehension of the original sentence. Nonetheless, it is worth noting that this was a tricky example since the original sentence is modernized into a much shorter sentence (which reflects that 14^{th} century Spanish used longer expressions than modern Spanish).

5. Conclusions and Future Work

In this work we proposed several machine translation approaches to modernize historical documents in order to break the language barrier and increase their accessibility to a broader audience.

We tested our approaches using three historical datasets (two of which were created for this work) from three different time periods and two different languages.

Our first approach was based in SMT and yielded, for all cases, the best results. With the exception of the dataset for which there were not available any suitable training data, this approach yielded significant improvements of around 22 to 67 BLEU points and 14 to 48 TER points.

Since the available training data was fairly small, the approach based on NMT produced less satisfactory results. While it was able to yield improvements for one dataset, the rest of the experiment resulted in either not significantly different than the baseline, or yielding a deterioration in terms of translation quality.

Finally, despite being successfully used in resources-poor scenarios, backtranslation was only able to improve results for one dataset, and only in terms of BLEU. Our best hypothesis is that historical documents are very language-specific and, therefore, choosing the monolingual corpus to use for creating the synthetic data is extremely important. While we tried to create the monolingual datasets using similar topics, the corpora's topics were too specific: religious texts, a cavalry novel and medieval tales.

In a future work, we would like to research the relation between the domains of the monolingual and training corpora deeper. Additionally, we want to explore the use of data selection techniques for constructing the monolingual corpus to use for backtranslation, and to create a training partition for cases in which we do not have suitable training data available (as was the case with *El Conde Lucanor*).

6. Acknowledgments

The research leading to these results has received funding from the Ministerio de Economía y Competitividad (MINECO) under project CoMUN-HaT (grant agreement TIN2015-70924-C2-1-R). We gratefully acknowledge the support of NVIDIA Corporation with the donation of a GPU used for part of this research, and Andrés Trapiello and Ediciones Destino for granting us permission to use their book in our research.

7. References

[1] M. Piotrowski, *Natural Language Processing for Historical Texts*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool, 2012, no. 17.

[2] A. Baron and P. Rayson, "VARD2: A tool for dealing with spelling variation in historical corpora," *Postgraduate conference in corpus linguistics*, 2008.

[3] J. Porta, J.-L. Sancho, and J. Gómez, "Edit transducers for spelling variation in old spanish," in *Proceedings of the workshop on computational historical linguistics*, 2013, pp. 70–79.

[4] Y. Scherrer and T. Erjavec, "Modernizing historical slovene words with character-based smt," in *Proceedings of the Biennial International Workshop on Balto-Slavic Natural Language Processing*, 2013, pp. 58–62.

[5] M. Bollmann and A. Søgaard, "Improving historical spelling normalization with bi-directional lstms and multi-task learning," in *Proceedings of the International Conference on the Computational Linguistics*, 2016, pp. 131–139.

[6] N. Ljubešic, K. Zupan, D. Fišer, and T. Erjavec, "Normalising slovene data: historical texts vs. user-generated content," in *Proceedings of the Conference on Natural Language Processing*, 2016, pp. 146–155.

[7] M. Domingo and F. Casacuberta, "Spelling normalization of historical documents by using a machine translation approach," in *Proceedings of the Annual Conference of the European Association for Machine Translation*, 2018, pp. 129–137.

[8] J. Crowther, *No Fear Shakespeare: Sonnets*. SparkNotes, 2004.

[9] E. Tjong Kim Sang, M. Bollmann, R. Boschker, F. Casacuberta, F. Dietz, S. Dipper, M. Domingo, R. van der Goot, M. van Koppen, N. Ljubešić, R. Östling, F. Petran, E. Pettersson, Y. Scherrer, M. Schraagen, L. Sevens, J. Tiedemann, T. Vanallemeersch, and K. Zervanou, "The CLIN27 shared task: Translating historical text to contemporary language for improving automatic linguistic annotation," *Computational Linguistics in the Netherlands Journal*, vol. 7, pp. 53–64, 2017.

[10] M. Domingo, M. Chinea-Rios, and F. Casacuberta, "Historical documents modernization," *The Prague Bulletin of Mathematical Linguistics*, vol. 108, pp. 295–306, 2017.

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

[12] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[13] P. Koehn, *Statistical Machine Translation*. Cambridge University Press, 2010.

[14] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 295–302.

[15] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *Proceedings of the Annual German Conference on Advances in Artificial Intelligence*, vol. 2479, 2002, pp. 18–32.

[16] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 2003, pp. 48–54.

[17] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," pp. 3104–3112, 2014.

[18] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] 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 (arXiv:1409.0473)*, 2015.

[21] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *arXiv preprint arXiv:1312.6026*, 2013.

[22] A. Poncelas, D. Shterionov, A. Way, G. Maillette de Buy Wenniger, and P. Passban, "Investigation backtranslation in neural machine translation," in *Proceedings of the Annual Conference of the European Association for Machine Translation*, 2018, pp. 249–258.

[23] R. Chatterjee, M. A. Farajian, M. Negri, M. Turchi, A. Srivastava, and S. Pal, "Multi-source neural automatic post-editing: Fbk's participation in the wmt 2017 ape shared task," in *Proceedings of the Second Conference on Machine Translation*, 2017, pp. 630–638.

[24] F. F. Jehle, *Works of Miguel de Cervantes in Old- and Modern-spelling*. Indiana University Purdue University Fort Wayne, 2001.

[25] A. Trapiello, *Don Quijote de la Mancha Puesto en castellano actual íntegra y fielmente por Andrés Trapiello*. Ediciones Destino, 2015.

[26] D. Varga, L. Németh, P. Halácsy, A. Kornai, V. Trón, and V. Nagy, "Parallel corpora for medium density languages," in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, 2005, pp. 590–596.

[27] L. López Nieves, *El Conde Lucanor*. Biblioteca Digital Ciudad Seva, 2002.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[29] 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 Association for Machine Translation in the Americas*, 2006, pp. 223–231.

[30] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 388–395.

[31] 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 Annual Meeting of the Association for Computational Linguistics*, 2007, pp. 177–180.

[32] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of the International Conference on Spoken Language Processing*, 2002, pp. 257–286.

[33] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 160–167.

[34] 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.

[35] D. Britz, A. Goldie, T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures," *arXiv preprint arXiv:1703.03906*, 2017.

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

[37] 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.

[38] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 1715–1725.

Multi-Source Neural Machine Translation with Data Augmentation

Yuta Nishimura[1] , *Katsuhito Sudoh*[1] , *Graham Neubig*[2,1], *Satoshi Nakamura*[1]

[1]Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
[2]Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

{nishimura.yuta.nn9, sudoh, s-nakamura}@is.naist.jp

gneubig@cs.cmu.edu

Abstract

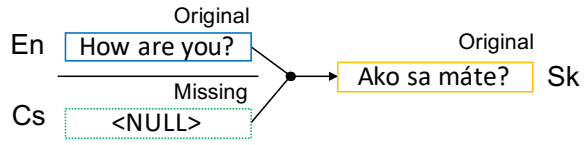
Multi-source translation systems translate from multiple languages to a single target language. By using information from these multiple sources, these systems achieve large gains in accuracy. To train these systems, it is necessary to have corpora with parallel text in multiple sources and the target language. However, these corpora are rarely complete in practice due to the difficulty of providing human translations in *all* of the relevant languages. In this paper, we propose a data augmentation approach to fill such incomplete parts using multi-source neural machine translation (NMT). In our experiments, results varied over different language combinations but significant gains were observed when using a source language similar to the target language.

1. Introduction

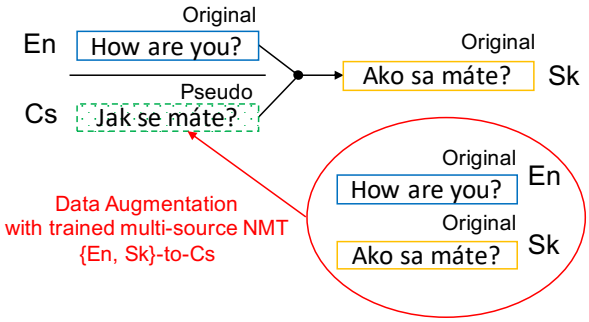
Machine Translation (MT) systems usually translate one source language to one target language. However, in many real situations, there are multiple languages in the corpus of interest. Examples of this situation include the multilingual official document collections of the European parliament [1] and the United Nations [2]. These documents are manually translated into all official languages of the respective organizations. Many methods have been proposed to use these multiple languages in translation systems to improve the translation accuracy [3, 4, 5, 6]. In almost all cases, multilingual machine translation systems output better translations than one-to-one systems, as the system has access to multiple sources of information to reduce ambiguity in the target sentence structure or word choice.

However, in contrast to the more official document collections mentioned above where it is mandated that all translations in all languages, there are also more informal multilingual captions such as those of talks [7] and movies [8]. Because these are based on voluntary translation efforts, large portions of them are not translated, especially into languages with a relatively small number of speakers.

Nishimura *et al.* [9] have recently proposed a method for multi-source NMT that is able to deal with the case of missing source data encountered in these corpora. The implementation is simple: missing source translations are replaced with a special symbol ⟨NULL⟩ as shown in Figure 1(a). This



(a) Multi-source NMT with filling in a symbol [9]



(b) Proposed Method: Multi-source NMT with data augmentation

Figure 1: Example of multi-source NMT with an incomplete corpus; The language pair is {English, Czech}-to-Slovak and the translation of Czech is missing.

method allows us to use incomplete corpora both at training time and test time, and multi-source NMT with this method was shown to achieve higher translation accuracy. If the model is trained on corpora with a large number of ⟨NULL⟩ symbols on the source side, a large number of training examples will be different from test time, when we actually have multiple sources. Thus, these examples will presumably be less useful in training a model intended to do multi-source translation. In this paper, we propose an improved method for utilizing multi-source examples with missing data: using a pseudo-corpus whose missing translations are filled up with machine translation outputs using a trained multi-source NMT system as shown in Figure 1(b). Experimental results show that the proposed method is a more effective method to incorporate incomplete multilingual corpora, achieving improvements of up to about 2 BLEU over the previous method where each missing input sentence is replaced by ⟨NULL⟩.

2. Related Work

2.1. Multi-source NMT

There are two major approaches to multi-source NMT; multi-encoder NMT [10] and mixture of NMT Experts [11]. In this work, we focus on the multi-encoder NMT that showed better performance in most cases in Nishimura *et al.* [9].

Multi-encoder NMT [10] is similar to the standard attentional NMT framework [12] but uses multiple encoders corresponding to the source languages and a single decoder.

Suppose we have two LSTM-based encoders and their hidden and cell states at the end of the inputs are h_1, h_2 and c_1, c_2, respectively. Multi-encoder NMT initializes its decoder hidden state h and cell state c using these encoder states as follows:

$$h = \tanh(W_c[h_1; h_2]) \tag{1}$$

$$c = c_1 + c_2 \tag{2}$$

Attention is then defined over encoder states at each time step t and resulting context vectors d_t^1 and d_t^2 are concatenated together with the corresponding decoder hidden state h_t to calculate the final context vector $\tilde{h_t}$.

$$\tilde{h_t} = \tanh(W_c[h_t; d_t^1; d_t^2]) \tag{3}$$

Our multi-encoder NMT implementation is basically similar to the original one [10] but has a difference in its attention mechanism. We use global attention used in Nishimura *et al.* [9], while Zoph and Knight used local-p attention. The global attention allows the decoder to look at everywhere in the input, while the local-p attention forces to focus on a part of the input [13].

2.2. Data Augmentation for NMT

Sennrich *et al.* proposed a method to use monolingual training data in the target language for training NMT systems, with no changes to the network architecture [14]. It first trains a seed target-to-source NMT model using a parallel corpus and then translates the monolingual target language sentences into the source language to create a *synthetic* parallel corpus. It finally trains a source-to-target NMT model using the seed and synthetic parallel corpora. This very simple method called *back-translation* makes effective use of available resources, and achieves substantial gains. Imamura *et al.* proposed a method that enhances the encoder and attention using target monolingual corpora by generating mutliple source sentences via sampling as an extension of the back-translation [15].

There are also other approaches for data augmentation other than back-translation. Wang *et al.* proposed a method of randomly replacing words in both the source sentence and the target sentence with other random words from their corresponding vocabularies [16]. Kim and Rush proposed a sequence-level knowledge distillation in NMT that uses machine translation results by a large teacher model to train a small student model as well as ground-truth translations [17].

Our work is an extension of the back-translation approach in multilingual situations by generating pseudo-translations using multi-source NMT.

3. Proposed Method

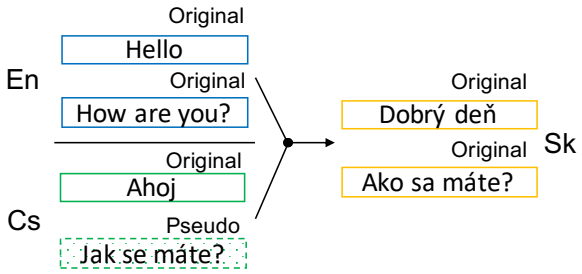
We propose three types of data augmentation for multi-encoder NMT; "fill-in", "fill-in and replace" and "fill-in and add." Firstly, we explain about the data requirements and overall framework using Figure 1(b). We used three languages; English, Czech and Slovak. Our goal is to get the Slovak translation, and to do so we take three steps. There are not any missing data in English translations, but Slovak and Czech translations have some missing data. In the first step, we train a multi-encoder NMT model (Source: English and Slovak, Target: Czech) to get Czech pseudo-translations using the baseline method, which is to replace a missing input sentence with a special symbol $\langle$NULL$\rangle$. In the second step, we create Czech pseudo-translations using multi-encoder NMT which was trained on the first step. We conducted three types of augmentation, which we introduce later. Finally in the third step, we switch the role of Czech and Slovak, in other words, we train a new multi-encoder NMT model (Source: English and Czech, Target: Slovak). At this time, we use Slovak pseudo-translations in the source language side. This method is similar to back-translation but taking advantage of the fact that we have an additional source of knowledge (Czech or Slovak) when trying to augment the other language (Slovak or Czech respectively).

We next introduce three types of augmentation. Figure 2 illustrates their examples in {English, Czech}-to-{Slovak} case where one Czech sentence is missing.

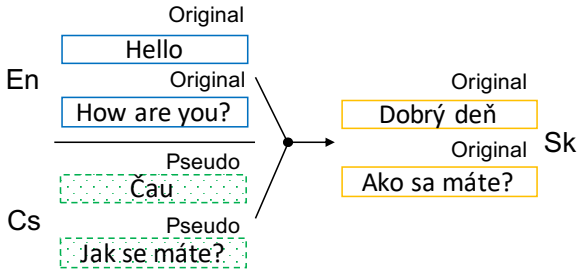
(a) **fill-in:** where only missing parts in the corpus are filled up with pseudo-translations.

(b) **fill-in and replace:** where we both augment the missing part and replace original translations with pseudo-translations in the source language except English whose translations has not any missing data. The motivation behind this method is not to use unreliable translation. Morishita *et al.* [18] demonstrated the effectiveness of applying back-translation for an unreliable part of a provided corpus. Translations of TED talks are from many independent volunteers, so there may be some differences between translations other than original English, or even they may include some free or over-simplified translations. We aim to fill such a gap using data augmentation.

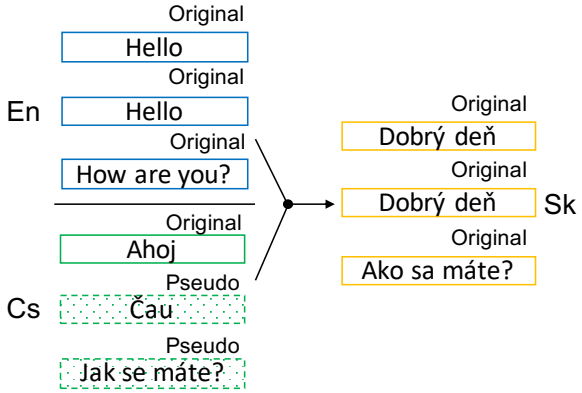
(c) **fill-in and add:** where we both augment the missing part and added pseudo-translations from original translations in the source language except English. This helps prevent introduction of too much noise due to the complete replacement of original translations with pseudo-translations in the second method.



(a) fill-in



(b) fill-in and replace



(c) fill-in and add

Figure 2: Example of three types of augmentation; Language Pair is {English, Czech}-to-{Slovak} and Czech translation corresponding to "How are you?" is missisng. In this example, the dotted background indicates the pseudo-translation produced from multi-source NMT and the white background means the original translation.

4. Experiment

We conducted MT experiments to examine the performance of the proposed method using actual multilingual corpora of TED Talks.

4.1. Data

We used a collection of transcriptions of TED Talks and their multilingual translations. The numbers of these voluntary translations differs significantly by language. We chose three different language sets for the experiments: {English (en), Croatian (hr), Serbian (sr)}, {English (en), Slovak (sk), Czech (cs)}, and {English (en), Vietnamese (vi), Indonesian (id)}. Since the great majority of TED talks are in English, the experiments were designed for the translation from English to another language with the help of the other language in the language set, with no missing portions in the English sentences. Table 1 shows the number of training sentences for each language set. At test time, we experiment with a complete corpus with both source sentences represented, as this is the sort of multi-source translation setting that we are aiming to create models for.

Table 1: "train" shows the number of available training sentences, and "missing" shows the number and the fraction of missing sentences in comparison with English ones.

Pair	Trg	train	missing
en-hr/sr	hr	118949	35564 (29.9%)
	sr	133558	50203 (37.6%)
en-sk/cs	sk	100600	58602 (57.7%)
	cs	59918	17380 (29.0%)
en-vi/id	vi	160984	87816 (54.5%)
	id	82592	9424 (11.4%)

4.2. Baseline Methods

We compared the proposed methods with the following three baseline methods.

One-to-one NMT: a standard NMT model from one source language to another target language. The source language is fixed to English in the experiments. If the target language part is missing in the parallel corpus, such sentences pairs cannot be used in training so they are excluded from the training set.

Multi-encoder NMT with back-translation: a multi-encoder NMT system using English-to-X NMT to fill up the missing parts in the other source language X.[1]

Multi-encoder NMT with ⟨NULL⟩: a multi-encoder NMT system using a special symbol ⟨NULL⟩ to fill up the missing parts in the other source language X [9].

4.3. NMT settings

NMT settings are the same for all the methods in the experiments. We use bidirectional LSTM encoders [12], and global attention and input feeding for the NMT model [13]. The number of dimensions is set to 512 for the hidden and embedding layers. Subword segmentation was applied using SentencePiece [19]. We trained one subword segmentation model for English and another shared between the other two

[1]This is not exactly *back*-translation because the pseudo-translations are not from the target language but from the other source language (English) in our multi-source condition. But we use this familiar term here for simplicity.

languages in the language set because the amount of training data for the languages other than English was small. For parameter optimization, we used Adam [20] with gradient clipping of 5. We performed early stopping, saving parameter values that had the best log likelihoods on the validation data and used them when decoding test data.

4.4. Results

Table 2 shows the results in BLEU [21]. We can see that our proposed methods demonstrate larger gains in BLEU than baseline methods in two language sets: {English, Croatian, Serbian}, {English, Slovak, Czech}. On these pairs, we can say that our proposed method is an effective way for using incomplete multilingual corpora, exceeding other reasonably strong baselines. However, in {English, Vietnamese, Indonesian}, our proposed methods obtained lower scores than the baseline methods. We observed that the improvement by the use of multi-encoder NMT against one-to-one NMT in the baseline was significantly smaller than the other language sets, so multi-encoder NMT was not as effective compared to one-to-one NMT in the first place. Our proposed method is affected by which languages to use, and the proposed method is likely more effective for similar language pairs because the expected accuracy of the pseudo-translation gets better by the help of lexical and syntactic similarity including shared subword entries.

5. Discussion

5.1. Different Types of Augmentation

We examined three types of augmentation: "fill-in", "fill-in and replace", "fill-in and add". In Table 2, we can see that there were no significant differences among them, despite the fact that their training data were very different from each other. We conducted additional experiments using incomplete corpora with lower quality augmentation by one-to-one NMT to investigate the differences of the three types of augmentation. We created three types of pseudo-multilingual corpora using back-translation from one-to-one NMT and trained multi-encoder NMT models using them. Our expectation here was that the aggressive use of low quality pseudo-translations caused to contaminate the training data and to decrease the translation accuracy.

Table 3 shows the results. In {English, Croatian, Serbian} and {English, Slovak, Czech}, we obtained significant drop in BLEU scores with the aggressive strategies ("fill-in and replace" and "fill-in and add"), while there are few differences in {English, Vietnamese, Indonesian}. One possible reason is that the quality of pseudo-translations by one-to-one NMT in Indonesian and Vietnamese was better than the other languages; in other words, the BLEU from one-to-one NMT in Table 2 was sufficiently good without multi-source NMT. Thus the translation performance for Croatian, Serbian, Slovak and Czech could not improve in the experiments here due to *noisy* pseudo-translations of those languages. Contrary, the BLEU from "fill-in and add" was the highest when the target language was Indonesian. We hypothesize that this is due to much smaller fraction of the missing parts in Indonesian corpus as shown in Table 1, so there should be little room for improvement if we fill in only the missing parts even if the accuracy of the pseudo-translations is relatively high.

5.2. Iterative Augmentation

It can be noted that if we have a better multi-source NMT system, it can be used to produce better pseudo-translations. This leads to a natural iterative training procedure where we alternatively update the multi-source NMT systems into the two target languages.

Table 4 shows the results of {English, Croatian, Serbian}. We found that this produced negative results; BLEU decreased gradually in every step. We observed very similar results in the other language pairs, while we omit the actual numbers here. This indicates that the iterative training may be introducing more noise than it is yielding improvements, and thus may be less promising than initially hypothesized.

5.3. Non-parallelism

A problem in the use of multilingual corpora is non-parallelism. In case of TED multilingual captions, they are translated from English transcripts independently by many volunteers, which may cause some differences in details of the translation in the various target languages. For example in {English, Croatian, Serbian}, Croatian and Serbian translations may not be completely parallel. Table 5 shows such an example where the Serbian translation does not have a phrase corresponding to "let me." This kind of non-parallelism may be resolved by overriding such translations with pseudo-translations with "fill-in and replace" and "fill-in and add". Here, the Serbian pseudo-translation includes the corresponding phrase "Dozvolite mi" and can be used to compensate for the missing information. This would be one possible reason of the improvements by "fill-up and replace" or "fill-up and add".

6. Conclusions

In this paper, we examined data augmentation of incomplete multilingual corpora in multi-source NMT. We proposed three types of augmentation; fill-in, fill-in and replace, fill-in and add. Our proposed methods proved better than baseline system using the corpus where missing part was filled up with "⟨NULL⟩", although results depended on the language pair. One limitation in the current experiments with a set of three languages was that missing parts in the test sets could not be filled in. This can be resolved if we use more languages, and we will investigate this in future work.

Table 2: Main results in BLEU for English-Croatian/Serbian (en-hr/sr), English-Slovak/Czech (en-sk/cs), and English-Vietnamese/Indonesian (en-vi/id).

Pair	Trg	baseline method			proposed method		
		one-to-one (En-to-Trg)	multi-encoder NMT (fill up with symbol)	multi-encoder NMT (back translation)	fill-in	fill-in and replace	fill-in and add
en-hr/sr	hr	20.21	28.18	27.57	29.17	29.37	**29.40**
	sr	16.42	23.85	22.73	24.41	**24.96**	24.15
en-sk/cs	sk	13.79	20.27	19.83	20.26	20.43	**20.59**
	cs	14.72	19.88	19.54	20.78	**20.90**	20.61
en-vi/id	vi	24.60	25.70	26.66	**26.73**	26.48	26.32
	id	24.89	**26.89**	26.34	26.40	25.73	26.21

Table 3: The difference of three types of augmentation in BLEU for English-Croatian/Serbian (en-hr/sr), English-Slovak/Czech (en-sk/cs), and English-Vietnamese/Indonesian (en-vi/id). We used one-to-one model to produce pseudo-translations.

Pair	Trg	multi-encoder NMT (back-translation)		
		fill-in	fill-in and replace	fill-in and add
en-hr/sr	hr	**27.57**	24.05	24.79
	sr	**22.73**	17.77	22.02
en-sk/cs	sk	**19.83**	16.75	18.16
	cs	**19.54**	17.04	18.40
en-vi/id	vi	**26.66**	26.39	26.65
	id	26.34	23.90	**26.67**

7. Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Numbers and JP16H05873 and JP17H06101.

8. References

[1] 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.

[2] 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)*, 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.

[3] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, "Multi-Task Learning for Multiple Language Translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 1723–1732.

[4] O. Firat, K. Cho, and Y. Bengio, "Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism," 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. 866–875.

[5] M. Johonson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Vigas, 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.

[6] T.-L. Ha, J. Niehues, and A. Waibel, "Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder," in *Proceedings of the 13th International Workshop on Spoken Language Translation*, Seattle, Washington, December 2016.

[7] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web Inventory of Transcribed and Translated Talks," in *Proceedings of the 16th EAMT Conference*, May 2012, pp. 261–268.

[8] J. Tiedemann, "News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces," in *Recent Advances in Natural Language Processing*, N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, Eds. Borovets, Bulgaria: John Benjamins, Amsterdam/Philadelphia, 2009, vol. V, pp. 237–248.

[9] Y. Nishimura, K. Sudoh, G. Neubig, and S. Nakamura, "Multi-Source Neural Machine Translaion with Missing Data," in *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Association for Computational Linguistics, July 2018, pp. 92–99.

Table 4: BLEU (and BLEU gains compared to step 1) in each step of iterative augmentation.

Pair	Trg	step 1	step 2	step 3	step 4
en-hr/sr	hr	29.17 (+0.00)	29.03 (-0.14)	29.10 (-0.07)	29.05 (-0.12)
	sr	24.41 (+0.00)	24.18 (-0.23)	24.17 (-0.24)	23.95 (-0.46)

Table 5: Example of the Serbian pseudo-translation. This pseudo-translation is the output of {English, Croatian}-to-Serbian translation.

Type	Sentence
Original (En)	So **let me** conclude with just a remark to bring it back to the theme of choices.
Original (Sr)	Da zakljuim jednom konstatacijom kojom se vraam na temu izbora.
Pseudo (Sr)	**Dozvolite mi** da zakljuim samo jednom opaskom, da se vratim na temu izbora.

[10] B. Zoph and K. Knight, "Multi-Source Neural 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. 30–34.

[11] E. Garmash and C. Monz, "Ensemble Learning for Multi-Source Neural Machine Translation," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, December 2016, pp. 1409–1418.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proceedings of the 3rd International Conference on Learning Representations*, May 2015.

[13] T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 1412–1421.

[14] 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)*. Association for Computational Linguistics, 2016, pp. 86–96.

[15] K. Imamura, A. Fujita, and E. Sumita, "Enhancement of encoder and attention using target monolingual corpora in neural machine translation," in *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*. Association for Computational Linguistics, 2018, pp. 55–63. [Online]. Available: http://aclweb.org/anthology/W18-2707

[16] X. Wang, H. Pham, Z. Dai, and G. Neubig, "Switchout: an efficient data augmentation algorithm for neural machine translation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[17] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 1317–1327. [Online]. Available: https://aclweb.org/anthology/D16-1139

[18] M. Morishita, J. Suzuki, and M. Nagata, "NTT Neural Machine Translation Systems at WAT 2017," in *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, 2017, pp. 89–94.

[19] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018, pp. 66–75.

[20] D. P. K. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, May 2015.

[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 of the Association for Computational Linguistics (ACL)*, Philadelphia, July 2002, pp. 311–318.

Transfer Learning in Multilingual Neural Machine Translation with Dynamic Vocabulary

Surafel M. Lakew[†+], *Aliia Erofeeva*[†], *Matteo Negri*[+], *Marcello Federico*[⋆], *Marco Turchi*[+]

[†]University of Trento, [+]Fondazione Bruno Kessler, Trento, Italy
[⋆]Amazon AI, East Palo Alto, CA 94303, USA
[†]name.surname@unitn.it, [+]surname@fbk.eu, [⋆]marcfede@amazon.com

Abstract

We propose a method to transfer knowledge across neural machine translation (NMT) models by means of a shared dynamic vocabulary. Our approach allows to extend an initial model for a given language pair to cover new languages by adapting its vocabulary as long as new data become available (i.e., introducing new vocabulary items if they are not included in the initial model). The parameter transfer mechanism is evaluated in two scenarios: *i)* to adapt a trained single language NMT system to work with a new language pair and *ii)* to continuously add new language pairs to grow to a multilingual NMT system. In both the scenarios our goal is to improve the translation performance, while minimizing the training convergence time. Preliminary experiments spanning five languages with different training data sizes (i.e., 5k and 50k parallel sentences) show a significant performance gain ranging from +3.85 up to +13.63 BLEU in different language directions. Moreover, when compared with training an NMT model from scratch, our transfer-learning approach allows us to reach higher performance after training up to 4% of the total training steps.

1. Introduction

Neural Machine Translation (NMT) has shown to surpass phrase based Machine Translation approaches not only in high-resource language settings, but also with low-resource [1] and zero-resource translation tasks [2, 3]. Although recent approaches yield promising results, training models in low-resource settings remains a challenge for MT research [4]. [2] have shown that a multilingual NMT (M-NMT) model that utilizes a concatenation of data covering multiple language pairs (including high-resourced ones) can result in better performance in the low-resource translation task. Alternatively, [5] proposed a transfer-learning approach from an NMT "*parent-model*" trained on a high-resource language to initialize a "*child-model*" in a low-resource setting showing consistent translation improvements on the latter task.

Though effective, training models on a concatenation of data covering multiple language pairs or initializing them by transferring knowledge from a parent model does not consider the dynamic nature of new language vocabularies. In relation to how and when model vocabularies are built, there can be two distinct scenarios. In the first one, all the training data for all the language pairs are available since the beginning. In this case, either separate or joint sub-word segmentation models can be applied on the training material to build vocabularies that represent all the data [6, 7]. In the second scenario, training data covering different language directions are not available at the same time (most real-world MT training scenarios fall in this category, in which new data or new needs in terms of domains or language coverage emerge over time). In such cases, either: *i)* new MT models are trained from scratch with new vocabularies built from the incoming training data, or *ii)* the word segmentation rules of a prior (parent) model are applied on the new data to continue the training as a fine-tuning task. In all the scenarios, accurate word segmentation is crucial to avoid out-of-vocabulary (OOV) tokens. However, different strategies for the different training conditions can result in longer training time or performance degradations. More specifically, limiting the target task with the initial model vocabulary will result in: *i)* a word segmentation that is unfavorable for the new language directions and *ii)* a fixed vocabulary/model dimension despite the varying language and training dataset size.

NMT models are not only data-demanding, but also require considerable time to be trained, optimized, and put into use. In particular real-word scenarios, strict time constraints prevent the possibility to deploy and use NMT technology (consider, for instance, emergency situations that require to promptly enable communication across languages [8]). On top of this, when the available training corpora are limited in size, delivering usable NMT systems (i.e., systems that can be used with the requirement of not making severe errors [9]) becomes prohibitive. In summary: *i)* on the data side, acquiring new training material for x undefined languages is costly and not always possible, and *ii)* on the model side, building an NMT system from scratch when new data become available raises efficiency and performance issues that are particularly relevant in low-resource scenarios.

We address these issues by introducing a method to transfer knowledge across languages by means of a dynamic vo-

(*) Work conducted while this author was at FBK.

cabulary. Starting from an initial model, our method allows to build new NMT models, either in a single or multiple language translation directions, by dynamically updating the initial vocabulary to new incoming data. For instance, given a trained German-English NMT system (L_1), the learned parameters can be transferred across models, while adopting new language vocabularies. In our experimental setting we test two transfer approaches:

- `progAdapt`: train a chain of consecutive M-NMT models by transferring the parameters of an initial model for L_1 to new language pairs $L_2 \dots L_N$. In this scenario, the goal is to maximize performance on the new language pairs.
- `progGrow`: progressively introduce new language pairs to the initial model L_1 to create a growing M-NMT model covering N translation directions. In this scenario, the goal is to maximize performance on all the language pairs.

Our experiments are carried out with Italian−English, Romanian−English, and Dutch−English training data sets of different size, ranging from low-resource (50k) to extremely low-resource (5k) conditions.

As such, in a rather different way from previous work [5], we show our transfer-learning approach in a multilingual NMT model with dynamic vocabulary both in the source and target directions. Our contributions are as follows:

- we develop a transfer-learning technique for NMT based on a dynamic vocabulary, which adapts the parameters learned on a parent task (language direction) to cover new target tasks;
- through experiments in different scenarios, we show that our approach improves knowledge transfer across NMT models for different languages, particularly in low-resource conditions;
- we show that, with our transfer learning approach, it is possible to train a faster converging model that achieves better performance than a system trained from scratch.

2. Related work

2.1. Transfer Learning

Recent efforts [10, 11] in natural language processing (NLP) research have shown promising results when transfer-learning techniques are applied to leverage existing models to cope with the scarcity of training data in specific domains or language settings. The advancements in NLP came following a much larger impact of transfer-learning in computer vision tasks, such as classification and segmentation, either using features of ImageNet [12] or by fine-tuning the last layers of a deep neural network [13]. Specific to NLP, pre-trained word embeddings [14] used as input to the first layer of the network have become a common practice. In a broader sense, pre-trained models have been successfully exploited for several NLP tasks. [15] used an MT model as a pre-training step to further contextualize word vectors for downstream tasks like sentiment analysis, question classification, textual entailment, and question answering. In a similar way, a language model is utilized for pre-training in sequence labeling tasks [16], question answering, textual entailment, and sentiment analysis [17].

Close to our approach, [5] explored techniques for transfer-learning across two NMT models. First, a "parent" model is trained with a large amount of available data. Then the encoder-decoder components are transferred to initialize the parameters of a low-resourced "child" model. In this parent-child setting, the decoder parameters of the child model are fixed at the time of fine-tuning. Later, in [18], the parent-child approach has been extended to analyze the effect of using related languages on the source side.

Although this work shares a related approach with [5], we diverge by our hypothesis not to selectively update only the encoder, allowing all the parameters to be updated as a beneficial strategy in our setting. Our strategy is based on both the source→target and target→source translation directions that we consider as transferable. Moreover, our transfer-learning approach relies on a dynamic vocabulary that enforces changes in the trainable parameters of the network in contrast to fixing them[1].

2.2. Multilingual NMT

In a one-to-many multilingual translation scenario, [19] proposed a multi-task learning approach that utilizes a single encoder for the source language and separate attention mechanisms and decoders for each target language. [20] used distinct encoder and decoder networks for modeling multiple language pairs in a *many-to-many* setting. Later, [21] introduced a way to share the attention mechanism across multiple languages. Aimed at avoiding translation ambiguities on the decoder side, a *many-to-one* character level NMT setup [22] and a two/multi-source NMT [23] were also proposed. Inspired by [24], who automatically annotated the source side with artificial flags to manage the politeness level of the output, other works focused on controlling the grammatical voice [25], the text domain [26, 27], and enforcing gender agreement [28]. Simplified yet efficient multilingual NMT approaches have been proposed by [2] and [3]. The approach in [3] applies a language-specific code to words from different languages in a mixed-language vocabulary. The approach in [2], by prepending a *language flag* to the input string, greatly simplified multilingual NMT eliminating the need of having separate encoder/decoder networks and attention mechanism for each new language pair. In this work we follow a similar strategy by incorporating an artificial language flag.

[1] In future work, we plan to further study which parameters are more beneficial if transferred and which part of the network to selectively update.

3. Transfer Learning in M-NMT

In this work, we cast transfer-learning in a multilingual neural machine translation (M-NMT) task as the problem of dynamically changing/updating the vocabulary of a trained NMT system. In particular, transfer-learning across models is assumed to: *i)* include a strategy to add new language-specific items to an existing NMT vocabulary, and *ii)* be able to manage a number of new translation directions in different transfer rounds, either by covering them one at a time (i.e., in a chain where new languages are covered stepwise) or simultaneously (i.e., pursuing all directions at each step). Our investigation focuses on two aspects. The first one is how the parameters of an existing model can be transferred to a target one for a new language pair. The second aspect is how to limit the impact of parameters' transfer on the performance of the initial model as long as new language directions are added. For convenience, we refer to our approach as `TL-DV` (*Transfer-Learning using Dynamic Vocabulary*).

As shown in Figure 1, our transfer-learning approach is evaluated in two conditions:

- `progAdapt`, in which progressive updates are made on the assumption that new target NMT task data become available for one language direction at a time (i.e., new language directions are covered sequentially). In this condition, our goal is to maximize performance on the new target tasks by taking advantage of parameters learned in their parent task;

- `progGrow`, in which progressive updates are made on the same assumption of receiving new target task data as in `progAdapt`, but with the additional goal of preserving the performance of the previous language directions.

We discuss these two scenarios below in §3.2 and §3.3.

3.1. Dynamic Vocabulary

In the defined scenarios, we update the vocabulary V_p of the previous model with the current language direction vocabulary V_c. The approach simply keeps the intersection (same entries) between V_p and V_c, whereas replacing V_p entries with V_c if the entries of the former vocabulary do not exist in the latter. At training time, these new entries are randomly initialized, while the intersecting items maintain the embeddings of the former model. The alternative approach to dynamic vocabulary in a continuous model training is to use the initial model vocabulary V_p, which we refer to as static-vocabulary.

3.2. Progressive Adaptation to New Languages

In this scenario, starting from the `init` model (L_1), we perform progressive adaptation by initializing the training of a model at each step (L_n) with the previous model (L_{n-1}). At time of reloading the model from L_{n-1}, a `TL-DV` update is performed as described in §3. In this approach, the dataset of the initial model is not included at the current training stage. This allows the adaptation to the new language without unnecessary word segmentation that may arise by applying the initial model's segmentation rules. As shown in Figure 1 (left), the adaptation on any of the L_n stages is language independent, though subject to the available training dataset. We refer to the application of this approach in the experimental settings and discussion as `progAdapt`.

3.3. Progressive Growth of Translation Directions

In this scenario, an initial model L_1 is simultaneously adapted to an incremental number of translation directions, under the constraint that the level of performance on L_1 has to be maintained. For a simplified experimental setup, we will incorporate a single language pair (source→target) at a time, when adapting to L_n from L_{n-1} (see Figure 1 (right)). We refer to the application of this approach in the experimental settings and discussion as `progGrow`.

4. Experimental Setting

4.1. Dataset and Preprocessing

Our experimental setting includes the `init` model language pair (German-English) and three additional language pairs (Italian-English, Romanian-English, and Dutch-English) for testing the proposed approaches. We use publicly available datasets from the WIT[3] TED corpus [29]. Table 1 shows the summary of the training, dev, and test sets. To simulate an extremely low-resource (M_{ELR}) and low-resource (M_{LR}) model settings, 5K and 50K sentences are sampled from the last three language pairs' training data.

At the preprocessing step, we first tokenize the raw data and remove sentences longer than 70 tokens. As in [2], we prepend a "language flag" on the source side of the corpus for all multilingual models. For instance, if a German source is paired with an English target, we append `<2ENG>` at the beginning of source segments. Next, a shared byte pair encoding (BPE) model [6] is trained using the union of the source and target sides of each language pair. Following [30], the number of BPE segmentation rules is set to $8,500$ for the data size used in our experimental setting. At different levels of training (L_i), a BPE model with respect to the language pairs is then used to segment the training, dev, and test data into sub-word units. While, the vocabulary size of the `init` is fixed, the vocabulary varies in the consecutive training stages depending on the overlap of sub-word units and lexical similarity between two language pairs.

4.2. Experimental Settings

All systems are trained using the Transformer [31] model implementation of the OpenNMT-tf sequence modeling framework[2] [32]. At training time, to alternate between dynamic

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

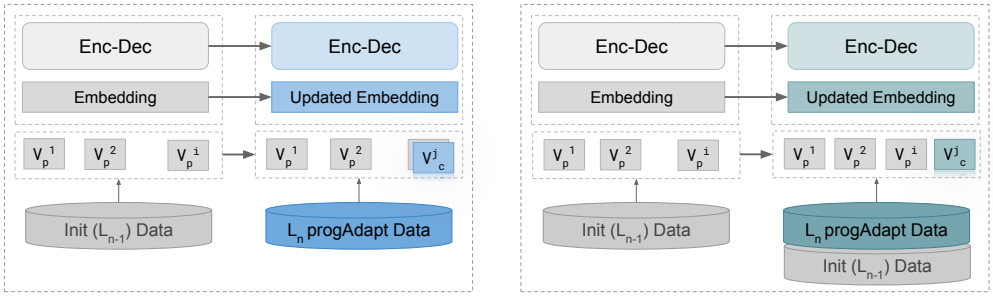


Figure 1: *Transfer-Learning; (left) from an initial NMT model to a new language pair, model is applied after inserting the new vocabulary entries, for instance, the initial model L_{n-1} parameters are transfered to L_n with the updated embedding space (i.e., keeping V_p^1, V_p^2 as overlapping entries, while replacing the non-overlapping V_p^i with V_c^j new language vocabularies), and (right) from an initial model L_{n-1} to L_n, but incorporating both the previous and new language pair data and vocabulary entries.*

Table 1: *Languages and dataset sizes for train, dev, and test sets of the* `init` *model for De-En direction and other pairs assumed to be received progressively (It-En, Ro-En, Nl-En).*

Language	Train	Dev	Test	Received
German(De)-En	200k	1497	1138	`init`
Italian(It)-En	5k/50k	1501	1147	L_2
Romanian(Ro)-En	5k/50k	1633	1129	L_3
Dutch(Nl)-En	5k/50k	1726	1181	L_4

and static vocabulary, we utilized an updated version of the script within the same framework. For all trainings, we use LazyAdam, a variant of the Adam optimizer [33], with an initial learning rate constant of 2 and a dropout [34, 35] of 0.3. The learning rate is increased linearly in the early stages (*warmup_training_steps*=16, 000), and afterwards it is decreased with an inverse square root of the training step.

To train our models using Transformer, we employ a uniform setting with 512 hidden units and embedding dimension, and 6 layers of self-attention encoder-decoder network. The training batch size is of 4096 sub-word tokens. At inference time, we use a beam size of 4 and a batch size of 32. Following [31] and for a fair comparison, all baseline experiments are run for 100k training steps, i.e., all models are observed to converge within these steps. The consecutive experiments converge in variable training steps. However, to make sure a convergence point is reached, all restarted experiments on L_i are run for additional 50K steps. All models are trained on a `GeForce-GTX-1080` machine with a single GPU. Systems are compared in terms of BLEU [36] using the *multi-bleu.perl* implementation[3], on the single references of the official IWSLT test sets.

4.3. Baseline Models

To evaluate and compare with our approach, we train single language pair baseline models corresponding to the newly introduced language pairs at each L_i training stage. The baseline models, referred to as `Bi-NMT`, are separately trained from scratch in a bi-directional setting (i.e., source $\leftrightarrow$ target). In addition, we report scores from a multilinugal (`M-NMT`) model trained with the concatenation of all available data in each training stage. The alternative baseline are built by fine-tuning the `init` model. These models use the vocabulary (word segmentation rules) of the `init` model, avoiding the proposed dynamic vocabulary. This fine-tuning approach is prevalent in continued model trainings, for adapting NMT models [37, 38] or improving zero-shot and low-resource translation tasks [39, 40, 41]. For the alternative baseline where we fine-tune `init` with its static-vocabulary, we observed that results were mostly analogous to `Bi-NMT` models. Hence, we avoided this comparison in this work and relied on the former baselines.

[3] A script from the Moses SMT toolkithttp://www.statmt.org/moses

5. Results and Discussion

Experiments are performed using the `progAdapt` (see §3.2) and `progGrow` (§3.3) approaches. The experimental results with the associated discussion are presented in Table 2 for models characterized by relatively low-resource data (M_{LR}), and in Table 3 for an extremely low-resource condition (M_{ELR}). In both dataset conditions, the performance of the proposed approaches is compared with the baseline systems (`Bi-NMT` and `M-NMT`, see §4.3).

The `init` model which is trained with a data size 4X larger than M_{LR} and 40X the size of M_{ELR}, achieves BLEU scores of 26.74 and 23.30, respectively, for the De-En and En-De directions. In Table 2 and 3, the `progAdapt` is reported for each training stage (i.e., L_2, L_3, and L_4), whereas the `progGrow` is reported for the final stage L_4. Moreover, Table 4 analyzes the effect of language relatedness and training stage reordering in our `TL-DV` approach. Bold highlighted BLEU scores show the best performing approach, while the $^{\uparrow\downarrow}$ arrows indicate statistically significant differences of the hypothesis against the better performing baseline (`M-NMT`) using bootstrap resampling ($p < 0.05$) [42].

Table 2: *M_{LR} models performance i) at L_1 for the* `init` *De-En direction and baseline (*`Bi-NMT`*) It-En, Ro-En, and Nl-En directions, ii) at $L_{2/3/4}$ for* `progAdapt`*, and iii) at L_4 for the* `progGrow` *approach.*

	Dir	De-En	It-En	Ro-En	Nl-En
`Init/Bi-NMT`	>	**26.74**	25.21	10.80	21.75
	<	23.30	22.39	12.94	19.75
`M-NMT`	>	24.14	26.42	22.17	24.00
	<	21.80	23.57	17.35	21.25
`ProgAdapt`	>	-	↑**30.08**	↑**24.43**	↑**26.36**
	<	-	↑**26.24**	↑**20.31**	↑**25.52**
`ProgGrow`	>	26.22	↑**29.61**	**23.23**	**24.78**

Table 3: *M_{ELR} models performance i) at L_1 for the* `init` *De-En direction and baseline (*`Bi-NMT`*) It-En, Ro-En, and Nl-En directions, ii) at $L_{2/3/4}$ for* `progAdapt`*, and iii) at L_4 for the* `progGrow` *approach.*

	Dir	De-En	It-En	Ro-En	Nl-En
`Init/Bi-NMT`	>	**26.74**	7.64	4.56	5.69
	<	23.30	5.25	3.86	5.14
`M-NMT`	>	24.96	**16.26**	**12.67**	**15.59**
	<	21.67	10.38	8.67	12.72
`ProgAdapt`	>	-	↓15.16	↓11.03	↓11.52
	<	-	↑**14.40**	↑**11.10**	**13.57**
`ProgGrow`	>	25.61	↓15.02	↓11.20	↓13.56

5.1. Low-Resource Setting

For each language pair (i.e., It-En, Ro-En, and Nl-En), the results of the baseline models `Bi-NMT` trained using the available 50K parallel data (M_{LR} setting) are presented in the first two rows of Table 2. The `progAdapt` results are reported from three consecutive adaptations to new language directions. These include the `init` to It-En, followed by the adaptation to Ro-En, and then to Nl-En. Compared to the corresponding `Bi-NMT` and M-NMT models, all of the three progressive adaptations using the dynamic vocabulary technique achieved a higher performance gain.

If we look at the specific level of adaption (L_i) against the `Bi-NMT`, we observe that the It-En direction showed a +4.87 and +3.85 gain for the En and It target, respectively. When we take this model and continue the adaptation to Ro-En and Nl-En, we see a similar trend where the highest gain is observed on L_3 for the Ro-En direction with +13.63 and +7.37 points. These significant improvements over the baseline models tell us that transfer-learning using dynamic vocabulary in a multilingual setting is a viable direction. Its capability to quickly tune the representation space of the `init` model to deliver improved results is an indication of the importance of using different word representations for each language pair[4].

In case of the `progGrow`, we observed a similar improvement trend as in the `progAdapt` approach. The results are reported from the final stage (L_4) of the model growth, but improvements are consistent throughout the L_2 and L_3 stages. The `M-NMT` outperformed the `Bi-NMT` models except for De-En pair. However, compared to the multilingual model as an alternative method for achieving cross-lingual transfer-learning, our approach shows improvements in the consecutive training stages. Overall, our observation is that the suggested `progGrow` model can accommodate new translation directions when the data are received. Most importantly, improvements are observed for these newly introduced languages without altering the performance of the `init` model in the De-En direction.

Specific to each language direction, It-En shows a comparable performance with the `progAdapt` approach, whereas in case of Ro-En and Nl-En a small degradation ranging from 0.47 (De-En) to 1.58 (Nl-En) is observed. The loss in performance is likely due to the increased ambiguities in the encoder side of the `progGrow` model, where at both training and inference time there does not exist a disambiguation mechanism between languages except the prepended language flag. This observation, which sheds a light on our initial expectation of more data aggregation benefiting the model performance, requires further investigation.

5.2. Extremely Low-Resource Setting

In a similar way with what we observed in the M_{LR} experiments, the baseline models in the extremely low-resource setting demonstrate poor performance. Looking at our approaches, we observe a relatively higher gain at the first stage of `progAdapt` and `progGrow`. For instance, for the It-En pair there is a +7.52 improvement compared to the +4.87 in the M_{LR} models (see Table 2) over the `Bi-NMT` model. In the subsequent additional language directions (i.e., Ro-En and Nl-En), we also observe a similar trend. However, in comparison with the M-NMT, both of our approach perform poorly when translating to the En target. The main reason for this could be the aggregation of all the available data for a single run in the `M-NMT` model, while our approaches exploit data when it becomes available in a continuous training. Alternatively the distance between each language pair could play a significant role when we adapt in an extremely sparse data.

prog-Adapt/Grow with Related Languages. When related language pairs are consecutively added (L_{n-1} and L_n) at each training stages, our `TL-DV` approach showed the best performance. For instance, for the Nl-En experiments, we changed the sequence of the added language pair

[4]We reserve the adaptation from the `init` model directly to all the three new language pairs and the comparison with the current setting for future work.

Table 4: *M_{LR} and M_{ELR} models performance at L_1 for* `progAdapt` *and* `progGrow` *approaches in a closely related De-En (*`init`*) and Nl-En language pairs setting.*

		M_{LR}		M_{ELR}	
	Dir	De-En	Nl-En	De-En	Nl-En
`ProgAdapt`	$>$	-	↑**27.23**		**16.21**
	$<$	-	↑25.51		**15.86**
`ProgGrow`	$>$	26.62	↑**26.41**	26.52	↑**15.52**

moving from a random order to a sequence based on the similarity to the `init` model. Table 4 shows the results from `progAdapt` and `progGrow`, when the Nl-En pair is used at the L_1 training stage. The M_{LR} results confirm the trend observed in Table 2, however, with a relatively better performance when translating in to English. Most importantly, the M_{ELR} results show a consistent and larger gain of $+4.69$ (Nl-En) and $+2.29$ (En-Nl) with the `progAdapt`, and $+1.96$ (Nl-En) with `progGrow` compared to the corresponding results in Table 3. Thus, we emphasize on the degree of language similarity as a direct influencing factor when incorporating a new language pair both in `progAdapt` and `progGrow` approaches. .

Prog-Adapt/Grow with Faster Convergence. The other main advantage of our `TL-DV` approach comes from the time a model takes to restart from the `init` model and reach a convergence point with better performance. In all experiments with our `TL-DV` approach a converged model is found within 10K steps for M_{ELR} and 20K for M_{LR} training settings. Compared to ≈100K steps needed by a model trained from scratch to reach good performance, our approach takes only 4% to 20% of training steps with significantly higher performance. For instance, taking into consideration the M_{ELR} models, Figure 2 illustrates the steps required for the baseline systems to converge (Table 3), in comparison with our approach where `progGrow` shows to converge slightly faster than `progAdapt`. However, with the relatively larger data of the M_{LR} models, the `progAdapt` approach proves to converge much faster than `progGrow`, for the reason that the newly introduced vocabulary and training dataset sizes are smaller compared to the concatenation of the `init` and L_i data.

We further analyzed the influence of shared vocabularies between models L_i and L_{i+1} on the performance of `TL-DV`. For this discussion, we took the `progAdapt` M_{LR} model from all stages. Figure 3 summarizes the improvement differences from consecutive models in relation to the percentage of shared vocabularies. For instance, `init` and the L_2 (It-En) model vocabularies have a 47% overlap, whereas L_3 and L_4 share 53% and 51% with the previous model. The interesting aspect of the shared vocabulary comes from the increase in model performance with a higher fraction of shared vocabulary entires. Thus, a larger number of shared parameters between two consecutive models allows for a better gain in performance of the latter.

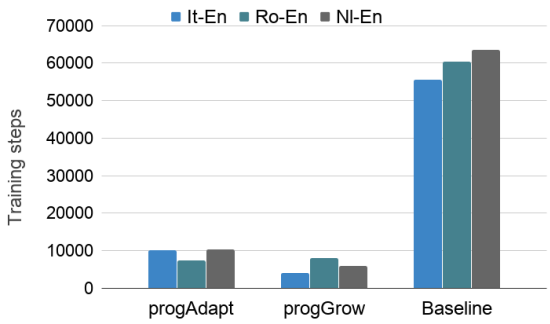


Figure 2: *Model training steps number comparison for the three different language pairs between the baseline (rightmost) and the proposed approaches in the M_{ELR} setting.*

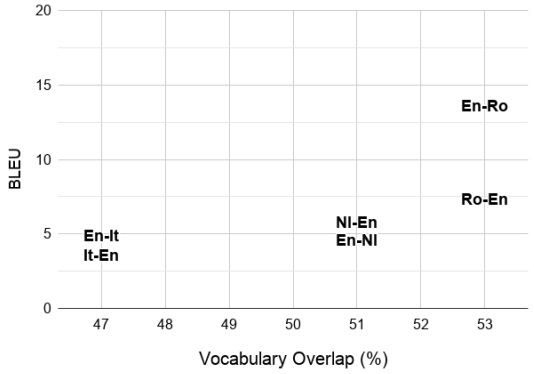


Figure 3: *The difference in performance between the baseline and* `progAdapt` *models (Tgt→Src and Src→Tgt directions) in relation with the shared vocabulary between model L_i and new language pair model L_{i+1}.*

The results achieved by the transfer-learning with dynamic vocabulary approach in two different training size conditions show that: *i)* adapting a trained NMT model to a new language pair improves performance on the target task significantly, and *ii)* it is possible to train a model faster to achieve better performance. Overall, the capability of injecting new vocabularies for new language pairs in the initial model is a crucial aspect for efficient and fast adaptation steps.

6. Conclusions

In this work, we proposed a transfer-learning approach within a multilingual NMT. Experimental results show that our dynamic vocabulary based transfer-learning improves model performance in a significant way of up to 9.15 in an extremely low-resource and up to 13.0 BLEU in a low-resource setting over a bilingual baseline model.

In future work, we will focus on finding the optimal way of transferring model parameters. Moreover, we plan to test our approach for various languages and language varieties.

7. Acknowledgments

This work has been partially supported by the EC-funded project ModernMT (H2020 grant agreement no. 645487). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. Moreover, we thank the Erasmus Mundus European Program in Language and Communication Technology.

8. References

[1] 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.

[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] P. Koehn and R. Knowles, "Six challenges for neural machine translation," *arXiv preprint arXiv:1706.03872*, 2017.

[5] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," *arXiv preprint arXiv:1604.02201*, 2016.

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

[7] 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.

[8] W. Lewis, "Haitian creole: How to build and ship an mt engine from scratch in 4 days, 17 hours, & 30 minutes," in *14th Annual conference of the European Association for machine translation*. Citeseer, 2010.

[9] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico, "Neural versus phrase-based mt quality: An in-depth analysis on english-german and english-french," *Computer Speech & Language*, vol. 49, pp. 52–70, 2018.

[10] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 328–339.

[11] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2017.

[12] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[14] Y. Qi, D. S. Sachan, M. Felix, S. J. Padmanabhan, and G. Neubig, "When and why are pre-trained word embeddings useful for neural machine translation?" *arXiv preprint arXiv:1804.06323*, 2018.

[15] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Advances in Neural Information Processing Systems*, 2017, pp. 6294–6305.

[16] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," *arXiv preprint arXiv:1705.00108*, 2017.

[17] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[18] T. Q. Nguyen and D. Chiang, "Transfer learning across low-resource, related languages for neural machine translation," *arXiv preprint arXiv:1708.09803*, 2017.

[19] 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.

[20] 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.

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

[22] J. Lee, K. Cho, and T. Hofmann, "Fully character-level neural machine translation without explicit segmentation," *arXiv preprint arXiv:1610.03017*, 2016.

[23] B. Zoph and K. Knight, "Multi-source neural translation," *arXiv preprint arXiv:1601.00710*, 2016.

[24] R. Sennrich, B. Haddow, and A. Birch, "Controlling politeness in neural machine translation via side constraints," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 35–40.

[25] H. Yamagishi, S. Kanouchi, T. Sato, and M. Komachi, "Controlling the voice of a sentence in japanese-to-english neural machine translation," in *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, 2016, pp. 203–210.

[26] W. Chen, E. Matusov, S. Khadivi, and J.-T. Peter, "Guided Alignment Training for Topic-Aware Neural Machine Translation," in *Association for Machine Translation in the Americas (AMTA)*, jul 2016. [Online]. Available: http://arxiv.org/abs/1607.01628

[27] C. Chu, R. Dabre, and S. Kurohashi, "An Empirical Comparison of Domain Adaptation Methods for Neural Machine Translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017, pp. 385–391. [Online]. Available: http://arxiv.org/abs/1701.03214%0Ahttp://aclweb.org/anthology/P17-2061

[28] M. Elaraby, A. Y. Tawfik, M. Khaled, and A. Osama, "Gender Aware Spoken Language Translation Applied to English-Arabic," in *Proceedings of the Second International Conference on Natural Language and Speech processing*, 2018. [Online]. Available: https://arxiv.org/pdf/1802.09287.pdf

[29] 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.

[30] M. Denkowski and G. Neubig, "Stronger baselines for trustable results in neural machine translation," in *Proceedings of the First Workshop on Neural Machine Translation*, 2017, pp. 18–27.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[32] 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.

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

[34] 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.

[35] 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.

[36] 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.

[37] P. Michel and G. Neubig, "Extreme adaptation for personalized neural machine translation," *arXiv preprint arXiv:1805.01817*, 2018.

[38] D. Vilar, "Learning hidden unit contribution for adapting neural machine translation models," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, vol. 2, 2018, pp. 500–505.

[39] S. M. Lakew, Q. F. Lotito, N. Matteo, T. Marco, and F. Marcello, "Improving zero-shot translation of low-resource languages," in *14th International Workshop on Spoken Language Translation*, 2017.

[40] G. Lample, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," *arXiv preprint arXiv:1711.00043*, 2017.

[41] J. Gu, H. Hassan, J. Devlin, and V. O. Li, "Universal neural machine translation for extremely low resource languages," *arXiv preprint arXiv:1802.05368*, 2018.

[42] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, vol. 4, 2004, pp. 388–395.

Machine Translation Human Evaluation: an investigation of evaluation based on Post-Editing and its relation with Direct Assessment

Luisa Bentivogli[(1)], *Mauro Cettolo*[(1)], *Marcello Federico*[(2)], *Christian Federmann*[(3)]

[(1)] FBK - Trento, Italy
[(2)] Amazon AI - East Palo Alto, CA, USA
[(3)] Microsoft Cloud+AI - Redmond, WA, USA

[bentivo|cettolo]@fbk.eu
marcfede@amazon.com
chrife@microsoft.com

Abstract

In this paper we present an analysis of the two most prominent methodologies used for the human evaluation of MT quality, namely evaluation based on Post-Editing (PE) and evaluation based on Direct Assessment (DA). To this purpose, we exploit a publicly available large dataset containing both types of evaluations. We first focus on PE and investigate how sensitive TER-based evaluation is to the type and number of references used. Then, we carry out a comparative analysis of PE and DA to investigate the extent to which the evaluation results obtained by methodologies addressing different human perspectives are similar. This comparison sheds light not only on PE but also on the so-called reference bias related to monolingual DA. Also, we analyze if and how the two methodologies can complement each other's weaknesses.

1. Introduction

The evaluation of machine translation (MT) is of crucial importance and has a long research history. Both human and automatic evaluation have been explored extensively within the MT community, in the effort to find more and more suitable, efficient and reliable methods and metrics. Automatic metrics play a central role in the progress of the field and the improvement of MT quality over time. However, they represent a proxy for human evaluation which – despite being costly and time-consuming – is to be considered primary.

Among the various human evaluation methods that have been devised and tested along the years, currently two approaches have become well-established standards in the field, namely evaluation based on *Post-Editing* (PE) and evaluation based on *Direct Assessment* (DA).

In the *PE-based evaluation*, the MT outputs are post-edited, *i.e.* manually corrected, according to the source sentence (*bilingual* PE) or to an existing reference translation (*monolingual* PE). The original MT outputs are then evaluated against their post-edited versions through TER-based automatic metrics [1]. Relying on the post-edit instead of an independently created reference translation ensures that only true errors in the MT output are counted, and not those differences due to linguistic variation, which are accounted for by post-editors. PE has become the standard evaluation metric for the yearly evaluation campaign of the International Workshop of Spoken Language Translation since 2013 (IWSLT-2013) and is described in detail in [2].

The *DA-based* evaluation [3] consists of collecting human assessments of translation quality for single MT systems. Assessors see a candidate translation and a corresponding translation hint (*e.g.* the source text, a reference translation, or multimodal content) and are asked to assign a quality score from 0 to 100. DA has become the standard evaluation metric for the yearly Conference on Machine Translation (WMT) in 2017 [4]. Following the findings of WMT17, the main focus for DA is on semantic transfer (which corresponds to *adequacy*) while syntactic transfer (or *fluency*) has turned out to be less relevant. Traditionally, in the DA task MT quality is assessed according to a reference translation, without access to the source text. This is called *reference-based DA* (*DA-ref*). A problematic issue with DA-ref is its inherent dependence on reference translations, which can lead to *reference bias*, both in the form of giving an implicit boost to candidate translations which are very similar (*e.g.*, in syntax or lexical choice) to the corresponding reference text, or by penalizing good translations because of translation errors affecting the reference itself. To address the reference bias, *source-based DA* (*DA-src*) can be used, where translation quality is assessed directly according to the source text. DA-src has been tested on a large scale for the first time in the IWSLT 2017 evaluation campaign [5].

DA and PE are different and complementary methodologies, not only from the point of view of their design but also concerning their practical usage. First, the two evaluation

(2) Work conducted while this author was at FBK.

methods 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. Furthermore, while DA is based only on human annotators, in PE an automatic component (*i.e.* TER) is applied to quantify the errors of the MT output. Finally, in terms of data collection DA is less costly then PE and thus more viable when used within the research scenario; however PE has the double advantage of *(i)* producing a set of additional reference translations, and *(ii)* being particularly suitable for performing fine-grained analyses of the MT systems, since it produces a set of edits pointing to specific translation errors [6, 7, 8].

Given the importance of human evaluation for MT improvement and the specific features of these two most prominent frameworks, we present an empirical analysis of these different methodologies as a contribution to their better understanding.

The analysis is conducted on the publicly available Human Evaluation dataset created as part of the IWSLT 2017 evaluation campaign [5]. The dataset covers two language directions, namely Dutch-to-German and Romanian-to-Italian. For each direction, it includes DA-src, DA-ref, and PE human evaluation data for nine different state-of-the art neural MT systems on the same 603 segments. DA evaluation was performed by linguists, while professional translators carried out the bilingual PE task. Besides making our study possible, the size, variety and high quality of this three-way evaluation dataset ensure sound empirical analyses and generalizable outcomes.

The main investigations presented in the paper are:

- *New analyses on PE data*. The availability of multiple post-edits allows us to investigate how sensitive TER-based evaluation is to the type (external *versus* post-edit) and number of references used, both in terms of reliability and informativeness of the evaluation;
- *New comparative analysis of PE and DA*. In this empirical comparison we investigate the extent to which the evaluation results obtained by methodologies addressing different human perspectives are similar. This investigation gives us insight not only on PE but also on the relations between DA-src and DA-ref. Also, we analyze if and how PE and DA can complement each other's weaknesses.

2. Related Work

Human Evaluation has always received a lot of attention in the field of MT and many methodologies have been devised and tested in different scenarios. The same holds for the two methods addressed in this paper.

PE-based evaluation was the focus of various studies [1, 9, 6] and was commonly employed in large-scale evaluation campaigns, such as IWSLT [2, 10, 11, 12, 5] and the MT Quality Estimation Task at WMT-2015 [13].

Also research on DA has been very active since its introduction as method for human evaluation of MT [3, 14]. Large-scale evaluations were carried out through DA-ref [4] and, more recently, also through DA-src [5, 15].

As specifically regards the impact of different numbers and types of post-edits in PE-based evaluation, a study on multiple references was presented in [16], but it did not target PE-based evaluation.

Concerning the issue of reference bias in DA-ref evaluation, it was examined in detail in [17], [18], and [19]. To this aim, [17] compares directly DA-src and DA-ref but on a very small dataset, not comparable to the one used in our investigation.

As regards the comparative analysis of DA and PE, correlation results between DA-ref and HTER for 9 language directions are presented in [19]. However, the evaluation data differs in many respects, making results not comparable. First, the dataset used in this paper includes both DA-ref and DA-src. Furthermore, PE data is made of multiple bilingual post-edits created by professional translators native in the target language and working in their professional CAT environment. On the contrary, the post-edits used to calculate HTER in [19] were created through monolingual post-editing, probably based on the same reference used to collect DA-ref judgments.

3. Evaluation Data

To perform our investigations on DA and PE we relied on the Human Evaluation dataset created as part of the IWSLT 2017 evaluation campaign [5]. The resource is publicly available at the WIT[3] website [20], where all IWSLT data and tools are released by the organizers of the campaign. [1]

The dataset is based on TED talks[2] and includes 603 sentences (around 10,000 source words), corresponding to the first half of ten different TED talks. It covers two language pairs, namely Dutch-German ($NlDe$) and Romanian-Italian ($RoIt$) which – belonging to two distinct families (West-Germanic and Romance, respectively) – show rather different characteristics.

For each language direction, evaluation data were collected for nine different state-of-the-art neural MT systems: three standard *bilingual* systems (*i.e.* a different system is created for each language direction) and six *multilingual* systems (*i.e.* one single system for multiple language directions), out of which three in the *zero-shot* condition (*i.e.* tested on language pairs that are not present in the training data). Furthermore, systems differ also for their architecture, since some of them implement *Recurrent Neural Networks*, while others are based on the *Transformer* model [21].[3]

[1] https://wit3.fbk.eu/show.php?release=2017-02&page=subjeval&texthead=Evaluation%20Data

[2] www.ted.com

[3] All details about the MT systems can be found in [22, 23, 24].

The MT systems were evaluated on all the 603 dataset sentences according to PE, source-based DA, and reference-based DA. Details on human evaluation data are given in the following.

3.1. Post-Editing data

This evaluation was carried out through *bilingual* post-editing: the outputs of the nine MT systems on the 603 test sentences were assigned to nine professional translators to be manually corrected directly according to the source sentence.

To ensure the soundness of the evaluation and cope with translators' variability, an equal number of outputs from each MT system was assigned randomly to each translator, in such a way that each translator had to post-edit all the sentences in the test set but only once.

The resulting PE data used in this study consists of nine new reference translations for each sentence of the test set. Each one of these references represents the *targeted reference* of the system output from which it was derived, while the post-edits of the other systems are available for evaluation as additional references. All details about data preparation and post-editing can be found in [2, 5].

In addition to the PE data, an external - independently created - reference was also available, for a total of ten references for each of the 603 sentences in the dataset.

3.2. Direct Assessment data

Both DA-src and DA-ref data were collected for all the MT system outputs on all the 603 test sentences employing bilingual linguists. To ensure the reliability of the human assessments, part of the collected data was used for quality control. Based on artificially degraded translation output—which should be scored worse than the corresponding candidate translation—it is possible to identify users who randomly assign scores without paying attention to the presented data and, thus, work unreliably. Only annotations from reliable annotators were used to compute the final system evaluation. Furthermore, as annotators may have different annotation behaviour, the collected scores (at least two for each sentence) were standardized into *z scores*, which capture the number of standard deviations a score is different from (*i.e.* better or worse than) the respective annotator's mean score. Then, z scores were averaged at segment and system level to determine the overall MT system quality as observed by all annotators.

4. Analysis of PE-based evaluation

As described in Section 1, evaluation via post-editing is based on TER, which measures the amount of editing that a human would have to perform to change an automatic translation so that it exactly matches a given reference translation. Since TER is an automatic metric that works on exact word matching, it is unable to distinguish differences between MT output and reference due to normal linguistic variation from those due to real MT errors.

For this reason the reference translations used in TER-based evaluation (as in all automatic evaluations) play a central role in determining its reliability and informativeness.

It is widely accepted that the most suitable reference to evaluate an MT system is its corresponding post-edit (targeted reference), since it is derived from that specific system and thus should differ from the MT output only with respect to the parts of it that are incorrect. External references are at the other hand of the spectrum, since they are manually generated by translating the source text from scratch, independently from any MT system output. A particular case of reference is the post-edit of an actual system output which is not the one under evaluation. In this case the reference represents one of the many possible translation options and can indeed differ from the evaluated MT output due to linguistic variation. However, being created starting from an MT output, it is possible that its peculiar features make it more suitable to MT evaluation. This type of reference is particularly interesting since it can be easily gathered, being a natural by-product of professional translation in the CAT framework. Finally, the usage of multiple references has often been investigated as a way to address the issue of acceptable linguistic variation, under the assumption that the more references the highest the reliability of the evaluation.

In this section we exploited the PE data – *i.e.* one external reference and nine post-edits created from the nine evaluated MT systems – to carry out different analyses aimed at understanding if and how TER-based evaluation is sensitive to the type and number of references used.

Depending on the reference(s) used in the analysis, we relied on different variants of TER, namely: *(i) Human-targeted TER* (HTER), where TER is computed between the machine translation and its post-edited version (targeted reference); *(ii) Multiple reference TER* (mTER), where TER is computed against the closest reference – *i.e.* the one which minimizes the number of edits – among all the available ones.

We empirically analyzed the impact of references in the evaluation from two different angles: *(i)* for each evaluated MT system, we investigated the specific contribution of each of the nine available post-edits to the mTER score of the system; *(ii)* for each language pair, we calculated how overall MT system performance (*i.e.* TER score) varies depending on the type and number of references used.

Figure 1 shows an example of the distribution of the identity of systems which originated the post-edits that were chosen as closest reference translation in the computation of mTER. Four *NlDe* systems are presented in the figure, among which three were post-edited (BL.lab1, SD.lab2, ZS.lab3) and one was not (SD.lab4), and is shown for comparison purposes. The same behaviour of the *NlDe* systems presented in the figure was observed also for the other *NlDe* systems as well as for the *RoIt* direction.

As expected, the peak occurs in correspondence of the post-edit of the system under evaluation. Looking at the cor-

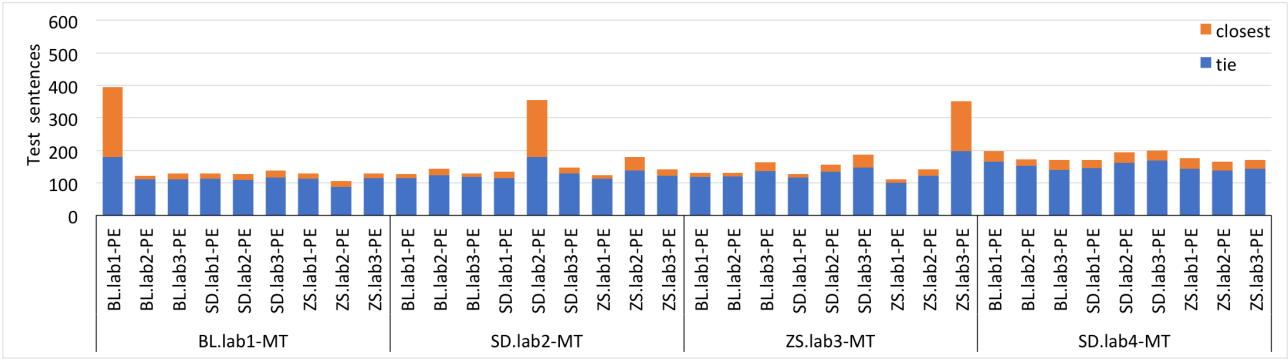


Figure 1: Frequency distribution of the closest PE selected in the computation of mTER of four *NlDe* systems. For each system that originated the PE, in orange the number of sentences for which that PE was the closest translation to the MT under investigation, in blue the number of sentences where the PE was the closest together with at least another PE.

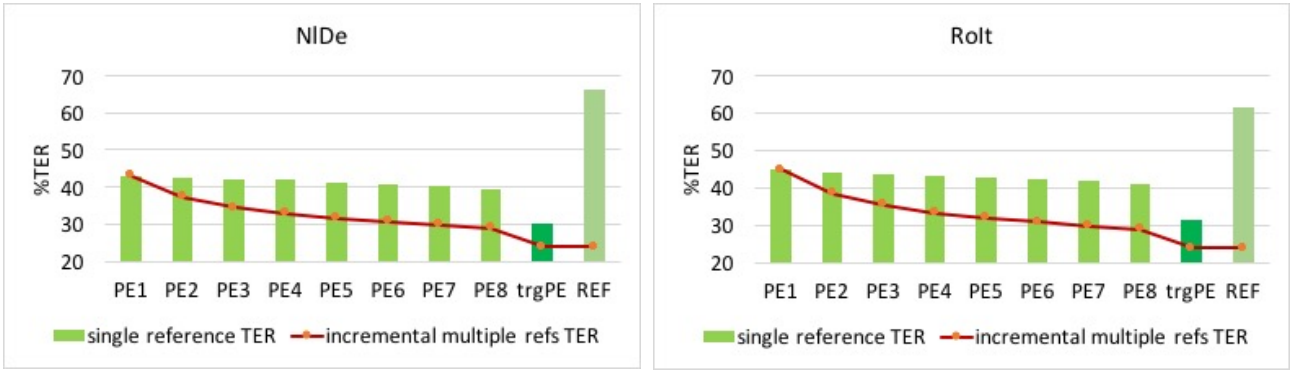


Figure 2: TERs on single references (green bars) and mTER on increasing number of references (red line).

responding column, we note however that the targeted reference is the closest to the MT output only for around one-third of the test set (orange-coloured), while for another third there is at least one equivalent post-edit from another system (blue-coloured). Interestingly enough, for the remaining third of the test set, the closest reference is a post-edit from another system.

Looking at the columns of the post-edits originated from the other 8 MT systems, we see that for a non-negligible number of test sentences these references represent the closest translation (orange). This is particularly relevant when confronted to the results of the external reference translation, which is not shown in the figure since it was never picked as the closest reference translation. It is also worthwhile to note that the post-edits of other MT systems created by the same Lab – which are expected to have similar outputs – are not chosen as closest references significantly more often than the post-edits of other Labs' systems. This suggests that the advantage of the post-edits of other systems does not rely in the similarity of the MT systems but more generally in the fact that the reference translation is derived from an MT output.

From the point of view of the number of references used in the evaluation, we understand from Figure 1 that a certain degree of variability is present also in the targeted translation – since for one-third of the test set it does not ensure the lowest edit distance with the MT output. We can thus confirm that – even when a targeted reference is available – mTER guarantees the highest reliability of the evaluation. Finally, the rightmost part of Figure 1 presents results for a system (SD.lab4) for which no post-edit was created. We can see that the closest references are equally distributed among all the available references, further confirming the importance of having multiple references.

The same conclusions can be drawn by analyzing the overall performances of the MT systems when using different reference translations. For each language direction, Figure 2 shows the impact that each of the ten references at our disposal has on TER, averaged across systems. The vertical bars provide the TER score computed using a single reference, be it one of the external post-edits, the targeted post-edit, or the external reference; for each system, the PEs are considered in reverse order with respect to their overall score, that is from the farthest to the closest to the system output, which invariably is the targeted PE; the external reference is presented as the last; the red line represents the mTER computed on an incremental set of references.

The low TER results obtained using a single non-targeted post-edit are quite interesting. Indeed evaluating a system

against a post-edit created for another system is more sound than using an external reference. This is particularly relevant in a real application scenario where obtaining a post-edit of a system is easy and inexpensive. On the same line, considering the mTER cumulative score, it is interesting to see that the same HTER results obtained with the targeted reference (trgPE, dark green bar) can be achieved using seven external post-edits for the *NlDe* direction and six for the *RoIt* direction.

For completeness, Table 1 gives the exact figures of the most relevant information contained in Figure 2, namely mTER using all 9 available post-edits, HTER, and TER over the external reference.

Indeed we can observe a considerable TER reduction when using all collected post-edits with respect to both the HTER obtained using the targeted post-edit and the TER obtained using the independent reference. This reduction clearly confirms that exploiting all the available reference translations allows to produce a score which is not only more reliable but also more informative about the real performance of the systems.

	mTER 9 PE refs	**HTER** tgt PE	**TER** 1 ext ref
NlDe	23.80	29.96	66.10
RoIt	23.64	31.25	61.56

Table 1: %TERs computed on different (set of) references.

5. Comparative analysis of DA-based and PE-based evaluation

As introduced in Section 1, the DA-based and PE-based evaluation tasks focus on different aspects of automatic translation: general quality for the reader and usefulness for translator, respectively. To investigate the extent to which PE and DA lead to similar results, for each evaluated system we calculated the Pearson correlation between PE-based scores and DA-based scores for each sentence in the test set. The correlation results obtained for each system were then averaged through the Fisher transformations suggested in [25].

Table 2 presents the average correlation results. Correlations are calculated for both DA-src and DA-ref and for all the metrics investigated for PE-based evaluation, namely mTER, HTER and TER.

As expected, correlation is good, that is, in general segments judged as poor by DA annotators (low DA scores) also need substantial post-editing (high PE scores) or vice-versa.

Results slightly vary across language directions, but the same trends can be observed. First, the highest correlation is found between DA-src and mTER, confirming that these are the two most highly reliable human evaluation measures. As regards PE, mTER correlates better than HTER with DA, showing once again the importance of having multiple references. As regards DA, correlation with PE is considerably

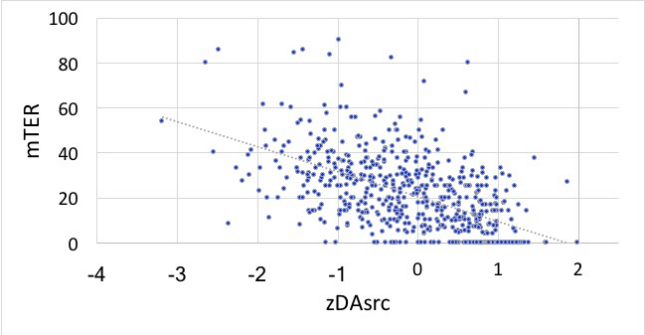


Figure 3: ZS.lab3 *RoIt* system: scatter plot of source-based DA standardized scores and mTER scores.

higher for DA-src than DA-ref. This indicates that the so-called reference-bias affects not only automatic metrics but also DA-based human evaluation. This is further confirmed by the results obtained for TER, which is calculated on the same external reference translation used in DA-ref. Although TER correlation scores are very low, TER correlates much better with DA-ref than DA-src, showing an opposite behaviour with respect to mTER and HTER.

Given the correlation results obtained, we carried out a further analysis to investigate whether having both evaluations can help improving the evaluation quality, *i.e.* whether the two methodologies can complement each other's weaknesses.

Figure 3 shows the scatter plot of the correlation between mTER and DA-src for one of the investigated *RoIt* systems (r=–0.5812). Conflicting evaluations appear in the lower-left and upper-right quadrants of the scatter plot. The first quadrant includes segments which resulted good according to PE evaluation (low PE scores) but were judged as poor by DA annotators (low DA scores); the second includes segments which needed substantial post-editing (high PE scores) but were judged as good by DA annotators (high DA scores).

Conflicting evaluation cases are particularly relevant since PE is known to be more informative (see Section 1), but DA could identify issues that PE-based evaluation cannot spot. We manually inspected a sample of the sentences with conflicting evaluations and we found some interesting patterns. Examples are provided in Table 3.

When PE scores are low (*i.e.* few edits are needed to correct the MT output) but the translation is bad according to DA, typically the sentence contains few but crucial errors, which make it difficult to understand the meaning of the sentence (see Example 1 in the table). In these cases, the conflict is not solvable since from the point of view of DA - which is focused on adequacy - the MT output is rightfully not good, while from the point of view of the translator who has access to the source sentence, the MT output is indeed useful to speed-up translation.

In the opposite situation, *i.e.* high PE scores but good translation according to DA, we have two main causes for

avg(**r**)		*NlDe*			*RoIt*		
		mTER	HTER	TER	mTER	HTER	TER
zDA	src	-0.5466	-0.4796	-0.1918	-0.5294	-0.4306	-0.2137
	ref	-0.4491	-0.4100	-0.3579	-0.4524	-0.3882	-0.3570

Table 2: Average (DA,PE) correlations across systems.

			mTER	DA-src (abs)
1.	SRC	Nu are flapsuri, balamale, eleroane, actuatoare sau alte suprafee de control, doar o simplă elice. *It has no flaps, no hinges, no ailerons, no actuators, no other control surfaces, just a simple propeller.*		
	MT	Non ha **fiori**, **balconi**, **elenchi**, attuatori o altre superfici di controllo, solo una semplice elica. *It has no flowers, no balconies, no lists, no actuators, no other control surfaces, just a simple propeller.*	14.43%	28
	PE	Non ha **flaps**, **cerniere**, **alettoni**, attuatori o altre superfici di controllo, solo una semplice elica.		
2.	SRC	Prietenele mele, feministe convinse, au fost șocate. *My [female] friends, committed feminist, were aghast*		
	MT	**I miei** amic**i**, femministe convint**i**, sono rimasti scioccat**i**. *My [male] friends, committed feminist, were aghast.*	47.87%	88
	PE	**Le mie** ami**che**, femministe convint**e**, sono rimasti scioccat**e**.		

Table 3: *RoIt* language direction. Examples of conflicting DA-PE evaluation.

conflicts. First, we found very short or long sentences which are indeed good translations but the mTER score was not correct due to tokenization (and consequently alignment) problems. These cases highlight the main weakness of PE-based evaluation, namely the fact that it relies on automatic metrics to compute the edit distance. The other type of conflict (see Example 2 in the table) regards those segments that have to be heavily post-edited for amending errors which do not alter the overall comprehension, like in chains of morphological errors. In these cases, the MT errors affect more fluency than adequacy, to which DA-based assessment is less sensitive.

6. Conclusions

In order to shed light on the properties, strengths and weaknesses of human evaluation it is crucial to rely on high quality datasets. The specific characteristics of the IWSLT-17 Human Evaluation dataset used in this investigation - size, variety and high quality of the three-way human evaluation - ensured sound empirical analyses and generalizable outcomes. The main findings of this paper are summarized in the following.

Analysis on PE evaluation data:

- the targeted reference is the closest to the MT output only for one-third of the test sentences. Thus, mTER guarantees the highest reliability of the evaluation over HTER;
- evaluating a system against a post-edit created for another system is more sound than using an external reference, independently from the similarity of the two MT systems;
- the same results obtained with the targeted reference (HTER) can be achieved using six/seven external post-edits (mTER), not including the targeted reference.

Comparative analysis of DA and PE:

- the highest correlation is found between DA-src and mTER, confirming that these are the two most highly reliable human evaluation measures;
- correlation with PE is considerably stronger for DA-src than DA-ref. This indicates that the so-called reference-bias affects not only automatic metrics but also DA-based human evaluation;
- conflicting evaluations between DA-src and mTER exist. In some cases DA-src can help mitigate the weakness of PE which depends on its automatic component. In other cases conflicts are caused by inherent differences due to the fact that the two evaluation methods address different human perspectives.

To conclude, we are planning to extend our research on both the analyses presented in this paper. First, we will further verify and generalize the results obtained on PE data by carrying out the analyses on other publicly available IWSLT datasets, which include multiple post-edits for other language directions such as English-German, English-French, and Vietnamese-English. Second, we will compare more deeply how DA-ref and DA-src behave on the same data. Finally, we will perform the manual analysis also on *NlDe* data.

7. References

[1] 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, pp. 223–231.

[2] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, and M. Federico, "Report on the 10th IWSLT evaluation campaign," in *Proc. of IWSLT*, Heidelberg, Germany, 2013.

[3] Y. Graham, T. Baldwin, A. Moffat, and J. Zobel, "Continuous measurement scales in human evaluation of machine translation," in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Association for Computational Linguistics, 2013, pp. 33–41. [Online]. Available: http://www.aclweb.org/anthology/W13-2305

[4] 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 *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 169–214. [Online]. Available: http://www.aclweb.org/anthology/W17-4717

[5] 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 *Proc. of IWSLT*, Tokyo, Japan, 2017.

[6] M. Denkowski and A. Lavie, "Choosing the right evaluation for machine translation: An examination of annotator and automatic metric performance on human judgment tasks," in *Proceedings of the 9th Conference of the Association of Machine Translation in the Americas (AMTA)*, Denver, CO, USA, 2010.

[7] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico, "Neural versus phrase-based machine translation quality: a case study," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 257–267. [Online]. Available: https://aclweb.org/anthology/D16-1025

[8] ——, "Neural versus phrase-based mt quality: An in-depth analysis on english-german and english-french," *Computer Speech Language*, vol. 49, pp. 52 – 70, 2018.

[9] M. Snover, N. Madnani, B. J. Dorr, and R. Schwartz, "Fluency, adequacy, or hter?: exploring different human judgments with a tunable mt metric," in *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2009, pp. 259–268.

[10] 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.

[11] 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.

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

[13] 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 *WMT@EMNLP*, 2015.

[14] Y. Graham, T. Baldwin, A. Moffat, and J. Zobel, "Is machine translation getting better over time?" in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014, pp. 443–451.

[15] H. Hassan, A. Aue, C. Chen, V. Chowdhary, J. Clark, C. Federmann, X. Huang, M. Junczys-Dowmunt, W. Lewis, M. Li, S. Liu, T. Liu, R. Luo, A. Menezes, T. Qin, F. Seide, X. Tan, F. Tian, L. Wu, S. Wu, Y. Xia, D. Zhang, Z. Zhang, and M. Zhou, "Achieving human parity on automatic chinese to english news translation," *CoRR*, vol. abs/1803.05567, 2018. [Online]. Available: http://arxiv.org/abs/1803.05567

[16] A. Lommel, "Blues for bleu: Reconsidering the validity of reference-based mt evaluation," in *Proceedings of the LREC 2016 Workshop Translation Evaluation-From Fragmented Tools and Data Sets to an Integrated Ecosystem*, 2016.

[17] M. Fomicheva and L. Specia, "Reference bias in monolingual machine translation evaluation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2016, pp. 77–82. [Online]. Available: http://www.aclweb.org/anthology/P16-2013

[18] Q. Ma, Y. Graham, T. Baldwin, and Q. Liu, "Further investigation into reference bias in monolingual evaluation of machine translation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 2476–2485. [Online]. Available: http://aclweb.org/anthology/D17-1262

[19] Y. Graham, T. Baldwin, M. Dowling, M. Eskevich, T. Lynn, and L. Tounsi, "Is all that glitters in machine translation quality estimation really gold?"

in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, 2016, pp. 3124–3134. [Online]. Available: http://www.aclweb.org/anthology/C16-1294

[20] M. Cettolo, C. Girardi, and M. Federico, "WIT^3: Web inventory of transcribed and translated talks," in *Proc. of EAMT*, Trento, Italy, May 2012. [Online]. Available: http://hltshare.fbk.eu/EAMT2012/html/Papers/59.pdf

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

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

[23] 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.

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

[25] D. M. Corey, W. P. Dunlap, and M. J. Burke, "Averaging correlations: Expected values and bias in combined Pearson *r*s and Fisher's *z* transformations," *The Journal of General Psychology*, vol. 125(3):245-261, 1998.

The USTC-NEL Speech Translation system at IWSLT 2018

Dan Liu[1,2], *Junhua Liu*[1,2], *Wu Guo*[1]
Shifu Xiong[2], *Zhiqiang Ma*[2], *Rui Song*[2], *Chongliang Wu*[2], *Quan Liu*[2]

University of Science and Technology of China [1]
IFLYTEK Co. LTD. [2]

{danliu,jhliu}@mail.ustc.edu.cn wuguo@ustc.edu.cn
{danliu, jhliu, sfxiong, zqma2, ruisong, clwu4, quanliu}@iflytek.com

Abstract

This paper describes the USTC-NEL (short for "National Engineering Laboratory for Speech and Language Information Processing University of science and technology of china") system to the speech translation task of the IWSLT Evaluation 2018. The system is a conventional pipeline system which contains 3 modules: speech recognition, post-processing and machine translation. We train a group of hybrid-HMM models for our speech recognition, and for machine translation we train transformer based neural machine translation models with speech recognition output style text as input. Experiments conducted on the IWSLT 2018 task indicate that, compared to baseline system from KIT, our system achieved 14.9 BLEU improvement.

1. Introduction

Conventional speech translation systems consist of three components: source-language automatic speech recognition (ASR), post-processing over ASR outputs, and source-to-target text translation. This pipeline system suffers from error accumulation, which means speech recognition and translation models trained separately may perform well individually, but do not work well together because their error surface do not compose well [1].

In the most recent years, end-to-end speech translation based on encoder-decoder with attention mechanisms has been very promising for reducing accumulated errors [2, 1, 3]. However, parallel speech data is much smaller than those available to train text-based machine translation (MT) systems, particularly neural systems that needs to learn a relatively large parameters. As a result, an end-to-end speech translation system can often outperform pipeline systems with same training data, but is hard to beat pipeline system with dozens of training data [1].

In addition, to translate very long speech (e.g. translate a full talk), an end-to-end system must rely on voice activity detection (VAD) method to split raw audio into sentence-like fragments, in which mis-segmented sentence fragments are very likely to cause serious translation errors. Therefore, for pipeline systems, sentence re-segmentation based on ASR results may be done in post-processing step, which can improve performance significantly [4].

To reduce the error accumulation of pipeline systems, we introduce a data augmentation based solution to train translation model with ASR results as source directly, instead of normalize ASR results (e.g. insert punctuations, normalization for case, numerals, etc.) in post-processing. Text normalization cannot bring any new information, it just produces texts that translation system likes, and this may lead to additional errors. In our experiments, the data augmentation based solution performs significantly better than pipeline system with text normalization and end-to-end speech translation system.

This paper is organized as follows. We first describe the processing for speech and text training data in Section 2, following is our full system and training details. Our experiments are presented in Section 4.

2. Data Processing

We conduct experiments on IWSLT speech translation task [5] from English to German. All experiments were performed under requirements of IWSLT 2018 evaluation campaign speech translation task. The training data for speech recognition and translation after filtering are listed in Table 1 and Table 2.

Table 1: *speech training data.*

Corpus	# of seg.	Speech hours
TED LIUM2 [6]	92976	207h
Speech Translation	171121	272h

2.1. speech recognition training data

The speech data contains TED LIUM2 [6] and speech translation data by IWSLT evaluation campaign. In TED LIUM2, only raw wave files and manual transcriptions (without punctuation) were offered. And in speech translation data, raw wave files, English transcriptions and the corresponding German translations were offered, but some transcriptions is not

Table 2: *text training data.*

Corpus	raw	filtered
commoncrawl	2.39M	1.80M
rapid	1.32M	1.00M
europal	1.92M	1.81M
commentary	0.284M	0.233M
paracrawl	36.35M	12.35M
opensubtitles	22.51M	14.24M
WIT3(in domain)	0.209M	0.207M

match to there corresponding audio. Besides this, about 166 hours of audio in speech translation data were not labeled, we regard them as unsupervised data.

To utilize those data, we firstly train initial acoustic model based on TED LIUM2. Using this model, we do force alignment on IWSLT speech translation data, and discard utterances with significantly abnormal scores. After this process, the supervised data size of IWSLT has been reduced to 246 hours from 272 hours. Meanwhile, the unsupervised data is recognized by our initial model and filtered based on ASR confidence to expand the training set.

To further increase the amount of data in the training set, we perform data augmentation by noise and speed perturbations. For each speech signal, a noise version is created initially. Speed perturbation is then performed on the raw signals with speed factors 0.8 and 1.2. Eventually, up to (207+246+166)*4 hours of data may be used.

2.2. speech translation training data

The speech translation training data is the same as the speech recognition training data. The target references for LIUM2 and unsupervised data are generated by our best text machine translation system.

2.3. text translation training data

The text translation training data contains parallel data and monolingual training data. As for parallel data, we use all of the allowed training data for Speech Translation Task which includes TED corpus, data provided by WMT 2018 and OpenSubtitles2018 [7]. The data is pre-processed before training and translation. Sentences longer than 100 words and duplicated sentence pairs are removed. Also, numbers are normalized in order to match the ASR outputs. NMT systems are more vulnerable to noisy training data, rare occurrences in the data, and the training data quality in general. So we measure the cross-lingual similarities between source and target sentences, and then reject sentences with similarity below a specified threshold. After filtering, we can get relevant and high quality data. The training data after filtering are listed in Table 2.

As for monolingual training data provided by WMT 2018, we clean the noisy data for English and German, and then we use the supervised convolutional neural network method [8] to select monolingual training data that are close to the TED domain. After this processes, we select 91M monolingual English data and 43M mono-lingual German data for language model training.

3. System Description

3.1. speech recognition

The primary system of our speech recognition is a hybrid-HMM system. The acoustic model contains multiple deep neural networks based on CNN and LSTM structure. State level posterior fusion technique is used for the final ASR results. The details of model structure and training criterion are as following:

1. DenseNet [9]: DenseNet with 13 dense connection blocks and 3 max-pooling steps with stride 2 on both time and frequency domain, trained with cross-entropy (CE) and sequence-discriminative training (SDT) criterion [10].

2. BiLSTM [11]: 3 layers BiLSTM network trained with CE and SDT criterion.

3. CLDNN [12]: CNN-BiLSTM-DNN structure trained with CE and SDT criterion.

The language models are trained on English monolingual data described in Section 2.3. The first-pass decoding is performed with the HMM and 3-gram LM. A 4-gram LM is used for second-pass decoding and followed by a LSTM-based LM.

In this task we should do speech recognition on full talk, so we have to split the raw audio into sentence-like pieces for speech recognition. We do speech segmentation with LSTM based VAD model, which is trained on TED LIUM2 dataset with speech/nonspeech labels extracted by force alignment with our hybrid-HMM model.

3.2. post-processing vs data augmentation

It has been shown that post-processing is crucial for achieving good speech translation performance [4], this comes from two aspects. First, segmentation boundaries for ASR are based on VAD, which inevitably leads to fragments with incomplete semantics, and sentence re-segmentation based on ASR results is needed. Second, translation models are trained with written text as input, which means text normalization of ASR results is essential for conventional systems.

We know punctuations may contain rich semantic information, but in post-processing for speech translation, punctuations are only generated from ASR output word sequences. In this case, these punctuations can not bring more information than words. The main goal of post-processing is just to produce text suitable for machine translation. However, it should be noted that errors in punctuation prediction may be propagated in machine translation process.

Here we introduce a new solution with respect to mismatch between ASR results and machine translation inputs. Instead of transform ASR results to written text on decoding step, we transform the source text for machine translation training data into the style of ASR results on training step. The difficulty of normalizing ASR results to written text seems equal to the difficulty of normalizing written text to ASR results. However, data augmentation with fake ASR results for machine translation is more robust for errors compared to text normalization on decoding step.

We train a neural machine translation (NMT) model to translate written text into ASR results. To build the training data, we process the English written data by rule (remove punctuations, lower case and translate Arabic numerals into English words), the generated text is similar to ASR results except for recognition errors. We also build real data with the ASR results and source written texts provided in speech translation dataset. The NMT model from written text to ASR results are trained on these two dataset and fine-tuned on only real data. This model may generate ASR output style text with common ASR errors. And we augment the text machine translation dataset by translating the source written texts into ASR output style texts. As a comparison, we also trained an inverted NMT for text normalization.

The data augmented based solution can translate directly from ASR result, which reduces errors caused by text normalization. Besides this, our model has the ability to tolerate common recognition errors. E.g., our ASR system may mistake "two" to "to" in some special contexts, and our NMT system may translate "top to percent" to "top zwei Prozent".

Sentence re-segmentation are still important to speech translation system, because training data for machine translation are all semantically complete sentences. Data augmentation with semantic incomplete sentence fragments may suffer from reordering between source and target language. So we train a LSTM based model to re-segmented sentences based only on text infomation. This model is trained on TED and OpenSubtitle dataset, with one whole paragraph as input, and the punctuation ".!?" as sentence boundaries.

3.3. machine translation

3.3.1. text machine translation

Transformer [13] is adopted as our baseline, all experiments use the following hyper-paramter settings based on Tensor2Tensor transformer_relative_big settings [1]. This corresponds to a 6-layer transformer with a model size of 1024, a feed forward network size of 8192, and 16 heads relative attention. Model is trained on the full dataset described in Section 2.3 and fine-tuned on speech translation dataset. We trained both conventional NMT model and NMT model with augmented data described in Section 3.2.

[1] https://github.com/tensorflow/tensor2tensor/tree/v1.6.3

3.3.2. end-to-end speech translation

For our end-to-end speech translation model, DenseNet described in Section 3.1 followed by one BiLSTM layer is employed as encoder, and the decoder is same as transformer model in Section 3.3.1. It is difficult to train speech translation model from random initialization parameters, for reordering between source and target language are difficult to align with frame based speech representations. Pre-training with speech recognition task significantly improves the performance. And this encoder-decoder based ASR model is used for rescoring our final ASR results.

End-to-end speech translation system has no chance to re-segment sentences. We found splicing audio segments acquired by VAD may improve the translation performance, but still has a significant gap to performance based on sentence re-segmentation.

4. Experimental Results

In this section, we present a summary of our experiments for the IWSLT 2018 speech translation evaluation task. We test WER (word error rate) for our speech recognition system on dev2010, which is the only dataset with CTM format transcriptions. And we test our speech translation systems on IWSLT dev2010, tst2010, tst2013, tst2014 and tst2015. Case sensitive BLEU based on realigning system outputs to reference by minimizing WER [14] is used for our speech translation evaluation metric.

4.1. Results of Speech Recognition

In this section, we demonstrate the results of our ASR system. The acoustic model of our primary system is the deep CNN model, and we decode with 3-gram for first-pass decoding and 4-gram for second-pass. We test our performance in dev2010. First, we compare the impact of training data in Table 3. Here "spv." represents supervised data, "usv." represents unsupervised data and "spd." represents speed disturbed data. As show in Table 4, by training with noisy data, the WER is relatively reduced by 7.32%.

Table 3: *WER for speech recognition with different training data on dev2010*

Training Data	WER
spv.	9.7
noisy spv.	8.99
noisy spv. usv.	8.92
noisy spd. spv. usv.	8.86

Based on the above results, we train three acoustic models with different structures. Further promotion is achieved by fusing multiple acoustic models, rescoring with RNN-LM. We also test the encoder-decoder based speech recognition model described in Section 3.3.2, which performs significantly worse than our hybrid-HMM systems. But rescoring

with encoder-decoder system brings a small improvement. Details are showed in Table 4.

Table 4: *Results of fusion of different models for speech recognition on dev2010* .

DenseNet	8.86
BiLSTM	8.72
CLDNN	8.40
Encoder-Decoder	14.64
DenseNet +BiLSTM + CLDNN	8.22
+ RNN	7.61
+Encoder-Decoder	7.3

4.2. Results of End-to-end Speech Translation

In this section, we describe our experiments on end-to-end speech translation. The average BLEU score of our baseline end-to-end speech translation system is 20.50, which is significantly worse than our pipeline system (Tabel 6). The degradation comes from two aspects. Firstly, our encoder-decoder speech recognition performs worse than baseline speech recognition system (WER 7.61% to14.64%). Secondly, the end-to-end system has no chance to re-segment sentences based on source recognition results.

To reduce the influence of incomplete sentence fragments caused by VAD, we splice the VAD fragments to at least 10 seconds, which brings the improvements of about 1 BLEU. For comparison, we present the performance of a system that re-segment audio based on speech recognition results, which brings another 1.3 BLEU gain, but this is not a "end-to-end" system. At last, the ensemble of 4 different models improves about 1 BLEU compared to corresponding single model. The details are showed in Table 5.

4.3. Results of Pipeline Speech Translation

In this section, experiments are all based on the best ASR results described in Section 4.1. At test time, we use a beam size of 80 and a length penalty of 0.6. All data used for training are described in Section 2. All reported scores are computed using IWSLT speech translation evaluation metric.

4.3.1. post processing

The post processing procedure.includes two parts: sentence re-segmentation and text normalization. And we introduced one data augmentation based solution to remove text normalization. We compare the performance for different solutions in Table 6.

We see sentence re-segmentation has a huge impact on performance. Since sentence-like pieces obtained by VAD do not carry any semantic information, it is very unfavorable for machine translation. Other than this, our data augmentation based solution achieves a average BLEU score of 28.76, 1.3 BLEU higher over system with post processing. And we found the models with text regularization and data augmentation can be combined to get better results.

4.3.2. fusion of different models

We train 3 groups of different models, one for text regularization and two for data augmentation (L2R and R2L, which denotes the target order left to right and right to left). For each group we train 4 models with different initialized parameters, and decoded with the ensemble models to get 80-best hypotheses with beam size of 80. The 3 groups of hypotheses are merged and rescored by all translation models, target language model and end-to-end speech translation model. Performances are shown in Table 7.

4.4. Submission Results

We submitted 3 systems for speech translation task. The primary system is the best fusion system demonstrated at row 7 in Table 7, and the contrastive systems are all based on encoder-decoder model from audio features. Contrastive0 is based on sentence re-segmentation with source speech recognition results, which is not real "end-to-end", while contrastive2 is real end-to-end systems with only single model.

We compared our submitted systems to KIT speech translation system (noted as "Baseline_KIT")[2], which is the baseline system provided by KIT, performance is shown in Table 7. Our primary system achieves a average BLEU of 30.26, which is 14.9 BLEU higher than baseline from KIT.

5. Conclusion

In this paper we presented our speech translation systems for IWSLT 2018 evalution. Our results indicated that the end-to-end system still performs significantly worse than the conventional pipeline system, and NMT with data augmentation performs better than solutions with text regularization. Our best ensemble system achieved 14.9 BLEU improvement compared to baseline system from KIT.

6. References

[1] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

[2] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn, "An attentional model for speech translation without transcription," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 949–959.

[3] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech transla-

[2]https://github.com/jniehues-kit/SLT.KIT

Table 5: *BLEU scores for end-to-end speech translation .*

system	dev2010	tst2010	tst2013	tst2014	tst2015	average
VAD	21.45	21.41	21.76	20.06	17.83	20.50
splice 10s	22.14	22.16	22.76	21.00	19.52	21.52
re-segment	23.79	24.18	24.18	22.22	20.07	22.89
ensemble(splice)	23.43	22.97	23.58	21.96	20.67	22.52
ensemble(re-segment)	24.78	24.92	25.41	23.23	21.01	23.87

Table 6: *BLEU scores for pipeline speech translation system*

re-segment	text regularization	data augmentation	dev2010	tst2010	tst2013	tst2014	tst2015	average
N	Y	N	26.47	27.71	28.04	25.65	24.00	26.37
N	N	Y	27.58	28.26	29.81	26.79	25.65	27.62
Y	Y	N	27.75	28.90	29.01	26.88	24.52	27.41
Y	N	Y	28.98	29.98	30.69	28.19	25.99	28.76

Table 7: *BLEU scores for fusion systems*

system	dev2010	tst2010	tst2013	tst2014	tst2015	average
text normalization	28.64	29.41	29.59	27.37	25.13	28.03
augment L2R	29.45	30.01	30.78	28.37	26.14	28.95
augment R2L	28.42	29.58	30.88	27.98	26.47	28.66
fusion	30.28	31.01	32.28	29.38	27.40	30.07
+target LM	30.30	31.00	32.37	29.44	28.14	30.25
+e2e model	30.50	31.06	32.31	29.35	28.06	30.26

Table 8: *performance of submitted systems*

system	end2end	single model	dev2010	test2010	test2013	test2014	test2015	average
Baseline_KIT	N	Y	17.07	12.37	16.59	15.42	15.15	15.32
PRIMARY	N	N	30.50	31.06	32.31	29.35	28.06	30.26
Contrastive0	N	N	24.78	24.92	25.41	23.23	21.01	23.87
Contrastive2	Y	Y	22.14	22.16	22.76	21.00	19.52	21.52

tion of audiobooks," *arXiv preprint arXiv:1802.04200*, 2018.

[4] 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.

[5] C. Mauro, F. Marcello, B. Luisa, N. Jan, S. Sebastian, S. Katsuitho, Y. Koichiro, and F. Christian, "Overview of the iwslt 2017 evaluation campaign," in *International Workshop on Spoken Language Translation*, 2017, pp. 2–14.

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

[7] P. Lison, J. Tiedemann, and M. Kouylekov, "Opensubtitles 2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora," in *Proceedings of the 11th edition of the Language Resources and Evaluation Conference (LREC 2018), Miyazaki, Japan.(accepted)*, 2018.

[8] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *CVPR*, vol. 1, no. 2, 2017, p. 3.

[10] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Interspeech*, 2013, pp. 2345–2349.

[11] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 273–278.

[12] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4580–4584.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[14] E. Matusov, G. Leusch, O. Bender, and H. Ney, "Evaluating machine translation output with automatic sentence segmentation," in *International Workshop on Spoken Language Translation (IWSLT) 2005*, 2005.

The ADAPT System Description for the IWSLT 2018 Basque to English Translation Task

Alberto Poncelas, Andy Way
ADAPT Centre, School of Computing,
Dublin City University, Dublin, Ireland
{firstname.lastname}@adaptcentre.ie

Kepa Sarasola
Ixa Group (UPV/EHU), Faculty of Informatics
University of the Basque Country
kepa.sarasola@ehu.eus

Abstract

In this paper we present the ADAPT system built for the Basque to English Low Resource MT Evaluation Campaign. Basque is a low-resourced, morphologically-rich language. This poses a challenge for Neural Machine Translation models which usually achieve better performance when trained with large sets of data.

Accordingly, we used synthetic data to improve the translation quality produced by a model built using only authentic data. Our proposal uses back-translated data to: (a) create new sentences, so the system can be trained with more data; and (b) translate sentences that are close to the test set, so the model can be fine-tuned to the document to be translated.

1. Introduction

We participated in the Basque to English Low Resource MT Evaluation Campaign as part of the International Workshop on Spoken Language Translation (IWSLT) 2018. In this task, we aimed to build an MT model to translate subtitles of TED (Technology, Entertainment, Design) talks from Basque into English.

Basque (or Euskera), which is mainly spoken in the Basque Country in Northern Spain and Southern France, is considered an isolated language. Linguistically, it is an agglutinative language, and morphologically more complex than English. Furthermore, Basque is a low resource language. Due to these characteristics, creating an MT system that deals with Basque is a challenging task.

As the MT Evaluation Campaign consists of translating subtitles from TED talks, we built our MT engines mainly from available subtitles. TED Talks[1] is an event where experts in different fields, such as education, business, science, etc. give a talk of up to 18 minutes to disseminate their ideas.

The use of subtitles as training data is potentially problematic as they may not be literal translation, causing the original and translated sentences not to be truly parallel. This is because subtitles are subjected to a great deal of adaptation. Localization strategies (adapting the text to suit consumers of a particular locale or culture), combined with the requirement to meet time constraints (where sentences in the source and target languages which have different length are supposed to appear on the screen within the same time frame), results in sentences which are comparable but not necessarily parallel [1].

Although the adaptation does not hinder human comprehension of the intended message, when these sentences are used as training data for an MT model, the translation inaccuracies become obstacles for the system to correctly learn to translate.

The system presented in this paper aims to overcome the aforementioned problems. First, the creation of synthetic data has two purposes: (i) it provides a new set of parallel sentences that mitigates the problem of Basque being a low resourced language; and (ii), artificially-created sentences tend to be more literal than usual translated subtitles. Therefore the former may constitute better training data for an MT model than the latter. Secondly, as TED Talks topics cover a wide variety of domains, we use data selection techniques to adapt an MT model to a particular test set.

The remainder of the paper is structured as follows. In Section 2, we describe related work regarding MT models that include Basque as source or target language. We also describe previous work on the use of synthetic data and data selection algorithms that are related to the systems described below. Section 3 describes the two steps (hybrid corpus creation and model adaptation) performed for building the MT system. In Section 4 we present an estimation of the performance of the models created. Finally, an overview of the system is described in Section 5.

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

2. Related Work

The system described in this paper is based on two main techniques: (a) incorporating synthetic sentences as training data (Section 2.2), and (b) adapting the model to the test set (Section 2.3).

2.1. Basque Machine Translation

Most of the work on MT involving Basque is based on the Basque-Spanish pair. We can find multiple MT approaches including Rule Based MT (RBMT) [2], or data-driven approaches [3] such as Example-based MT [4] or hybrid (Statistical MT and RBMT) [5] systems.

Dealing with low-resource languages is a problem for NMT approaches as they require large amounts of data in order to generate good translations. For some language pairs, SMT models can outperform NMT models when trained in limited amount of data [6]. In the work of Unanue et al. (2018) [7] they perform a comparison of Basque-English SMT and NMT models. Their finding reveals that SMT models trained with *PaCo2-EuEn* corpus in the Basque-to-English direction perform better than NMT models. In the reverse direction, however, NMT models can perform better when pre-trained embeddings (which have been trained using additional sentences from Basque Wikipedia) are given to the model.

Regarding Basque-Spanish NMT models, the most recent work is presented by Etchegoyen et al. (2018) [8] where they explored different methods of splitting words into morphemes to improve the translation.

2.2. Addition of Back-translated Sentences

As Basque is a low-resource language, the amount of available parallel data is very limited. A technique to increase the number of sentences is to artificially create sentences. Sennrich et al. (2016) [9], showed that NMT models could be boosted by adding backtranslated data.

Backtranslation in this paper designates the process of translating monolingual sentences in the target language into the source-side language. By doing this, a synthetic parallel corpus is created. Adding this corpus as training data can improve the performance of the model. In fact, models built using solely back-translated data can even achieve comparable performance to those trained with authentic or hybrid data [10].

2.3. Adaptation of the MT Model to the Test Set

There are several techniques for adapting a model to a particular domain [11], such as selecting relevant data (*data-centric* approaches), or modifying the model (*model-centric* approaches).

In the case where the test set is available, it is possible to adapt the model so it performs better in the given test. In our work, we used a combination of data-centric and model-centric approaches. First, we selected data that are relevant for the test set, and then we used fine-tuning to bias the model towards the test set.

Fine-tuning [12; 13], consists of using a pre-built NMT model (trained on general domain data), and training the last epochs on smaller amounts of in-domain data. An alternative to this technique is *gradual fine tuning* [14], which involves reducing the training data as the training proceeds.

While these fine-tuning techniques aim to adapt the NMT models towards a particular domain, Li et al. (2018) [15] proposed to use fine-tuning to adapt the model to the test set, which is closer to our approach. The main difference is that while in their work the model is adapted sentence-wise (one model for each sentence), in ours, it is adapted document-wise (one model for the document).

In order to select sentences that are closer to the test set we used Feature Decay Algorithms (FDA) [16; 17; 18]. This technique has been successfully applied in both SMT [19; 20; 21] and NMT [22].

FDA is a data selection method that not only aims to select sentences that are close to a seed (generally the test set), but also to promote the variability of the training data selected.

In order to achieve that, FDA scores each sentence s in the parallel data, and the sentence with the highest score is added to a list of selected sentences L. The score of the sentence is based on how similar it is to the seed (counting the *n*-grams shared with the seed), and how different it is to previously selected sentences (penalizing *n*-grams already contained in L), which increases the variability.

Using default values of the parameters, the score of a sentence is computed as in Equation (1):

$$score(s|L) = \frac{\sum_{ngr \in s} 0.5^{C_L(ngr)}}{length(s)} \quad (1)$$

where $C_L(ngr)$ is the count of the *n*-gram ngr in the pool of selected sentences L. The more occurrences of ngr there are in L the more penalized ngr is. The factor $0.5^{C_L(ngr)}$ in Equation (1) causes the *n*-gram to contribute less to the total score of the sentence.

3. System Description

The system built consists of two steps. First, (Section 3.2) we created a basic model using authentic and synthetic data. In the second step (Section 3.3), the model was fine-tuned to be adapted to the test set.

3.1. Basque-English Data

The Basque-English parallel data used in this work were obtained by combining the OpenSubtitles2016 (173K sentences), OpenSubtitles2018 [23] (805K sentences) and the *PaCo2-EuEn* corpus[2] [24] (130K sentences) provided in the

[2] komunitatea.elhuyar.org/ig/files/2016/01/PaCo_EuEn_corpus.tgz

IT domain MT Shared Task of WMT 2016 [25]. We randomly sampled 5000 sentences as our dev set and the rest (1M sentences) as the training set.

In order to build the NMT models we used OpenNMT-py, which is the Pytorch port of OpenNMT [26]. All the NMT models we built were configured with the same settings (the only difference is the training data used to build them). The value parameters were the default ones of OpenNMT-py (i.e. 2-layer LSTM with 500 hidden units, vocabulary size of 50000 words for each language). All the models were executed for 13 epochs, and we also used Byte Pair Encoding (BPE) [27] with 30000 merge operations, following the work of Etchegoyen et al. (2018) [8].

3.2. Addition of Synthetic Data

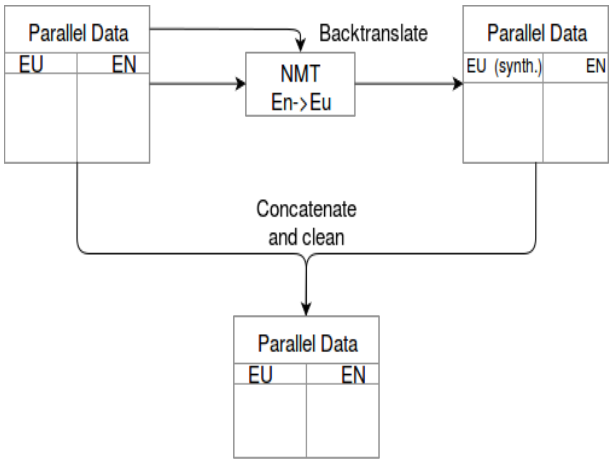


Figure 1: Creation of hybrid parallel corpus.

The first step in the construction of a baseline system is to extend the parallel corpus. In Figure 1 we present a diagram of how we built the corpus. Using an initial corpus of parallel Basque-English sentences we built an NMT model capable of translating sentences from English into Basque. Then, the English side of that parallel corpus was translated into Basque using the English to Basque NMT model.

Intuitively, translating the same sentences that were fed as training data should not be useful as it is likely to produce very similar sentences. However, the sentences produced by the model tend to be more literal translations, thereby avoiding the problems previously mentioned.

In Table 1 we show some examples of how synthetic data present Basque sentences that are closer to literal translation than a human-produced sentences. For example, in the first row, the translation for the English sentence "do I need to be there?" is "joan behar dut?", which literally means "do I have to go?". The artificially-created sentence is a more precise translation, as it uses the verb "be" ("egon") instead of the verb "go" ("joan"). In certain contexts, the use of one or another sentence does not affect the general understanding. However, using the wrong translation as training data for a model can hurt performance.

A similar effect is observed in the second row of Table 1 for the sentence "keep her steady, now.". The Basque translation of this sentence is "ez dadila mugitu." which uses the verb "mugitu" ("to move"), so it could be translated as "it shall not move" or "do not let it move". In contrast, the MT model produced the sentence, "eutsi gogor.", which used the verb "hold" ("eutsi"). Both translations are appropriate, but they belong to different contexts.

Finally, we see in the third row the English sentence "a suicide?". The corresponding sentence in Basque is "nor zen?" ("who was?"). In any other context, the two sentences have completely different meanings. The synthetic sentence by contrast is a literal translation.

	Authentic Basque	Synthetic Basque	English
1	joan behar dut?	hor egon behar dut?	do I need to be there?
2	ez dadila mugitu.	eutsi gogor.	keep her steady, now.
3	nor zen?	suizidioa?	a suicide?

Table 1: Examples of sentences in Basque (authentic), Basque (synthetic) and English translation.

Following backtranslation we obtained two parallel sets, with authentic and synthetic sentences. Next, we concatenated them as a single corpus. Note that, by doing so, the target-language sentences are duplicated.

Finally, we removed those sentences in which the length of the source and target sides differed substantially. In our work we kept a sentence pair (s, t) if $0.5 < \frac{len(s)}{len(t)} > 1.5$, in order to remove the 10% outliers. In total 255K sentences were removed (137K sentences 118K sentences from authentic and synthetic sets, respectively). The hybrid corpus contained, therefore, 1.93M sentence pairs.

We applied these criteria to both corpus of authentic, and synthetic sentences, so the potentially unaligned sentences are ignored and bad translated sentences are not considered, respectively.

3.3. Adaptation to the Test Set

The second step of building the model is to adapt it to a particular test set. The work of Poncelas et al. (2018) [22] showed that when the test set is available during training time it is possible to fine-tune a model to improve the translation of that particular test set.

In Figure 2 we show how we fine-tuned our NMT model, which requires three phases as follows:

1. Data Selection: In this phase we aimed to retrieve English sentences that were close to the test set. As the test set was in Basque, we first created an approximated translation using the NMT model built as ex-

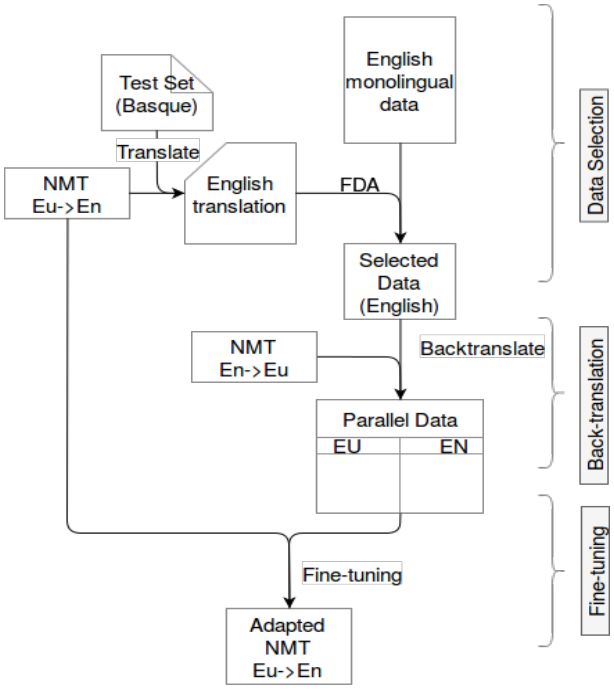


Figure 2: Fine tuning with synthetic data

plained in Section 3.2. This translation can be used as the seed for FDA and extract a set of sentences, from a monolingual English corpus, that were close to the pre-translated set and hence, to the test set. In this work we extracted 50,000 sentences from English training data provided in the WMT 2015 Translation Task [28].

2. Back-translation: The subset of selected English sentences were back-translated (we reused the same English-to-Basque model built as explained in Section 2.2 to create backtranslated data) in order to build a parallel corpus.

3. Fine-tuning: The synthetic parallel corpus was used to fine-tune the MT model for one epoch. In this way, the model was tailored to the test set.

4. Experimental Results

In order to estimate the performance of the final and intermediate models described through Section 3 we evaluated them using the development set (containing 1K sentences extracted from subtitles of TED talks) provided by the organizers of the IWSLT Evaluation Campaign.

The models evaluated are: (a) the model built with only authentic data (*base model*); (b) the model built with the combination of authentic and synthetic data (*hybrid model*); and (c), the *hybrid model* adapted to the test set using FDA-retrieved data (*FDA model*).

We used several evaluation metrics to compare the outputs of the three models to a human-translated reference. In Table 2 we can see the evaluation scores for each model. The

	base model	*hybrid model*	*FDA model*
BLEU	0.1315	**0.1426***	**0.1450***
NIST	4.459	**4.683**	**4.733**
TER	0.8508	0.8576	0.8666
METEOR	0.1429	**0.1501***	**0.1528****
CHRF3	34.05	**35.92**	**36.24**
CHRF1	37.40	**38.67**	**38.81**

Table 2: Evaluation of the model built only with authentic data and using both authentic and synthetic data.

metrics we present are BLEU [29], NIST [30], TER [31], METEOR [32] and CHRF3 [33].

We also marked in bold the scores that outperform those of the *base model* (first column of Table 2) and marked with an asterisk the scores (among BLEU, TER and METEOR) that are statistically significant at level p=0.01. This was computed with multeval [34] using Bootstrap Resampling [35]. The two asterisks in column *FDA model* (METEOR row) indicate that it is statistically significant at p=0.01 when compared not only to the *base model* but also to the *hybrid model*.

As mentioned in Section 3.2, the addition of synthetic data (even if it consists of a backtranslation of the same data used for training the model) is helpful. This is verified with the results in column *hybrid model* in Table 2. As we can see, most of the *hybrid model* scores of the model are better than the model built with authentic data only (*base model* column) and according to two of the scores, the improvements are statistically significant at p=0.01. In fact, a model built using only synthetic data (Table 3) can achieve improvements over the *base model*, according to METEOR and CHRF3 evaluation metric.

	synth. model
BLEU	0.1224
NIST	4.074
TER	0.9769
METEOR	**0.1481**
CHRF3	**36.22**
CHRF1	36.40

Table 3: Evaluation of the model built with synthetic data only.

Finally, fine-tuning the *hybrid model* using sentences that are close to the test set is also beneficial. As we can see in the column *FDA model* (in Table 2), most of the scores (except TER) are better than those of the *base model* or even the *hybrid model*, and according to METEOR metric the improvement is statistically significant at p=0.01.

5. Conclusion

In this paper we have described the ADAPT system presented for the Low Resource MT Evaluation Campaign of IWSLT 2018. The system translates from Basque into English.

Basque is a morphologically rich language, which causes the task of building an MT model to be more difficult than languages such as Spanish or German. Furthermore, the available parallel Basque-English data are scarce.

Due to the limited resources of texts in Basque, we generated synthetic data that successfully boosted the performance of the MT model trained solely with authentic sentences.

Additionally, we have used a supplementary monolingual English corpus so we could retrieve sentences close to the test set and further improve our model.

6. Acknowledgments

The research leading to these results was carried out as part of the TADEEP project (Spanish Ministry of Economy and Competitiveness TIN2015-70214-P, with FEDER funding). This work has been supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

7. References

[1] M. Fishel, Y. Georgakopoulou, S. Penkale, V. Petukhova, M. Rojc, M. Volk, and A. Way, "From subtitles to parallel corpora," in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, Trento, Italy, 2012, pp. 3–6.

[2] A. Mayor, I. Alegria, A. D. De Ilarraza, G. Labaka, M. Lersundi, and K. Sarasola, "Matxin, an open-source rule-based machine translation system for Basque," *Machine translation*, vol. 25, no. 1, p. 53, 2011.

[3] G. Labaka, N. Stroppa, A. Way, and K. Sarasola, "Comparing rule-based and data-driven approaches to spanish-to-basque machine translation," in *Machine Translation Summit XI*, Copenhagen, Denmark, 2007, pp. 297–304.

[4] N. Stroppa, D. Groves, A. Way, and K. Sarasola, "Example-based machine translation of the basque language," pp. 232–241, 2006.

[5] G. Labaka, C. España-Bonet, L. Màrquez, and K. Sarasola, "A hybrid machine translation architecture guided by syntax," *Machine translation*, vol. 28, no. 2, pp. 91–125, 2014.

[6] M. Dowling, T. Lynn, A. Poncelas, and A. Way, "SMT versus NMT: Preliminary comparisons for Irish," in *Technologies for MT of Low Resource Languages (LoResMT 2018)*, Boston, USA, 2018, pp. 12–20.

[7] I. J. Unanue, L. G. Arratibel, E. Z. Borzeshi, and M. Piccardi, "English-Basque statistical and neural machine translation." in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018, pp. 880–885.

[8] T. Etchegoyhen, E. M. Garcia, A. Azpeitia, G. Labaka, I. Alegria, I. C. Etxabe, A. J. Carrera, I. E. Santos, and M. M. eta Eusebi Calonge, "Neural machine translation of Basque," in *Proceedings of the 21st Annual Conference of the European Association for Machine Translation (EAMT)*, Alacant, Spain, 2018, pp. 139–148.

[9] 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, 2016, pp. 86–96.

[10] A. Poncelas, D. Shterionov, A. Way, G. M. de Buy Wenniger, and P. Passban, "Investigating back-translation in neural machine translation," in *21st Annual Conference of the European Association for Machine Translation*, Alacant, Spain, 2018, pp. 249–258.

[11] C. Chu and R. Wang, "A survey of domain adaptation for neural machine translation," *arXiv preprint arXiv:1806.00258*, 2018.

[12] 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, 2015, pp. 76–79.

[13] M. Freitag and Y. Al-Onaizan, "Fast domain adaptation for neural machine translation," *arXiv preprint arXiv:1612.06897*, 2016.

[14] M. van der Wees, A. Bisazza, and C. Monz, "Dynamic data selection for neural machine translation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, 2017, pp. 1400–1410.

[15] X. Li, J. Zhang, and C. Zong, "One Sentence One Model for Neural Machine Translation," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018, pp. 910–917.

[16] E. Biçici and D. Yuret, "Instance selection for machine translation using feature decay algorithms," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, 2011, pp. 272–283.

[17] E. Biçici, Q. Liu, and A. Way, "ParFDA for fast deployment of accurate statistical machine translation systems, benchmarks, and statistics," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal, 2015, pp. 74–78.

[18] E. Biçici and D. Yuret, "Optimizing instance selection for statistical machine translation with feature decay algorithms," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 339–350, 2015.

[19] E. Biçici, "Feature decay algorithms for fast deployment of accurate statistical machine translation systems," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August 2013, pp. 78–84.

[20] A. Poncelas, A. Way, and A. Toral, "Extending feature decay algorithms using alignment entropy," in *International Workshop on Future and Emerging Trends in Language Technology*, Seville, Spain, 2016, pp. 170–182.

[21] A. Poncelas, G. M. de Buy Wenniger, and A. Way, "Applying n-gram alignment entropy to improve feature decay algorithms," *The Prague Bulletin of Mathematical Linguistics*, vol. 108, no. 1, pp. 245–256, 2017.

[22] A. Poncelas, G. M. Buy Wenniger, and A. Way, "Feature decay algorithms for neural machine translation," in *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, Alacant, Spain, 2018, pp. 239–248.

[23] J. Tiedemann, "News from OPUS - A collection of multilingual parallel corpora with tools and interfaces," in *Recent Advances in Natural Language Processing*, N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, Eds. Borovets, Bulgaria: John Benjamins, Amsterdam/Philadelphia, 2009, vol. V, pp. 237–248.

[24] I. San Vicente, I. Manterola, *et al.*, "PaCo2: A fully automated tool for gathering parallel corpora from the web." in *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, Turkey, 2012, pp. 1–6.

[25] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz, *et al.*, "Findings of the 2016 conference on machine translation." in *ACL 2016 First Conference on Machine Translation (WMT16)*, Berlin, Germany, 2016, pp. 131–198.

[26] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations*, Vancouver, Canada, 2017, pp. 67–72.

[27] 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)*, vol. 1, Berlin, Germany, 2016, pp. 1715–1725.

[28] 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*, Lisboa, Portugal, September 2015, pp. 1–46.

[29] 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, July 2002, pp. 311–318.

[30] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proceedings of the second international conference on Human Language Technology Research*, San Diego, CA, 2002, pp. 138–145.

[31] 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.

[32] S. Banerjee and A. Lavie, "Meteor: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, Ann Arbor, Michigan, 2005, pp. 65–72.

[33] M. Popovic, "chrF: character n-gram F-score for automatic MT evaluation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal, 2015, pp. 392–395.

[34] J. H. Clark, C. Dyer, A. Lavie, and N. A. Smith, "Better hypothesis testing for statistical machine translation: Controlling for optimizer instability," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, Portland, Oregon, 2011, p. 176–181.

[35] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004, pp. 388–395.

The University of Helsinki submissions to the IWSLT 2018 low-resource translation task

Yves Scherrer

Department of Digital Humanities
University of Helsinki, Finland
yves.scherrer@helsinki.fi

Abstract

This paper presents the University of Helsinki submissions to the Basque–English low-resource translation task. Our primary system is a standard bilingual Transformer system, trained on the available parallel data and various types of synthetic data. We describe the creation of the synthetic datasets, some of which use a pivoting approach, in detail. One of our contrastive submissions is a multilingual model trained on comparable data, but without the synthesized parts. Our bilingual model with synthetic data performed best, obtaining 25.25 BLEU on the test data.

1. Introduction

The University of Helsinki has participated in the IWSLT low-resource translation task on Basque-to-English translation with one primary and two contrastive systems. Our experiments mainly focused on creating synthetic training data for classical supervised neural machine translation models. In particular, we show that a bilingual system trained on partly synthetic data performs better than a multilingual system that includes the same data in their original, non synthetic form. Our best submitted system obtained a BLEU score of 25.25.

Section 2 describes the available Basque–English parallel datasets at the basis of our systems, as well as a baseline system trained on these parallel datasets alone. In Section 3, we present additional datasets that contain either Basque or English text, but not both. We discuss several strategies for synthetically creating parallel Basque–English datasets out of these sources, and show the impact of these synthetic datasets on translation quality. In Section 4, we present a contrastive system that uses the additional datasets in their original state, without the synthesized parts. Section 5 summarizes our submissions and details the post-processing steps carried out at prediction time.

2. Parallel Basque–English data

The IWSLT organizers released an in-domain data set for Basque-to-English translation containing 64 TED talks for training and 10 TED talks for development [1]. Another 10 TED talks have been held out for testing.

The only allowed out-of-domain data source containing parallel Basque–English datasets is OPUS [2, 3]: it contains computer program localization files (repositories GNOME, KDE4 and Ubuntu), crowd-sourced translations (Tatoeba) and film subtitles (OpenSubtitles2018). We only selected OpenSubtitles2018 as the largest and most domain-similar dataset for our experiments. Table 1 summarizes the available parallel data.[1]

Source	Talks	Lines	EU tokens	EN tokens
TED train	64	5623	97k	128k
OST	—	806k	4.8M	6.5M
TED dev	*10*	*1140*	*20k*	*27k*

Table 1: Basque–English parallel data.

2.1. Baseline system

We trained a baseline system using only the parallel data mentioned in the previous section. Data were tokenized and truecased using the Moses scripts [4]; no effort was spent on adapting the tokenization tools to Basque. Following the good results on various typologically diverse language pairs, we used the Transformer model setup [5] as implemented in Marian-NMT [6] (see Appendix). We used an initial setting of 20 000 BPE units [7] shared across both languages with tied embeddings. Training of this model converged after about 20 hours on a single-GPU node, obtaining a BLEU score of 15.40 on the development set (see first line of Table 5).[2]

3. Synthetic data

Backtranslation has proven to be an effective way of improving the performance of neural machine translation systems by taking advantage of existing monolingual datasets for the target language [8]. Monolingual data of the target language is translated to the source language using a target-to-source

[1] In all tables, validation and test sets are displayed in italics, whereas the translation output of the described system is displayed in bold (if applicable).

[2] All BLEU scores were computed using the *multi-bleu-detok.perl* script of the Moses distribution.

translation system. The resulting bilingual dataset, whose source is noisy, is then used as additional training data for the source-to-target translation system.

In our setting, direct backtranslation would amount to translating English data to Basque, but such an English-to-Basque system would have to be trained on the same small dataset as the baseline system presented above. Therefore, we experimented with other ways of creating synthetic data, exploiting the larger Spanish–Basque and Spanish–English datasets and using Spanish as a pivot language [9].[3] The different data augmentation strategies are discussed in Sections 3.1 to 3.3, whereas the Basque-to-English systems trained on these synthetic datasets are presented in Section 3.4 and Table 5.

3.1. Direct backtranslation of TED talks

The provided in-domain data contains a total 2683 English TED talks. Excluding those that already have Basque translations (for training, development or testing) and excluding those that do not have a Spanish translation (to provide comparability with the experiment below), 2576 English TED talks can be backtranslated to Basque.

Source	Talks	Lines	EN tokens	EU tokens
TED train	64	5623	128k	97k
OST	—	806k	6.5M	4.8M
TED dev	*10*	*1140*	*27k*	*20k*
TED direct-BT	2576	271k	6.2M	**3.9M**

Table 2: Basque–English data used to train the backtranslation model (above the line) and monolingual English data backtranslated with this model (below the line, backtranslation output in bold).

In this first experiment, we train an English-to-Basque system analogously to the baseline system above, using the same training data, parameter settings (20k joint BPE units) and development set for validation, obtaining a BLEU score of 8.65 on the English-to-Basque development set.[4] This low score confirmed our initial reservations about direct backtranslation. We nevertheless translate the monolingual English TED talks to Basque with this system. Table 2 summarizes the data of this experiment.

[3]Note that we employ the term *pivot language* in the context of a data augmentation strategy, not of a machine translation model *per se*. We take a parallel corpus of languages $\langle X, Y \rangle$ and translate its X side to language Z using a $X \rightarrow Z$ machine translation system, yielding a corpus of languages $\langle Z, Y \rangle$. This approach is simpler than the common acceptation of pivot-based translation, where two (more or less independent) translation models are trained, and the output of the first serves as the input of the second one. Examples of recent work in this area include [10, 11].

[4]The BLEU score of a Basque-to-English system including these backtranslations is 21.04, as shown in the second row of Table 5.

3.2. Pivot-based backtranslation of TED talks

We hypothesize that the direct backtranslation approach would not be particularly effective, as the system used to generate them would suffer from the same data sparsity issues as the baseline system (trained with the same data, but in the other direction). In order to take advantage of the other datasets provided by the organizers, we follow a pivot-based approach along the lines of [9]: we take all TED talks available in both English and Spanish (but not Basque), translate the Spanish version to Basque, and align the Basque side with the English side to constitute additional Basque–English data. In this setting, the backtranslation model needs to be trained on Spanish-to-Basque data; using the same 64+10 TED talks for training and validation, as well as the out-of-domain Open Data Euskadi (ODE) dataset and the Basque–Spanish OpenSubtitles (OST), we create a Transformer model with the same parameters as the baseline model. At the end of training, this system obtained a BLEU score of 14.52 on the Spanish-to-Basque development set.

The resulting data consists thus of the same English target sentences as above but different Basque source sentences. Details on the setup are given in Table 3. It is striking that the Basque sentences translated via Spanish are considerably longer than those translated directly from English (4501k total tokens in Table 3 vs. 3886 total tokens in Table 2). The experiments described below will show which of the two datasets improves translation most, and whether the two datasets are complementary or not.

Source	Talks	Lines	ES tokens	EU tokens
TED train	64	5546	124k	98k
OST	—	794k	5.8M	4.8M
ODE	—	927k	23.1M	17.5M
TED dev	*10*	*1122*	*26k*	*20k*
TED pivot-BT	2576	271k	EN 6.2M	**4.5M**

Table 3: Basque–Spanish data used to train the backtranslation model (above the line) and monolingual Spanish data backtranslated to Basque and aligned with English (below the line).

3.3. Pivot-based translation of Open Data Euskadi

Whereas backtranslation yields datasets with noisy source sides and clean target sides, we also wanted to explore the impact of a corpus with clean source side and noisy target side. This approach is not generally used in standard high-resource settings, but could yield additional improvements in low-resource settings. The Open Data Euskadi corpus is a good candidate for this approach. It is rather large and contains Basque–Spanish parallel data. In order to create a Basque–English version of this corpus, we proceed by translating the Spanish version to English and aligning it with the existing Basque one.

The Spanish–English system is trained using most of the parallel data that was made available in WMT 2013, the last year in which Spanish–English featured as a WMT news translation language pair (see Table 4) [12]. In particular, we use the CommonCrawl, Europarl V7, NewsCommentary V12 and UN datasets for training,[5] the NewsTest 2008-2012 corpora for validation and NewsTest 2013 for testing. We did not use OpenSubtitles as we did not find it helpful for translating the legal and news domain documents present in Open Data Euskadi. Due to the larger training corpora sizes, we increased the vocabulary to 40k joint BPE units, but kept the same Transformer architecture and parameters otherwise. This system obtained a BLEU score of 29.69 on the development set and 31.45 on the test set, slightly surpassing the best systems participating in WMT 2013.[6] The figures of the resulting Basque–English Open Data Euskadi corpus are shown on the last line of Table 4.

Source	Lines	ES tokens	EN tokens
CommonCrawl	1845k	49.5M	46.9M
Europarl	1965k	57.0M	54.5M
NewsCommentary	292k	8.5M	7.5M
UN	11196k	366.1M	320.0M
News dev	*13k*	*357k*	*336k*
News test	*3k*	*70k*	*64k*
ODE pivot-T	927k	EU 17.3M	**21.5M**

Table 4: Spanish–English data used to train the translation model (above the line) and monolingual Spanish data translated to English and aligned with Basque (below the line).

3.4. Bilingual systems using synthetic data

We trained various Basque-to-English systems with different combinations of the synthetic datasets described above. All experiments use the same Transformer model architecture, but slightly different vocabulary sizes (see below).

For some experiments, we introduce variants with domain labels [14, 15]. Tars et al. have found domain labeling useful to teach the model about possible domain mismatches in the training data. In our experiments, we use four labels, distinguishing text sources (TED, OST, OPD) and methods of corpus construction (TED-parallel and TED-BT). The validation and test instances are labeled as TED-parallel. Domain labels were included as the first tokens of each sentence. Table 5 summarizes these experiments.

Table 5 shows that any additional synthetic dataset helps in the given low-resource setting. The direct TED backtranslations are surprisingly helpful despite their low quality, although the pivot-based TED backtranslations are much more useful, presumably due to the higher quality of the system that generated them. The impact of the ODE synthetic dataset is less remarkable, but still improves BLEU scores by 2-3 absolute points. Interestingly, the direct and pivot-based TED backtranslations are somewhat complementary, yielding slight improvements compared to using just the pivot-based ones.

Training data	Domain labels	BPE	BLEU
Parallel (= TED train + OST)	No	20k	15.40
+ TED direct-BT	No	20k	21.04
+ TED pivot-BT	No	20k	23.20
+ TED direct-BT + ODE pivot-T	No	30k	23.20
	Yes	30k	23.84
+ TED pivot-BT + ODE pivot-T	No	30k	24.22
	Yes	30k	24.52
+ TED direct-BT + TED pivot-BT	No	30k	24.39
+ ODE pivot-T	Yes	30k	**25.06**

Table 5: Experiments with different combinations of training data.

On the basis of the *Parallel + TED pivot-BT* model (third line of Table 5), we performed a grid search to find the best subword encoding scheme. We used various sizes of joint BPE vocabularies with tied embeddings (10k, 15k, 20k, 25k, 30k, 35,) and various sizes of language-specific BPE vocabularies in conjunction with distinct embeddings (10k, 15k, 20k, 25k, 30k, 35k per language). The difference between the worst and best setting lay at 1.5 BLEU points. The best results were achieved with joint vocabularies and tied embeddings and a total of 25k-30k subword units. The final submissions were made with a joint vocabulary of 30k units, like most experiments presented in Table 5.

Domain labels show consistent improvements of about 0.5 BLEU points. As mentioned above, we labeled the validation data with TED-parallel. Additional experiments using other domain labels at test time have shown the following results: TED-BT +0.04 BLEU, OST -2.83 BLEU, OPD -4.70 BLEU, no label -1.71 BLEU. This experiment shows that the TED-parallel and TED-BT labels yield similar results (the difference is probably not statistically significant), suggesting that the distinction between genuinely parallel and backtranslated TED data may not have been necessary. We nevertheless kept the TED-parallel label also for the test data.

4. Multilingual system

Johnson et al. [14] have shown that multilingual translation models can be trained by using training data of various languages and directions and prepending a target language label to each source sentence. One interesting use case of such multilingual models is zero-shot translation, where the

[5] We experimented with a reduced training set consisting of Europarl and NewsCommentary only, but results were not quite as good as with the complete training data.

[6] The best WMT 2013 submissions were the phrase-based statistical systems by the University of Edinburgh team, with BLEU scores of 31.37 in the unconstrained setting and 30.59 in the constrained setting [13].

System	Model type and features		BLEU	NIST	TER
Primary	Bilingual model	With sentence splitting	25.01	6.45	59.48
Contrastive 1	Bilingual model	No sentence splitting	**25.25**	**6.47**	**58.83**
Contrastive 2	Multilingual model	With sentence splitting	22.55	6.10	60.48

Table 6: Submitted systems and official results on the test set.

source language and target language have both been seen by the model, but not in that particular combination. In our case, we are not interested in zero-shot translation, as we do have a sizeable set of Basque-to-English training data. Rather, we wanted to see to what extent multilingual modelling could supplant the creation of synthetic data. To this end, we train a single multilingual model with the following datasets: the parallel Basque–English TED and OpenSubtitles data (as in the baseline model), the parallel English–Spanish TED data in both directions (as used to train the pivot-based backtranslation model), and the Basque–Spanish Open Data Euskadi data (see Table 7). In this setting, we only have English and Spanish as target languages and consequently only use the two target language labels TO_EN and TO_ES. We do not use additional domain labels in this experiment. The model architecture remains the same, but we use a joint trilingual vocabulary consisting of 40k BPE units.

Source	Lines	Source tokens	Target tokens
TED train	5623	97k EU	128k EN
OST	805k	4.8M EU	6.5M EN
TED train	277k	6.3M EN	6.0M ES
TED train	277k	6.0M ES	6.3M EN
ODE	926k	17.5M EU	23.1M ES
TED dev	*1140*	*20k EU*	*27k EN*

Table 7: Data used to train the multilingual model.

Although we used almost the same datasets as in the systems presented above (with the exception of the WMT English–Spanish data), the multilingual model failed to achieve competitive results, with 22.55 BLEU on the validation set. There are several reasons for this lower-than-expected performance. First, the training of the multilingual model was stopped before convergence, after a training time of 72 hours. Nevertheless, the learning curve does not indicate the possibility of substantial improvements if training had continued. Second, the multilingual model has to learn three languages on the source side and two languages on the target side instead of a one-to-one mapping. Its task is thus inherently more complex, and it seems that the three languages in question (Basque, English and Spanish) are typologically too diverse for the model to generalize. Finally, [14] show that good data sampling strategies are crucial when training multilingual models with unbalanced data sizes. In this regard, oversampling the Basque-to-English resources or fine-tuning the model to the target language pair might have helped. Despite its lower performance, we base one of our contrastive submissions on the multilingual model.

5. Submissions

We decided to submit output from two models, the bilingual system trained with all synthetic data and domain labels (last line of Table 5), and the multilingual system described in Section 4.

We have found in a different context [16] that systems trained on single sentences may not be able to translate utterances consisting of several sentences completely. Although there was no particular evidence of such problems occurring in the experiments at hand (since a large portion of the TED training data already contains multi-sentence utterances), we carried out some experiments on this issue. Concretely, we applied a simple sentence splitter to the source text, translated each sentence separately, and merged them back together. In the validation set, 214 (of 1140) lines were split, and sentence splitting improved the BLEU score by 0.26 points. However, qualitative inspection of the results did not show convincing evidence in favor or against sentence splitting. Therefore, we submitted systems with and without sentence splitting.

Also, due to an error in the postprocessing script, the submitted translations were accidentally detokenized with the Basque detokenizer (and some additional rules) rather than the English one. The added rules minimized the adverse effect of this error, such that it only affected two tokens in the test set, resulting in an estimated impact on BLEU score of about 0.01.

Table 6 summarizes the submitted systems with the official results. Sentence splitting turned out to have a slightly negative impact on the translation of the test set, whereas the difference between the bilingual and multilingual system is comparable to the one that was observed with the validation set.

6. Conclusions

The University of Helsinki submissions on Basque–English leverage the existing parallel corpora for other language pairs to create synthetic data of various types. In particular, we have found pivot-based (back-)translation to be a useful approach to increase the amounts of Basque–English training data. In this setting, one side of a parallel corpus is translated to a third language, and this translated output is then aligned with the other side of the original parallel corpus. By using

various synthetic datasets, we were able to increase translation performance from 14.68 BLEU to 25.06 BLEU on the development set.

Our contrastive multilingual model performed less well, although it saw almost the same data as the bilingual model and its auxiliary models used to create the synthetic data. It remains to be seen if better balancing of the training data, possibly including some fine-tuning, as well as the inclusion of domain labels and additional Spanish–English training data could make this model more competitive. Also, both approaches could be combined by training a multilingual model with added synthetic data.

7. References

[1] 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.

[2] J. Tiedemann, "Parallel data, tools and interfaces in OPUS," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), May 2012.

[3] 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. Calzolari, 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.

[4] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *ACL'07: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Demonstration Session*. Association for Computational Linguistics, 2007, pp. 177–180.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[6] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, "Marian: Fast neural machine translation in C++," in *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 116–121.

[7] 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.

[8] ——, "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.

[9] J. Tiedemann, "Character-based pivot translation for under-resourced languages and domains," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012, pp. 141–151.

[10] Y. Cheng, Y. Liu, Q. Yang, M. Sun, and W. Xu, "Neural machine translation with pivot languages," *CoRR*, vol. abs/1611.04928, 2016.

[11] S. Ren, W. Chen, S. Liu, M. Li, M. Zhou, and S. Ma, "Triangular architecture for rare language translation," in *Proceedings of ACL 2018*, Melbourne, Australia, 2018.

[12] O. Bojar, C. Buck, C. Callison-Burch, C. Federmann, B. Haddow, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2013 Workshop on Statistical Machine Translation," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 1–44.

[13] N. Durrani, B. Haddow, K. Heafield, and P. Koehn, "Edinburgh's machine translation systems for European language pairs," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 114–121.

[14] 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.

[15] S. Tars and M. Fishel, "Multi-domain neural machine translation," in *Proceedings of the 21st Annual Conference of the European Association for Machine Translation (EAMT'2018)*, J. A. Pérez-Ortiz, F. Sánchez-Martínez, M. Esplà-Gomis, M. Popović, C. Rico, A. Martins, J. V. den Bogaert, and M. L. Forcada, Eds., 2018.

[16] A. Raganato, Y. Scherrer, T. Nieminen, A. Hurskainen, and J. Tiedemann, "The University of Helsinki submissions to the WMT18 news task," in *Proceedings of the Third Conference on Machine Translation (WMT18)*. Association for Computational Linguistics, 2018.

8. Appendix

All models presented in this paper were trained using the parameter settings described in `https://github.com/marian-nmt/marian-examples/tree/master/transformer`, which correspond roughly to the base setup of [5].

The relevant parameters are as follows:

```
marian --type transformer
--max-length 200 --mini-batch-fit
-w 10000 --maxi-batch 1000
--early-stopping 10 --valid-freq 5000
--valid-metrics cross-entropy
perplexity translation
--valid-mini-batch 64 --beam-size 6
--normalize 0.6 --enc-depth 6
--dec-depth 6 --transformer-heads 8
--transformer-postprocess-emb d
--transformer-postprocess dan
--transformer-dropout 0.1
--label-smoothing 0.1
--learn-rate 0.0003 --lr-warmup 16000
--lr-decay-inv-sqrt 16000
--optimizer-params 0.9 0.98 1e-09
--clip-norm 5 --tied-embeddings-all
--sync-sgd --exponential-smoothing
--seed 1111
```

The MeMAD Submission to the IWSLT 2018 Speech Translation Task

Umut Sulubacak * *Jörg Tiedemann* * *Aku Rouhe* † *Stig-Arne Grönroos* † *Mikko Kurimo* †

∗ Department of Digital Humanities / HELDIG
University of Helsinki, Finland
{umut.sulubacak | jorg.tiedemann}@helsinki.fi

† Department of Signal Processing and Acoustics
Aalto University, Finland
{aku.rouhe | stig-arne.gronroos | mikko.kurimo}@aalto.fi

Abstract

This paper describes the MeMAD project entry to the IWSLT Speech Translation Shared Task, addressing the translation of English audio into German text. Between the pipeline and end-to-end model tracks, we participated only in the former, with three contrastive systems. We tried also the latter, but were not able to finish our end-to-end model in time.

All of our systems start by transcribing the audio into text through an automatic speech recognition (ASR) model trained on the TED-LIUM English Speech Recognition Corpus (TED-LIUM). Afterwards, we feed the transcripts into English-German text-based neural machine translation (NMT) models. Our systems employ three different translation models trained on separate training sets compiled from the English-German part of the TED Speech Translation Corpus (TED-TRANS) and the OPENSUBTITLES2018 section of the OPUS collection.

In this paper, we also describe the experiments leading up to our final systems. Our experiments indicate that using OPENSUBTITLES2018 in training significantly improves translation performance. We also experimented with various pre- and postprocessing routines for the NMT module, but we did not have much success with these.

Our best-scoring system attains a BLEU score of 16.45 on the test set for this year's task.

1. Introduction

The evident challenge of speech translation is the transfer of implicit semantics between two different modalities. An end-to-end solution to this task must deal with the challenge posed by intermodality simultaneously with that of interlingual transfer. In a traditional pipeline approach, while speech-to-text transcription is abstracted from translation, there is then the additional risk of error transfer between the two stages. The MeMAD project[1] aims at multilingual description and search in audiovisual data. For this reason, multimodal translation is of great interest to the project.

Our pipeline submission to this year's speech translation task incorporates one ASR model and three contrastive NMT models. For the ASR module, we trained a time-delay neural network (TDNN) acoustic model using the Kaldi toolkit [1] on the provided TED-LIUM speech recognition corpus [2]. We used the transformer implementation of MarianNMT [3] to train our NMT models. For these models, we used contrastive splits of data compiled from two different sources: The n-best decoding hypotheses of the TED-TRANS [4] in-domain speech data, and a version of the OPENSUBTITLES2018 [5] out-of-domain text data (SUBS), further "translated" to an ASR-like format (SUBS-ASR) using a sequence-to-sequence NMT model. The primary system in our submission uses the NMT model trained on the whole data including SUBS-ASR, whereas one of the two contrastive systems uses the original SUBS before the conversion to an ASR-like format, and the other omits OPENSUBTITLES2018 altogether.

We provide further details about the ASR module in Section 2. Later, we provide a review of our experiments on the NMT module in Section 3. The first experiment we describe involves a pre-processing step where we convert our out-of-domain training data to an ASR-like format to avoid mismatch between source-side training samples. Afterwards, we report a postprocessing experiment where we retrain our NMT models with lowercased data, and defer case restoration to a subsequent procedure, and another where we translate several ASR hypotheses at once for each source sample, re-rank their output translations by a language model, and then choose the best-scoring translation for that sample. We present our results in Section 4 along with the relevant discussions.

2. Speech Recognition

The first step in our pipeline is automatic speech recognition. The organizers provide a baseline ASR implementation,

[1] https://www.memad.eu/

which consists of a single, end-to-end trained neural network using a Listen, Attend and Spell (LAS) architecture [6]. The baseline uses the XNMT toolkit [7]. However, we were not able to compile the baseline system, so we trained our own conventional, hybrid TDNN-HMM ASR system using the Kaldi toolkit.

2.1. Architecture

Our ASR system uses the standard Kaldi recipe for the TED-LIUM dataset (release 2), although we filter out some data from the training set to comply with the IWSLT restrictions. The recipe trains a TDNN acoustic model using the lattice-free maximum mutual information criterion [8]. The audio transcripts and large amount of out-of-domain text data included with the TED-LIUM dataset are used to train a heavily pruned 4-gram language model for first-pass decoding and less pruned 4-gram model for rescoring.

2.2. Word Error Rates

The LAS architecture has achieved state-of-the-art word error rates (WER) on a task with two orders of magnitude more training data than here [9], but on smaller datasets hybrid TDNN-HMM ASR approaches are still considerably better. Table 1 shows the results of our ASR model contrasted with those reported by XNMT in [7], on the TED-LIUM development and test sets.

Model	Dev WER	Test WER
TDNN + large 4-gram	8.24	8.83
LAS	15.83	16.16

Table 1: Word error rates on the TED-LIUM dataset.

3. Text-Based Translation

The ASR stage of our pipeline effectively converts the task of speech translation to text-based machine translation. For this stage, we build a variety of NMT setups and assess their performances. We experiment variously with the training architecture, different compositions of the training data, and several pre- and postprocessing methods. We present these experiments in detail in the subsections to follow, and then discuss their results in Section 4.

3.1. Data Preparation

We used the development and test sets from 2010's shared task for validation during training, and the test sets from the tasks between 2013 and 2015 for testing performance during development. In all of our NMT models, we preprocessed our data using the punctuation normalization and tokenization utilities from Moses [10], and applied byte-pair encoding [11] through full-cased and lowercased models as relevant, trained on the combined English and German texts in TED-TRANS and SUBS using 37,000 merge operations to create the vocabulary.

We experiment with attentional sequence-to-sequence models using the Nematus architecture [12] with tied embeddings, layer normalization, RNN dropout of 0.2 and source/target dropout of 0.1. Token embeddings have a dimensionality of 512 and the RNN layer units a size of 1024. The RNNs make use of GRUs in both, encoder and decoder. We use validation data and early stopping after five cycles (1,000 updates each) of decreasing cross-entropy scores. During training we apply dynamic mini-batch fitting with a workspace of 3GB. We also enable length normalization.

For the experiments with the transformer architecture we apply the standard setup with six layers in encoder and decoder, eight attention heads and a dynamic mini-batch fit to 8GB of work space. We also add recommended options such as transformer dropout of 0.1, label smoothing of 0.1, a learning rate of 0.0003, a learning-rate warmup with a linearly increasing rate during the first 16,000 steps, a decreasing learning rate starting at 16,000 steps, a gradient clip norm of 5 and exponential smoothing of parameters.

All translations are created with a beam decoder of size 12.

3.1.1. ASR Output for TED Talks

Translation models trained on standard language are not a good fit for a pipeline architecture that needs to handle noisy output from the ASR component discussed previously in Section 2. Therefore, we ran speech recognition on the entire TED-TRANS corpus in order to replace the original, human-produced English transcriptions with ASR output, which has realistic recognition errors.

To generate additional speech recognition errors to the training transcripts, we selected the top-50 decoding hypotheses. We did the same also for the development data to test our approach. We can now sample from those ASR hypotheses to create training data for our translation models that use the output of English ASR as its input. We experimented with various strategies varying from a selection of the top n ASR candidates to different mixtures of hypotheses of different ranks of confidence. Some of these are shown in Table 2. In the end, there was not a lot of variance between the scores resulting from this selection, and we decided to use the top-10 ASR outputs in the remaining experiments to encourage some tolerance for speech recognition errors in the system.

3.1.2. Translating Written English to ASR-English

The training data that includes audio is very limited and much larger resources are available for text-only systems. Especially useful for the translation of TED talks is the collection of movie subtitles in OPENSUBTITLES2018. For English-German, there is a huge amount of movie subtitles (roughly 22 million aligned sentences with over 170 million

Training data	Model	BLEU
TED-ASR-TOP-1	AMUN	16.65
TED-ASR-TOP-10	AMUN	16.28
TED-ASR-TOP-50	AMUN	15.88
TED-ASR-TOP-1	TRANSFORMER	18.25
TED-ASR-TOP-10	TRANSFORMER	17.90
TED-ASR-TOP-50	TRANSFORMER	18.14

Table 2: Translating the development test set with different models and different selections of ASR output and German translations from the parallel TED-TRANS training corpus.

tokens per language) that can be used to boost the performance of the NMT module.

The problem is, of course, that the subtitles come in regular language, and, again, we would see a mismatch between the training data and the ASR output in the speech translation pipeline. In contrast to approaches that try to normalize ASR output to reflect standard text-based MT input such as [13], we had the idea to transform regular English into ASR-like English using a translation model trained on a parallel corpus of regular TED talk transcriptions and the ASR output generated for the TED talks that we described in the previous section. We ran a number of experiments to test the performance of such a model. Some of the results are listed in Table 3.

Training data	Model	BLEU
TED-ASR-TOP-10	AMUN	61.87
TED-ASR-TOP-10	TRANSFORMER	61.91
TED-ASR-TOP-50	AMUN	61.82

Table 3: Translating English into ASR-like English using a model trained on TED-TRANS and tested on the development test set with original ASR output as reference.

As expected, the BLEU scores are rather high as the target language is the same as the source language, and we only mutate certain parts of the incoming sentences. The results show that there is not such a dramatic difference between the different setups (with respect to the model architecture and the data selection) and that a plain attentional sequence-to-sequence model with recurrent layers (AMUN) performs as well as a transformer model (TRANSFORMER) in this case. This makes sense, as we do not expect many complex long-distance dependencies that influence translation quality in this task. Therefore, we opted for the AMUN model trained on the top-10 ASR outputs, which we can decode efficiently in a distributed way on the CPU nodes of our computer cluster. With this we managed to successfully translate 99% of the entire SUBS collection from standard English into ASR-English. We refer to this set as SUBS-ASR.

We did a manual inspection on the result as well to see what the system actually learns to do. Most of the transformations are quite straightforward. The model learns to lowercase and to remove punctuation as our ASR output does not include it. However, it also does some other modifications that are more interesting from the viewpoint of an ASR module. While we do not have systematic evidence, Table 4 shows a few selected examples that show interesting patterns. First of all, it learns to spell out numbers (see *"2006"* in the first example). This is done consistently and quite accurately from what we have seen. Secondly, it replaces certain tokens with variants that resemble possible confusions that could come from a speech recognition system. The replacement of *"E.U."* with *"you"* and *"Stasi"* with *"stars he"* in these examples are quite plausible and rather surprising for a model that is trained on textual examples only. However, to conclude that the model learns some kind of implicit acoustic model would be a bit far-fetched, even though we would like to investigate the capacity of such an approach further in the future.

Original	**Because in the summer of 2006, the E.U. Commission tabled a directive.**
ASR-REF	because in the summer of two thousand and six the e u commission tabled directive
ASR-OUT	because in the summer of two thousand and six you commission tabled a directive

Original	**Stasi was the secret police in East Germany.**
ASR-REF	what is the secret police in east germany
ASR-OUT	stars he was the secret police in east germany

Table 4: Examples from the translations to ASR-like English. In the first column, ASR-REF refers to the top decoding hypothesis from the ASR model, while ASR-OUT is the output of the model translating the output to an ASR-like format.

In Section 4, we report on the effect of using synthetic ASR-like data on the translation pipeline.

3.2. Recasing Experiments

Our first attempt at a post-processing experiment involved using case-insensitive translation models, and deferring case restoration to a separate process unconditioned by the source side that we would apply after translation. We used the Moses toolkit [10] to train a recaser model on TED-TRANS. Afterwards, we re-trained a translation model on TED-ASR-TOP-10 and SUBS-ASR after lowercasing the training and validation sets, re-translated the development test set with this model, and then used the recaser to restore cases in the lowercase translations that we obtained. As shown in Table 5, evaluating the translations produced through these additional steps yielded scores that were very similar to those

obtained by the original case-sensitive translation models, and the result of this experiment was inconclusive.

Training data	BLEU	BLEU-LC
TED-ASR-TOP-10+SUBS-ASR	19.79	20.43
TED-ASR-TOP-10+SUBS-ASR-LC	19.73	20.91

Table 5: Case-sensitive models (TRANSFORMER) versus lowercased models with subsequent recasing. Recasing causes a larger drop than the model gains from training on lowercased training data. BLEU-LC refers to case-insensitive BLEU scores.

3.3. Reranking Experiments

In addition to using different subsets of the n-best lists output by the ASR model as additional training samples for the translation module, we also tried reranking alternatives using KenLM [14]. We initially generated a tokenized and lowercased version of TED-TRANS with all punctuation stripped, and then trained a language model on this set. We used this model to score and rerank samples in the 50-best lists, and then generated a new top-10 subset from this reranked version. However, when we re-trained translation models from these alternative sets, we observed that the model trained on the top-10 subsets before reranking exhibited a significantly better translation performance. We suspect that this is because, while the language model is useful for assessing the surface similarity of the ASR outputs to the source-side references, it was not uncommon for it to assign higher scores to ASR outputs that are semantically inconsistent with the target-side references, causing the NMT module to produce erroneous translations.

Similarly, we experimented with another language model trained on the target side of TED-TRANS, without the preprocessing. We intended this model to score and rerank outputs of the translation models, rather than the ASR module. To measure the effect of this language model, we fed the audio of our internal test set split through the ASR module, and produced 50-best lists for each sample. Afterwards, we used the language model to score and rerank the alternative transcripts for each sample produced by translating this set, and then selected the highest-scoring output for each sample. As in the previous language model experiment, employing this additional procedure significantly crippled the performance of our translation models.

4. Results

The results on development data reveal expected tendencies that we report below. First of all, as consistent with a lot of related literature, we can see a boost in performance when switching from a recurrent network model to the transformer model with multiple self-attention mechanisms. Table 6 shows a clear pattern of the superior performance of the transformer model that is also visible in additional runs that we do not list here. Secondly, we can see the importance of additional training data even if they come from slightly different domains. The vast amount of movie subtitles in OPENSUBTITLES2018 boosts the performance by about 3 absolute BLEU points. Note that the scores in Table 6 refer to models that do not use subtitles transformed into ASR-like English (SUBS-ASR) and which are not fine-tuned to TED talk translations.

Training data	Model	BLEU
TED-ASR-TOP-10	AMUN	16.28
TED-ASR-TOP-10+SUBS	AMUN	19.93
TED-ASR-TOP-10	TRANSFORMER	17.90
TED-ASR-TOP-10+SUBS	TRANSFORMER	20.44

Table 6: Model performance on the development test set when adding movie subtitles to the training data.

The effect of pre-processing by producing ASR-like English in the subtitle corpus is surprisingly negative. If we look at the scores in Table 7, we can see that the performance actually drops in all cases when considering only the untuned systems. We did not really expect that with the rather positive impression that we got from the manual inspection of the English-to-ASR translation discussed earlier. However, it is interesting to see the effect of fine-tuning. Fine-tuning here refers to a second training procedure that continues training with pure in-domain data (TED talks) after training the general model on the entire data set until convergence on validation data. Table 7 shows an interesting effect that may explain the difficulties of the integration of the synthetic ASR data. The fine-tuned model actually outperforms the model trained on standard data, which is due to a substantial jump from untuned models to the tuned version. The difference between those models with standard data is, on the other hand, only minor.

	BLEU	
Training data	Untuned	Tuned
TED-ASR-TOP-10+SUBS	20.44	20.58
TED-ASR-TOP-10+SUBS-ASR	19.79	20.80

Table 7: Training with original movie subtitles versus subtitles with English transformed into ASR-like English, before and after fine-tuning on TED-ASR-TOP-10 as pure in-domain training data (TRANSFORMER).

The synthetic ASR data look more similar to the TED-ASR data and, therefore, the model might get more confused between in-domain and out-of-domain data than it does for the model trained on the original subtitle data in connection with TED-ASR. Fine-tuning to TED-ASR brings the model

back on track again and synthetic ASR data becomes modestly beneficial.

Also of note is the contrast between the evaluation scores we obtained in development and those from the official test set. The translations we submitted obtain the BLEU scores shown in Table 8 on this year's test set.

Training data	BLEU
TED-ASR-TOP-10	14.34
TED-ASR-TOP-10+SUBS	16.45
TED-ASR-TOP-10+SUBS-ASR	15.80

Table 8: BLEU scores from our final models (TRANSFORMER)—respectively, the 2nd contrastive, 1st contrastive, and primary submission—on this year's test set. The scores from the two models with SUBS in their training data were obtained after fine-tuning on TED-ASR-TOP-10.

5. Conclusions

Apart from employing well-established practices such as normalization and byte-pair encoding as well as the benefits of using the transformer architecture, the only substantial boost to translation performance came from our data selection for the NMT module. The NMT module of our best-performing system on this year's test set was trained on TED-ASR-TOP-10 and the raw SUBS, and later fine-tuned on TED-ASR-TOP-10.

Although we ran many experiments to improve various steps of our speech translation pipeline, their influence on translation performance has been marginal at best. The effects of training with different TED-ASR subsets were hard to distinguish. While using SUBS-ASR in training seemed to provide a modest improvement in development, this effect was not carried over to the final results on the test set. The later experiments with lowercasing and recasing had an ambiguous effect, and those with reranking had a noticeably negative outcome.

In future work, our aim is to further investigate what factors in a good speech translation model, and continue experimenting in relation to these on the NMT module. We will also try to improve our TDNN-HMM ASR module by replacing the n-grams with an RNNLM, and try see how our complete end-to-end speech-to-text translation model performs after having sufficient training time.

6. Acknowledgements

This work has been supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No 780069, and by the Academy of Finland in the project 313988. In addition the Finnish IT Center for Science (CSC) provided computational resources.

7. References

[1] 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.

[2] A. Rousseau, P. Deléglise, and Y. Esteve, "TED-LIUM: an automatic speech recognition dedicated corpus," in *LREC*, 2012, pp. 125–129.

[3] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, "Marian: Fast neural machine translation in C++," in *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 116–121. [Online]. Available: http://www.aclweb.org/anthology/P18-4020

[4] 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.

[5] J. Tiedemann, "News from OPUS - A collection of multilingual parallel corpora with tools and interfaces," in *Recent Advances in Natural Language Processing*, N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, Eds. Borovets, Bulgaria: John Benjamins, Amsterdam/Philadelphia, 2009, vol. V, pp. 237–248.

[6] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4960–4964.

[7] G. Neubig, M. Sperber, X. Wang, M. Felix, A. Matthews, S. Padmanabhan, Y. Qi, D. S. Sachan, P. Arthur, P. Godard, *et al.*, "XNMT: The extensible neural machine translation toolkit," *arXiv preprint arXiv:1803.00188*, 2018.

[8] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech*, 2016, pp. 2751–2755.

[9] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina, *et al.*, "State-of-the-art speech recognition

with sequence-to-sequence models," *arXiv preprint arXiv:1712.01769*, 2017.

[10] 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.

[11] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *ACL16*, 2015.

[12] 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

[13] E. Matusov, A. Mauser, and H. Ney, "Automatic sentence segmentation and punctuation prediction for spoken language translation," in *International Workshop on Spoken Language Translation*, Kyoto, Japan, Nov 2006, pp. 158–165.

[14] K. Heafield, "KenLM: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2011, pp. 187–197.

Prompsit's Submission to the IWSLT 2018 Low Resource Machine Translation Task

Víctor M. Sánchez-Cartagena

Prompsit Language Engineering
Av. Universitat s/n. Edifici Quorum III
E-03202 Elx, Spain
vmsanchez@prompsit.com

Abstract

This paper presents Prompsit Language Engineering's submission to the IWSLT 2018 Low Resource Machine Translation task. Our submission is based on cross-lingual learning: a multilingual neural machine translation system was created with the sole purpose of improving translation quality on the Basque-to-English language pair. The multilingual system was trained on a combination of in-domain data, pseudo in-domain data obtained via cross-entropy data selection and backtranslated data. We morphologically segmented Basque text with a novel approach that only requires a dictionary such as those used by spell checkers and proved that this segmentation approach outperforms the widespread byte pair encoding strategy for this task.

1. Introduction

This paper presents Prompsit Language Engineering's submission to the IWSLT 2018 Low Resource Machine Translation task. The objective of this task is building an MT system for translating TED talks from Basque to English from a very limited amount of in-domain Basque–English parallel data. We relied on cross-lingual learning via a multilingual approach [1] to neural machine translation (NMT), extraction and cleaning of pseudo in-domain parallel text from out-of-domain data, and backtranslation of Spanish text into Basque for building our submission.

Moreover, we applied morphological segmentation to the Basque text. We took advantage of an existing spell checking dictionary and its inflection paradigms and used an automatic morphology inference model to decide between ambiguous segmentations. We proved that this method, that requires shallower linguistic information[1] than other segmentation approaches based on full morphological analysis and disambiguation [2, 3], outperforms the widespread byte pair encoding (BPE) segmentation strategy [4] in terms of translation quality for Basque-to-English NMT.

[1]Neither part of speech/morphological information in the dictionary nor a part of speech tagger/parser are needed. In principle, this approach could be applied to any language for which a Hunspell-based (`http://hunspell.github.io/`) spell checker exists.

Table 1: *Size of in-domain data. Processed segments are those that remain after removing talks included in the development and test sets.*

Language pair	# raw segments	# processed segments
eu-en	5 687	5 687
eu-es	6 742	5 610
eu-fr	7 021	5 878
es-en	280 947	279 737
fr-en	290 961	289 722

The remainder of the paper explains the steps followed to build the submitted NMT system. Next section explains how the in-domain and out-of-domain parallel corpora were processed and filtered, while Section 3 focuses on describing and assessing the impact of the morphological segmentation approach followed. Section 4 describes the NMT architecture and training process. Section 5 depicts the process followed to obtain the data set used to train our submission. Finally, the most relevant related approaches are reviewed in Section 6 and the paper ends with some concluding remarks.

2. Data acquisition and cleaning

Our submission was trained on a combination of in-domain and out-of-domain data. The only special cleaning applied to the in-domain training data provided by the organization is the removal of talks that are also included in the test/development sets. Table 1 shows the number of segments in the in-domain data for each language pair before and after removing such talks.

Following the shared task instructions,[2] we built the out-of-domain data collection by downloading all the corpora available from the Opus [5] and WMT [6] websites.[3] We also included the Basque–Spanish parallel data from Open

[2]`https://sites.google.com/site/iwsltevaluation2018/TED-tasks`

[3]If the same corpus was available from both websites (e.g. Europarl), we downloaded it from WMT. If the same corpus was available from different WMT editions, we downloaded it from the most recent one. We skipped some corpora from Opus which were too noisy, like EUBookshop.

Table 2: *Size of out-of-domain data before and after applying shallow cleaning.*

Language pair	# raw segments	# clean segments
eu-en	1.81M	928K
eu-es	1.64M	1.41M
eu-fr	711K	375K

Table 3: *Size of out-of-domain data before and after applying aggressive cleaning.*

Language pair	# raw segments	# clean segments
es-en	163M	65M
fr-en	164M	77M

Data Euskadi Repository published by the task organizers.

We followed two different strategies for out-of-domain parallel data cleaning. For language pairs with limited data availability, namely those including Basque, we followed a conservative shallow cleaning strategy since removing correct segments can be harmful for the quality of the final system. For the remaining language pairs, since only a subset of the data is finally used (see Section 5), we applied a more aggressive cleaning strategy.

The shallow cleaning consisted in deduplication and removal of parallel segments that meet any of the following conditions: they contain a low proportion of alphabetic characters, their source-language (SL) and target-language (TL) side are very similar (there is a low edit distance between them), they are too long or too short (shorter than 3 tokens or longer than 100), or they are written in another language (language is detected by means of `cld2`[4] and segments are only discarded when the language detection is *reliable* according to the `cld2` algorithm). Table 2 shows the size of the out-of-domain data for each language pair containing Basque before and after applying shallow cleaning.[5]

The aggressive cleaning consisted in two steps. Firstly, parallel segments were deduplicated and a more aggressive superset of the rules used in the shallow cleaning (implemented in the translation memory cleaning tool `Bicleaner`[6]) was applied. These rules are addressed at detecting evident flaws such as encoding errors, very different lengths in parallel segments, etc. Secondly, misaligned segments were detected and removed by means of an automatic classifier, described in [7]. The classifier is also part of the `Bicleaner` tool. Pre-trained models for the classifier were obtained from the Paracrawl project.[7] Table 3 shows the size of the out-of-domain data for each language pair before and after applying the aggressive cleaning.

[4] `https://github.com/CLD2Owners/cld2`

[5] Shallow cleaning was not applied to the Basque–Spanish parallel data from Open Data Euskadi Repository.

[6] `https://github.com/bitextor/bicleaner`

[7] `https://github.com/bitextor/bitextor-data/tree/master/bicleaner`

3. Morphological segmentation for Basque

Word segmentation based on linguistically-informed strategies such as morphological analysis [2] or simpler alternatives based on lists of relevant prefixes and suffixes [8] have shown to be able to outperform the popular BPE approach [4] for some agglutinative and highly inflected languages. In this section, we present the pseudo-morphological segmentation approach based on inflection paradigms we applied to Basque text in our submission and prove that it outperforms BPE.

3.1. Pseudo-morphological segmentation based on inflection paradigms

Inflection paradigms are commonly used in dictionaries (morphological dictionaries used in rule-based machine translation, spell checkers, etc.) in order to group regularities in the inflection of a set of words.[8] A paradigm is usually defined as a collection of suffixes and, optionally, their corresponding part-of-speech/morphological information; e.g., the paradigm assigned to many common English verbs indicates that by adding the suffix *-ing* to the stem, the gerund is obtained; by adding the suffix *-ed*, the past is obtained; etc. While morphological dictionaries from rule-based machine translation systems contain morphological information, spell checkers usually lack this information.

In languages with a high inflection degree, such as Basque, a surface form can be built by sequentially appending suffixes from different paradigms to a stem. For instance, the word *etxekoak* can be generated from the entry $etxe + PAR240$ if paradigm $PAR240$ contains the suffixes $-ko + PAR243$, $-z$, $-rekin$, etc. and paradigm $PAR243$ contains the suffix $-ak$.

As suffixes contained in inflection paradigms are usually based on linguistic knowledge, one can take advantage of inflection paradigms for splitting words for training NMT systems. In this way, words can be split in atomic units of meaning or *morphs*. For instance, in previous example, *etxekoak* (the plural form of "domestic") would be split into *etxe* ("house"), *-ko* (adjectivation) and *-ak* (plural mark).

In order to split a corpus using inflection paradigms, there are two types of words for which an additional strategy needs to be devised:

- Homograph words: those that can be generated by multiple combinations of stem and suffix(es).
- Unknown words: those that are not present in the morphological dictionary/spell checking dictionary.

In order to decide the best segmentation for these words, we took advantage of semi-supervised morphology learning methods. In particular, we used Morfessor [9]. Morfessor is

[8] Paradigms ease dictionary management by reducing the quantity of information that needs to be stored, and by simplifying revision and validation because of the explicit encoding of regularities in the dictionary.

a family of methods for automatic learning of morphology based on the minimum description length principle [10]: the words in a corpus are split in morphs in such a way that the size of the morph vocabulary and the length in tokens of the corpus are minimized. We used a semi-supervised variant of Morfessor in which the segmentation model can be estimated from a plain corpus and a set of already segmented words [11].

Our pseudo-morphological segmentation strategy comprises the following steps:

1. Segment words encoded in the morphological dictionary/spell checker which have only a candidate segmentation according to the inflection paradigms.

2. Train a Morfessor segmentation model in an semi-supervised way [11] from the Basque corpus we want to segment and the words segmented in the previous step.

3. Segment homograph words by choosing the segmentation with the highest likelihood according to the previous model.

4. Segment unknown words by choosing the segmentation with highest likelihood according to the model among those that can be generated by using solely suffixes from the inflection paradigms in the morphological dictionary/spell checker.

This approach hence allows us to segment a corpus in atomic units of meaning using a spell checker as the only linguistic resource. Unlike other approaches to NMT training corpus word segmentation based on linguistic information [2, 3], this approach does not require neither a full morphological analyzer with part-of-speech/morphological tags nor a part-of-speech tagger/parser for disambiguating between the different analyses of each word. Part-of-speech/morphological information (e.g. the fact the suffix *-ed* represents the past tense of a verb) is not used during the process and disambiguation is carried out by the Morfessor model which, in turn, controls the growth of vocabulary size.

In our submission, we used the Basque spell checker *Xuxen v5.1* as dictionary.[9] Moreover, following [8], we applied BPE splitting with a model learned on the concatenation of all training corpora after performing the pseudo-morphological segmentation. Note that applying BPE to further split the word pieces obtained after pseudo-morphological segmentation helps the system to translate proper nouns and compounds in Basque.

3.2. Evaluation

We evaluated the pseudo-morphological segmentation approach we employed in our submission and compared it with two baselines: a greedy alternative in which the segmentation with the most frequent stem is chosen for unknown and homograph words, and plain BPE splitting. In all cases, BPE was applied to all the languages of the training corpus (65 000 operations) and the model was learned from their concatenation after carrying out pseudo-morphological segmentation (except for the plain BPE system, for which morphological segmentation was not carried out).

[9] `https://xuxen.eus`

Table 4: *Results of the evaluation of the pseudo-morphological segmentation approach proposed, a greedy alternative, and plain BPE.*

Segmentation strategy	BLEU	TER
BPE	12.75	83.68
Paradigms/Greedy+BPE	13.28	87.80
Pseudo-morph+BPE	13.59	79.73

We trained multilingual NMT systems as described in Section 4 on parallel corpora segmented following the three strategies. The three multilingual NMT systems were trained on the in-domain data and included the language pairs Spanish–English, French–English, Basque–English, Basque–French and Basque–Spanish.

The evaluation was carried out only on the Basque-to-English direction. The values of the translation evaluation metrics BLEU [12] and TER [13] computed on the development set are reported in Table 4. We can observe that our pseudo-morphological segmentation approach (`Pseudo-morph+BPE`) outperforms both plain BPE segmentation and segmentation based on paradigms with a greedy strategy for homograph and unknown words.

Table 5 shows several examples of words segmented by the three alternatives evaluated. Furthermore, Table 6 depicts three Basque sentences from the development set, how they were segmented by the three alternatives evaluated and their translation with the NMT systems built. Note that, unlike the words in Table 5, the SL sentences in Table 6 were split with BPE after applying the splitting strategies based on inflection paradigms, as described previously in this section. In the first example, the Basque word *konpartimentutan* is formed by the stem *konpartimentu*, which means "compartment", plus the inessive suffix *-tan*). The segmentation strategies based on inflection paradigms are able to correctly detach the inessive suffix from the word, while the pure BPE approach fails to do it. As a consequence, the MT system built using the latter approach is not able to produce an adequate translation by taking advantage of the sentences in the training corpus that contain words starting with *konparti-*. Similarly, in the second example, the segmentation strategies based on inflection paradigms are able to segment *estudioa* into the stem *estudio* (that means "studio apartment") and the suffix *-a* (singular article). The pure BPE approach segments it into *estudi-* and *-oa*. Since *estudi-* is the stem of the verb "to study" in Spanish, the multilingual system wrongly generates that verb in the translation into English. Finally, in the third example, the greedy approach based on paradigms wrongly segments *Asia* into *as* and *-ia*, which prevents the NMT system from

Table 5: *Examples of Basque words segmented by the three approaches evaluated. The segmentation that best splits the word in atomic units of meaning is shown in bold.*

Word	BPE	Paradigms/Greedy	Pseudo-morph	meaning
adierazitako	adierazitako	adieraz itako	**adierazi tako**	"expressed", built from *adierazi* ("to express") plus *-tako* (suffix used in relative clauses)
izendatu	**izendatu**	izenda tu	**izendatu**	"nominate", atomic unit
ebaluaketa	ebalu aketa	ebaluaket a	**ebaluaketa**	"evaluation", atomic unit
birgaitzeko	bir gaitzeko	**birgaitze ko**	**birgaitze ko**	"rehabilitation" (*birgaitze*) plus genitive suffix (*-ko*)

producing the word *Asia* in English.

4. Training strategy

Our submission is based on cross-lingual learning. We aimed at improving the translation performance on the Basque-to-English language pair by means of the addition of training data from other language pairs. The different language pairs were combined by means of a multilingual NMT approach [1]. A TL marker was prepended to each SL segment. See Section 5 for more details about language pairs included and how the data for each of them was obtained.

Our submitted NMT system follows the Transformer architecture [14]. In particular, we used the implementation in the Marian NMT toolkit [15]. We generally used the hyperparameters of the *Transformer base model* [14], with the exception of *warmup_steps*, which was set to 16 000 instead of 4 000. This parameter was increased because our minibatch size was significantly smaller than that used in the original paper [14]. We limited segment length to 100 tokens and let the Marian toolkit set the batch size to fit 8 000 MiB of GPU memory. For a vocabulary size of around 70 000 words, the number of TL words in a minibatch was around 3 000, while [14] report 25 000 TL words per minibatch. A checkpoint was saved every 5 000 updates.

We used only the publicly released Basque–English *IWSLT18.TED.dev2018* corpus as a development set.[10] Training ended when perplexity on the development set did not improve in 10 consecutive checkpoints. We selected the checkpoint that obtained the highest BLEU score on the development set.

Concerning corpora preprocessing, text was tokenized with the `aggressive` strategy[11] implemented by the OpenNMT tokenizer [16]. Words were split in sub-word units as described in Section 3. The Morfessor model was trained on the concatenation of the Basque section of the training data for all language pairs that contained Basque. The BPE model (65 000 operations), which shared by all SLs and TLs, was learned from the concatenation of the morphologically segmented Basque data and the unsegmented data for the remaining languages and it was used to split these corpora. Text was lowercased prior to training and the resulting English translations were recased[12] with a recasing model estimated from the concatenation of the English side of the training corpora.

[10] It could be interesting to study whether using development data from other language pairs has a significant impact in translation quality for Basque–English.

[11] The only multi-character tokens allowed are sequences of strictly alphabetical characters.

5. Training data

This section describes the training data from which our submission was built and the experiments carried out to select it.

5.1. Language pairs

According to the experiments carried out by [1], including new language pairs that share either the SL or the TL with the language pair of interest helps to increase the translation quality for that language pair. Henceforth, our multilingual system contains only language pairs with Basque as SL or English as TL. Moreover, we included only language pairs for which the training set is published as part of this year's data. Hence, our multilingual system contains data from the Spanish–English, French–English, Basque–English, Basque–French and Basque–Spanish language pairs. Preliminary experiments showed no important gains when adding data from the German–English and Turkish–English language pairs to the training collection. Conducting more exhaustive experiments has been left as future work.

5.2. Cross-entropy data selection and oversampling

As shown in Table 3, there is a huge amount of out-of-domain parallel data available for the Spanish–English and French–English language pairs. If it was just concatenated to the in-domain data, the system would be biased towards the out-of-domain data. In order to avoid that issue, we selected only a subset of the out-of-domain data which is similar to the in-domain one (from now on, *pseudo in-domain data*) via cross-entropy difference [17].

The process was carried out as follows. Firstly, we sorted the out-of-domain data (after cleaning it as described in Section 2) by monolingual cross-entropy difference on the English side. The in-domain language model was estimated

[12] The Moses recaser was used: `http://www.statmt.org/moses/?n=Moses.SupportTools#ntoc10`.

Table 6: Result of applying each of the three segmentation strategies evaluated in Section 3 to a three sentences extracted from the development set. The translation of each sentence with a multilingual NMT system trained only on the in-domain data is also depicted. The character → at the end of a token implies that it is a sub-word unit originally attached to the token that follows it. Words whose segmentation has a visible impact on the translation are shown in bold.

#	segmentation strategy	sentence
1	source – BPE	burmu→ ina ez dago **kon**→ **parti**→ **men**→ **tutan** ban→ atuta .
	source – Paradigms/Greedy+BPE	bur→ mu→ in→ a ez dago **kon**→ **parti**→ **mentu**→ **tan** bana→ tuta .
	source – Pseudo-morph+BPE	bur→ mu→ in→ a ez dago **kon**→ **parti**→ **mentu**→ **tan** bana→ tuta .
	translation – BPE	There's no brain at all based on **bias**.
	translation – Paradigms/Greedy+BPE	You don't have a brain that's broken up into **blocks**.
	translation – Pseudo-morph+BPE	There's no **boundary** in the brain.
	reference	The brain isn't divided into **compartments**.
2	source – BPE	beraz urte batez **estudi**→ **oa** ix→ tea erabaki nuen .
	source-Paradigms/Greedy+BPE	bera→ z urte bat→ ez **estudio**→ **a** ix→ te→ a erabaki nu→ en .
	source – Pseudo-morph+BPE	beraz urte bat→ ez **estudio**→ **a** ix→ te→ a erabaki nuen .
	translation – BPE	So I decided to **study** for a year.
	translation – Paradigms/Greedy+BPE	So one year I decided to give it a try.
	translation – Pseudo-morph+BPE	So I decided to stay silent for a year.
	reference	So I decided to close it down for one year.
3	source – BPE	beraz **asia** aukeratu nuen .
	source – Paradigms/Greedy+BPE	bera→ z **as**→ **ia** aukera→ tu nu→ en .
	source – Pseudo-morph+BPE	beraz **asia** aukeratu nuen .
	translation – BPE	So I chose **Asia**.
	translation – Paradigms/Greedy+BPE	So I decided to give it a try.
	translation – Pseudo-morph+BPE	So I chose **Asia**.
	reference	So **Asia** it was.

from the English side of the parallel in-domain Spanish–English training corpus, while the out-of-domain one was obtained from a random sample with the same number of segments from the English side of all the available Spanish–English parallel data. The same language models were used for computing monolingual cross-entropy difference for both the Spanish–English and French–English language pairs. As other authors did previously [18], we split English corpora with BPE prior to training the language models and scoring the out-of-domain parallel segments.

Secondly, we carried out a set of experiments in order to decide which is the most appropriate amount of pseudo in-domain data for Spanish–English and French–English. In these experiments, we used all the available data for Basque–English, Basque–Spanish and Basque–French, and varying amounts of pseudo in-domain data, which was concatenated to the real in-domain data, for Spanish–English and French–English. In addition, we also studied the effect of oversampling the Basque–English data (concatenation of in-domain and out-of-domain) to match the size of the Spanish–English and French–English data.

Table 7 depicts the size of the pseudo in-domain parallel data[13] and the size of the Basque–English data included in the training set for the different configurations evaluated, together with the values of the evaluation metrics BLEU [12] and TER [13] computed on the development set. The original size of the Basque–English data is 933 356 segments. Those rows with values higher than 0.9M imply that the data the Basque–English has been oversampled. In other words, it has been included as many times as necessary for reaching the size depicted in the table. Systems were trained following the set-up described in Section 4. For the same data configurations, Table 8 shows automatic evaluation metrics computed after finetuning the systems on the in-domain data.[14] Results show no important gains when increasing the out-of-domain data size from 2M to 5M and confirm the importance of oversampling, in line with the results reported in [1]. Finetuning on in-domain data did not bring any positive impact. One possible reason could be the scarce amount of in-domain Basque–English data available (see Table 1). We chose the configuration with the highest BLEU score on the development set (depicted in bold in Table 7) for our submission.

5.3. Backtranslation

Backtranslation, that is, the translation of additional TL monolingual data into the SL with an MT system in order to obtain additional training material, is a widespread method to enhance the quality of NMT systems [19].

[13] For a given size N, the N parallel segments with the lowest cross-entropy score are selected from the out of domain data.

[14] When finetuning, the initial learning rate was set to the value employed in the last update of the main training process.

Table 7: *Results of the experiments carried out in order to determine the best size for pseudo in-domain data and for Basque–English data (with oversampling). Unlike the experiments depicted in Table 8, these experiments did not include finetuning on the in-domain data at the end of the training process. The configuration highlighted in bold is the one used in our submission.*

Pseudo in-domain size	eu-en size	BLEU	TER
2M	0.9M	21.05	68.15
2M	**2M**	**21.72**	**68.65**
5M	0.9M	19.47	71.86
5M	3M	20.46	70.17
5M	5M	21.10	68.95

Table 8: *Results of the experiments carried out in order to determine the best size for pseudo in-domain data and for Basque–English data (with oversampling). Unlike the experiments depicted in Table 7, these experiments included finetuning on the in-domain data at the end of the training process.*

Pseudo in-domain size	eu-en size	BLEU	TER
2M	0.9M	21.15	69.11
2M	2M	21.68	68.46
5M	0.9M	19.96	71.20
5M	3M	20.88	71.46
5M	5M	21.68	69.38

In our submission, we did not directly translate monolingual English data into Basque. Since there is high-quality Basque–Spanish parallel data not available for Basque–English (Open Data Euskadi Repository) we opted for translating the Spanish side of Spanish–English parallel data into Basque in order to build additional Basque–English training material. A similar approach has been successfully applied for enhancing phrase-based statistical machine translation systems [20].

In order to carry out the backtranslation, we trained an NMT system on all the available Spanish–Basque data with the set-up described in Section 4. Words were segmented as described in Section 3. That system was used to backtranslate the Spanish side of the in-domain Spanish–English training data and the top 5M segments[15] from the pseudo in-domain Spanish–English corpus.

We evaluated the impact of adding backtranslated data to the best dataset from the previous section (2M pseudo in-domain parallel segments, oversampling and no finetuning). We built NMT systems after adding the full backtranslated data (both the in-domain and the pseudo-in-domain data; row labeled as *5.2M*), and after adding the in-domain and only 2M pseudo-in-domain backtranslated segments (row labeled as *2.2M*). Results of the evaluation on the development set of the NMT systems trained with these data are depicted in Table 9. They show that using the whole backtranslated data has a strong positive impact on the quality of the resulting MT system. Hence, we used the 5.2M backtranslated segments in our submission.

[15]We could not backtranslate a larger amount of data because of time restrictions.

Table 9: *Results of the experiments carried out in order to determine the best size for backtranslated data. The configuration highlighted in bold is the one used in our submission.*

Size of backtranslated data	BLEU	TER
0	21.72	68.65
2.2M	22.51	67.54
5.2M	**23.45**	**66.94**

5.4. Final submission

Our final submission was trained on the best data collection from previous section. We experimented with finetuning and checkpoint ensembling [21, Sec. 3.2], but translation quality did not improve. Hence, we submitted just the result of translating the test set with the intermediate model that achieved the highest BLEU score on the development set.

6. Related approaches

Our submission is built with the help of morphological segmentation, cross-entropy data selection and cross-lingual learning via multilingual NMT. This section reviews the most relevant approaches in these three fields.

Morphological segmentation has been successfully applied to build a winning system [2] for the English–Finnish language pair in the WMT 2016 news translation shared task [22]. Simpler alternatives based on lists of prefixes/suffixes have also been reported to bring improvements in translation quality [8]. Morphological segmentation has already been applied to NMT for Basque [23]. However, unlike our approach, their strategy segments homograph words in a greedy way (longest stem). Besides morphological segmentation, there are other ways linguistic resources can be used to segment words for NMT training. For instance, TL words can be transformed into a sequence of stem and morphological inflection tags in order to achieve better morphological generalization when translating into highly inflected languages [3].

Cross-entropy data selection [17] has become a popular approach for leveraging out-of-domain data when building MT systems. This strategy has been used for collecting training data for phrase-based statistical machine translation systems [24] and NMT systems [18] in shared translation tasks such as WMT [6] and IWSLT [25].

In multilingual NMT [1], a single NMT model is used to translate between different language pairs. Some authors proposed multilingual NMT strategies in which the underly-

ing network architecture does not need to be modified [1, 26]. That property allowed us to perform multilingual MT with a Transformer [14] model despite the fact that the multilingual NMT approach we followed [1] was originally addressed to the encoder-decoder with attention architecture [27]. On the contrary, other authors [28] proposed modifying the network architecture to use an independent encoder and decoder for each language.

7. Concluding remarks

This paper presented Prompsit Language Engineering's submission to the IWSLT 2018 Low Resource MT track. We presented a novel method for morphological segmentation based solely on a dictionary with inflection paradigms such as those used by spell checkers and proved that it outperforms the widespread BPE segmentation method. Our submission relies on cross-lingual learning via multilingual NMT. Basque training data was segmented with the novel method. The NMT system follows the Transformer architecture. We experimented with varying amounts of pseudo in-domain data obtained via cross-entropy data selection and with varying amounts of backtranslated data and submitted the combination that maximized translation quality on the development set.

Our submission could be further improved with independent ensembles [21, Sec. 3.2]. The inclusion of additional language pairs has not been exhaustively evaluated and the quality of the final system might be improved by adding some more language pairs. The quality of the final system could also improve with the addition of more backtranslated data.

8. Acknowledgements

We would like to thank Prof. Mikel L. Forcada for the advice on Basque segmentation. Work supported by project IADAATPA, action number 2016-EU-IA-0132, funded under the Automated Translation CEF Telecom instrument managed by INEA at the European Commission.

9. 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," *Transactions of the Association of Computational Linguistics*, vol. 5, no. 1, pp. 339–351, 2017.

[2] V. M. Sánchez-Cartagena and A. Toral, "Abu-matran at WMT 2016 translation task: Deep learning, morphological segmentation and tuning on character sequences," in *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, vol. 2, 2016, pp. 362–370.

[3] A. Tamchyna, M. Weller-Di Marco, and A. Fraser, "Modeling target-side inflection in neural machine translation," in *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 32–42. [Online]. Available: http://www.aclweb.org/anthology/W17-4704

[4] 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)*, vol. 1, 2016, pp. 1715–1725.

[5] J. Tiedemann, "Parallel Data, Tools and Interfaces in OPUS," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, N. C. C. Chair), K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds. Istanbul, Turkey: European Language Resources Association (ELRA), may 2012.

[6] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, *et al.*, "Findings of the 2017 conference on machine translation (WMT17)," in *Proceedings of the Second Conference on Machine Translation*, 2017, pp. 169–214.

[7] V. M. Sánchez-Cartagena, M. Bañón, S. Ortiz-Rojas, and G. Ramírez-Sánchez, "Prompsit's submission to WMT 2018 Parallel Corpus Filtering shared task," in *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*. Brussels, Belgium: Association for Computational Linguistics, October 2018.

[8] M. Huck, S. Riess, and A. Fraser, "Target-side word segmentation strategies for neural machine translation," in *Proceedings of the Second Conference on Machine Translation*, 2017, pp. 56–67.

[9] S. Virpioja, P. Smit, S.-A. Grönroos, and M. Kurimo, "Morfessor 2.0: Python implementation and extensions for morfessor baseline, D4 Julkaistu kehittmis- tai tutkimusraportti tai -selvitys," 2013. [Online]. Available: http://urn.fi/URN:ISBN:978-952-60-5501-5

[10] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465 – 471, 1978. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005109878900055

[11] O. Kohonen, S. Virpioja, and K. Lagus, "Semi-supervised learning of concatenative morphology," in *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, ser. SIGMORPHON '10. Stroudsburg,

PA, USA: Association for Computational Linguistics, 2010, pp. 78–86. [Online]. Available: http://dl.acm.org/citation.cfm?id=1870478.1870488

[12] 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.

[13] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of association for machine translation in the Americas*.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[15] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, "Marian: Fast neural machine translation in C++," in *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 116–121. [Online]. Available: http://www.aclweb.org/anthology/P18-4020

[16] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "Opennmt: Open-source toolkit for neural machine translation," *Proceedings of ACL 2017, System Demonstrations*, pp. 67–72, 2017.

[17] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 355–362. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145474

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

[19] 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)*, vol. 1, 2016, pp. 86–96.

[20] M. Huck and H. Ney, "Pivot lightly-supervised training for statistical machine translation," in *Proc. 10th Conf. of the Association for Machine Translation in the Americas*, 2012, pp. 50–57.

[21] 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," in *Proceedings of the Second Conference on Machine Translation*, 2017, pp. 389–399.

[22] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz, *et al.*, "Findings of the 2016 conference on machine translation," in *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, vol. 2, 2016, pp. 131–198.

[23] T. Etchegoyhen, E. Martínez Garcia, A. Azpeitia, G. Labaka, I. Alegria, I. Cortes Etxabe, A. Jauregi Carrera, I. Ellakuria Santos, M. Martin, and E. Calonge, "Neural Machine Translation of Basque," in *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, Alacant, Spain, 2018, pp. 139–148.

[24] R. Rubino, A. Toral, V. M. Sánchez-Cartagena, J. Ferrández-Tordera, S. O. Rojas, G. Ramírez-Sánchez, F. Sánchez-Martínez, and A. Way, "Abu-MaTran at WMT 2014 translation task: Two-step data selection and RBMT-style synthetic rules," in *Proceedings of the ninth workshop on statistical machine translation*, 2014, pp. 171–177.

[25] C. Mauro, F. Marcello, B. Luisa, N. Jan, S. Sebastian, S. Katsuitho, Y. Koichiro, and F. Christian, "Overview of the IWSLT 2017 Evaluation Campaign," in *International Workshop on Spoken Language Translation*, 2017, pp. 2–14.

[26] T.-L. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," in *Proceedings of 2016 International Workshop on Spoken Language Translation*, 2016.

[27] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[28] O. Firat, K. Cho, and Y. Bengio, "Multi-way, multilingual neural machine translation with a shared attention mechanism," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 866–875.

Neural Speech Translation at AppTek

Evgeny Matusov, Patrick Wilken, Parnia Bahar[*]*, Julian Schamper*[*]*, Pavel Golik, Albert Zeyer*[*]*,*
Joan Albert Silvestre-Cerdà[+]*, Adrià Martínez-Villaronga*[+]*, Hendrik Pesch, and Jan-Thorsten Peter*

Applications Technology (AppTek), Aachen, Germany

{ematusov,pwilken,pbahar,jschamper,pgolik,jsilvestre,amartinez,hpesch,jtpeter}@apptek.com

[*]Also RWTH Aachen University, Germany [+]Also Universitat Politècnica de València, Spain

Abstract

This work describes AppTek's speech translation pipeline that includes strong state-of-the-art automatic speech recognition (ASR) and neural machine translation (NMT) components. We show how these components can be tightly coupled by encoding ASR confusion networks, as well as ASR-like noise adaptation, vocabulary normalization, and implicit punctuation prediction during translation. In another experimental setup, we propose a direct speech translation approach that can be scaled to translation tasks with large amounts of text-only parallel training data but a limited number of hours of recorded and human-translated speech.

1. Introduction

AppTek participated in the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2018 with the goal of obtaining best possible speech translation quality by streamlining the interface between ASR and machine translation (MT). We tested a new way of encoding multiple hypotheses of ASR as input to an NMT system. We also experimented with a novel direct neural translation model that translates source language speech into target language text, while at the same time benefiting from text-only parallel training data in a multi-task learning framework. To make these experiments possible, we made sure that our NMT system can handle different types of input, and its source language vocabulary is harmonized w.r.t. the ASR system vocabulary. We also fine-tuned the NMT model on ASR-like noise, making it more robust against recognition errors. Finally, we tested different punctuation prediction approaches and found that the implicit prediction of punctuation marks by the MT component works best in our setting.

Although improving our state-of-the-art NMT model was not our main focus, we benefited from fine-tuning the model on the in-domain data, as well as from ensembles of models which differ in architecture – recurrent neural network (RNN) model with attention [1] or transformer architecture [2] – and/or input modality – ASR confusion network (CN) or first-best ASR output. This paper is organized as follows. We start by reviewing related work in 2, pointing out some differences and novelties in our approach. In Section 3, we describe our methods for data filtering, preprocessing, and punctuation prediction. Section 4 gives an overview of our ASR system. Section 5 describes the details of AppTek's NMT system. Section 6 gives details of how ASR confusion networks can be encoded as an input of the NMT system. In Section 7, we describe our direct speech translation prototype. The results of our speech translation experiments are summarized in Section 8.

2. Related Work

Theoretical background for tighter coupling of statistical ASR and MT systems had been first published in [3]. In practice, it was realized e. g. as statistical phrase-based translation of ASR word lattices with acoustic and language model scores [4] or confusion networks with posterior probabilities [5]. In both cases moderate improvements of translation quality were reported when the ASR scores were included in the log-linear model combination; the improvements were larger when the baseline recognition quality was low.

In the first publication on word lattice translation using a neural model [6], the proposed lattice-to-sequence model had an encoder component with one hidden state for each lattice node, as well as attention over all lattice nodes. This is a different and more computationally expensive model as compared to what we propose in this work. In our encoder, the number of hidden states is the same as the number of slots in the input confusion network, which is usually only slightly higher than the number of words in the utterance.

Adapting the NMT system to ASR-like noise was proposed by [7]. We follow the same strategy, but the noise that we introduce is not random; it is sampled from a distribution of most common ASR errors based on statistics from recognizing the audio of the TED dataset.

Direct translation of foreign speech was proposed by [8], who used a character-level sequence-to-sequence model[1]. They report experimental results on a small (163 hours of speech with transcriptions and translations) Spanish-to-English Fisher and Callhome dataset. The authors use multi-task learning with a single speech encoder and two decoders, one for English (direct translation) and one for Spanish, which allows them to incorporate supervision from Spanish

[1]A later work by [9] extends the approach of [8] to word-level models.

transcripts. In contrast, we follow the opposite approach, in which we have a single target language decoder and two separate encoders, one for source language speech, and one for source language text. This approach allows us to benefit from large quantities of text-only parallel MT training data, in a multi-task learning scenario, and thus, in contrast to previous work, to potentially compete with the standard approach that uses strong, but separate components for ASR and MT.

Punctuation prediction in MT (and especially neural MT) context was investigated in comparative experiments in [10]. Similarly to that paper, we also confirmed experimentally that implicit prediction of punctuation marks by the NMT system resulted in the best BLEU and TER scores in our setting (see Section 3.3 for details).

3. Data Preparation

3.1. Parallel Data Filtering

In line with the evaluation specifications, we used the TED corpus, the OpenSubtitles2018 corpus [11], as well as the data provided by the WMT 2018 evaluation (Europarl, ParaCrawl, CommonCrawl, News Commentary, and Rapid) as the potential training data for our NMT system, amounting to 65M lines of parallel sentence-aligned text. We then filtered these data based on several heuristics, with the two most important ones described next.

Since especially the crawled corpora are very noisy, they often contain segments in a wrong language, or even things like programming code and XML markup. We used the CLD2 library[2] for sentence-level language identification to keep only those sentence pairs in which the source sentence was labeled as English and target sentence as German with the confidence of at least 90%.

Another heuristic was based on sentence length: we only kept sentences with at least 3 and at most 80 words (after tokenization). We also removed sentence pairs in which source and target sentence lengths differ by a factor of 5 or more.

Overall the filtering yielded a corpus of 37.6M lines and 556M words (on the English side, counted untokenized), which we used in all of the experiments presented in this paper. It included 256K unique lines of TED talks with 4.4M words on the English side.

3.2. Preprocessing

We used two types of preprocessing. The first one was the standard Moses tokenization [12] for text translation and lowercasing on the English side. The German side was truecased using a frequency-based method. The second preprocessing was used only for English with the goal of converting text into speech transcript similar to the one produced by the ASR system. Starting from the Moses tokenization, we removed all punctuation marks and spliced back contractions (e. g. `do n't` → `don't`) to match the corresponding tokens in the ASR lexicon. We also converted numbers written with digits to their spoken form using a tool based on the *num2words*[3] python library.

The final step for both types of preprocessing was segmentation into sub-word units with byte pair encoding (BPE) [13], separately for each language. We used 20K merging operations. During testing, we used the option to revert BPE merge operations resulting in tokens that were observed less than 50 times in the segmented training data.

3.3. Punctuation Prediction

To translate a speech transcript with an NMT system trained with the first, standard preprocessing described above, we need to automatically enrich it with punctuation marks. To this end, we trained a RNN for punctuation restoration similar to the one presented in [14]. Only the words in a sentence are used to predict punctuation marks (period, comma, and question mark only). The acoustic features are not used.

For the setup with the ASR-like preprocessing of English, punctuation prediction is done implicitly during translation, since the target side of the training corpus contains punctuation marks. Thus, the output of the ASR system can be directly used (after BPE) as input to the NMT system.

4. ASR system

The ASR system is based on a hybrid LSTM/HMM acoustic model [15, 16], trained on a total of approx. 390 hours of transcribed speech from the TED-LIUM corpus (excluding the black-listed talks) and the IWSLT Speech-Translation TED corpus[4]. We used the pronunciation lexicon provided with the TED-LIUM corpus. The acoustic model takes 80-dim. MFCC features as input and estimates state posterior probabilities for 5000 tied triphone states. It consists of 4 bi-directional layers with 512 LSTM units for each direction. Frame-level alignment and state tying were obtained from a bootstrap model based on a Gaussian mixtures acoustic model. We trained the neural network for 100 epochs by minimizing the cross-entropy using the Adam update rule [17] with Nesterov momentum and reducing the learning rate following a variant of the Newbob scheme.

The language model for the single-pass HMM decoding is a simple 4-gram count model trained with Kneser-Ney smoothing on all allowed English text data (approx. 2.8B running words). The vocabulary consists of the same 152k words from the training lexicon and the out-of-vocabulary rate is 0.2% on `TED.dev2010` and 0.5% on `TED.tst2015`. The LM has a perplexity of 133 on

[2] https://github.com/CLD2Owners/cld2

[3] https://github.com/savoirfairelinux/num2words

[4] We realized that the provided audio-to-source-sentence alignments of the TED talks were often not correct. As this could significantly degrade the performance of the audio encoder for the direct speech translation approach described in Section 7, we had to automatically recompute these alignments by force-aligning each TED recording to its corresponding source sentences, and applied heuristics to overcome the problem of transcription gaps (speech segments without a translation in the parallel data).

TED.dev2010 and 122 on TED.tst2015.

Since TED talks are a relatively simple ASR task, we decided not to proceed with sequence training of the acoustic model or LM rescoring with LSTM models in order to have more uncertainty in the lattices. Acoustic training of the baseline model and the HMM decoding were performed with the RWTH ASR toolkit [18]. We trained BLSTM models with RETURNN [19], which integrates into RWTH ASR as an external acoustic model for decoding. Prior to constructing CNs from lattices [20], we decomposed the words into individual arcs according to the BPE scheme described in Section 3.2. The construction algorithm uses arcs from the first-best path as pivot elements to initialize arc clusters [21].

5. Neural Machine Translation System

We used the RETURNN toolkit [22] based on TensorFlow [23] for all NMT experiments. We trained two different architectures of NMT models: an attention-based RNN model similar to [1] with additive attention and a Transformer model [2] with multi-head attention.

In the RNN-based attention model, both the source and the target words are projected into a 620-dimensional embedding space. The models are equipped with either 4 or 6 layers of bidirectional encoder using LSTM cells with 1000 units. A unidirectional decoder with the same number of units was used in all cases. We applied a layer-wise pretraining scheme that lead to both better convergence and faster training speed during the initial pre-train epochs [22]. We also augmented our attention computations using fertility feedback similar to [24, 25].

In the Transformer model, both the self-attentive encoder and the decoder consist of 6 stacked layers. Every layer is composed of two sub-layers: a 8-head self-attention layer followed by a rectified linear unit (ReLU). We applied layer normalization [26] before each sub-layer, whereas dropout [27] and residual connection [28] were applied afterwards. Our model is very similar to "base" Transformer of the original paper [2], such that all projection layers and the multi-head attention layers consist of 512 nodes followed by a feedforward layer equipped with 2048 nodes.

We trained all models using the Adam optimizer [17] with a learning rate of 0.001 for the attention RNN-based model and 0.0003 for the Transformer model. We applied a learning rate scheduling similar to the Newbob scheme based on the perplexity on the validation set for a few consecutive evaluation checkpoints. We also employed label smoothing of 0.1 [29] for all trainings. The dropout rate ranged from 0.1 to 0.3.

6. Translation of ASR Confusion Networks

To encode confusion networks as input to the NMT system, we propose a novel, simple scheme. For a given speech utterance represented by acoustic vectors $\mathbf{o}$, we treat a confusion network C with J slots as the source sentence for the NMT. Instead of the one-hot encoding $x_j \in \{0,1\}^K$ (where K is the source vocabulary size) at position j within the sentence, the input is encoded as a K-dimensional vector $\bar{x}_j \in \mathbb{R}^K$ with $\bar{x}_j^k := p_j(w_k|\mathbf{o}), k = 1,\ldots,K$. Here, w_k is the k-th word in the vocabulary, and $p_j(w_k|\mathbf{o})$ is the posterior probability of the word w_k to appear at position j in C. In practice, $p_j(w_k)$ is different from 0 only for a small number of words.

In the end, following the notation of [1], we represent the input to the RNN encoder as the vector $E\bar{x}_j$ where $E \in \mathbb{R}^{N\times K}$ is the word embedding matrix and N is the dimension of the word embedding (e. g. 620). Thus, $E\bar{x}_j$ is a weighted combination of word embeddings for all the words in the CN slot j, with the highest weight given to the word with the highest posterior probability. In the corner case of only one arc per slot with the posterior probability of 1.0, we obtain a single word sequence. Thus, we can still use normal sentence pairs (e.g. from text-only parallel data) for training, along with the pairs of source CNs and their target language translations. The new input representation $E\bar{x}_j$ has the same dimensions as Ex_j and thus can be directly used to train a standard RNN NMT model or any other model that uses word embeddings. We kept the posterior weights fixed during back-propagation.

Word sequences of different length can be obtained from a CN because epsilon arcs can be inserted as alternatives in some of the slots. The best solution when training an NMT system on CNs would be to add an artificial source language token EPS that would not appear in the original text-only training data. However, because we decided against re-training the system on CN input from scratch, we mapped all epsilon arcs to the English word "eh", which denotes hesitation. It appears often enough in the English side of the parallel text-only corpus, but is almost always omitted in the human translation into German.

We also used CNs to simulate ASR word errors in text data. Following the work of [7], we used such noisy data in the training of the NMT system to make it more robust against similar real ASR word errors. To this end, for each word w in the first-best ASR output for the TED training corpus, we collected all the slot alternatives w'_n, $n = 1,\ldots,N_w$ to this word in the corresponding ASR CNs with their averaged posterior probabilities. After re-normalization of these probabilities, for each word w we obtained a confusion probability distribution p_w. Then, in a given sentence, we replaced every occurrence of the word w by one of its alternatives w'_n with probability $p_w(w'_n)$ from this distribution. One of the alternatives can also be an epsilon arc, we keep them (converted to "eh" as described above) to adapt the NMT system to epsilon arcs in CN input, inserting up to 2 consecutive arcs after each word with a probability e.

Finally, we used two control parameters to limit the noise level: probability to change a word p and probability to change anything at all in a given sentence s. Experimentally, we determined the settings $e = 0.02, p = 0.25, s = 0.6$ which resulted in WER of the noisy text as compared to its original text that was similar to the WER of the baseline ASR system.

Table 1: Results measured in BLEU [%] and TER [%] for the individual systems for the English→German speech translation task, translation of correct transcript vs. first-best ASR output of the TED.tst2015 set.

		correct transcript		ASR output (WER of 10.9%)	
#	System	BLEU	TER	BLEU	TER
0	RWTH IWSLT 2017 best non-ensemble system	30.5	52.3	–	–
1	text translation baseline (RNN)	32.4	50.5	25.2	60.2
2	text translation baseline (Transformer)	33.0	50.5	26.3	58.7
3	speech translation baseline (RNN)	31.4	51.9	26.6	60.0
4	speech translation baseline (Transformer)	30.7	52.8	25.8	59.5

7. Direct Speech Translation

In the direct approach to speech translation, a single neural network is used to predict the target translation given the audio features of the source sentence. The amount of training data for this setting, i.e. audio with the corresponding reference translations pairs from the TED corpus, is comparatively low. To exploit the much larger parallel text corpora, we choose a multi-task setup in which the network simultaneously learns to translate either from source audio or from source text. For this, we extend the RNN-based attention model described in Section 5 with an additional audio encoder that takes MFCC features as input. It consists of 5 bidirectional LSTM layers with 512 units each. Max-pooling layers with a pool size of 2 are inserted after each of the first 3 LSTM layers, reducing the sequence length by a factor of 8. Also, a separate attention mechanism is added for the audio encoder. The decoder switches between the context vector from the text encoder $c_{i,\text{text}}$ and the one from the audio encoder $c_{i,\text{audio}}$ depending on which input is given (using notation from [1]). The remaining part of the decoder is shared between both tasks.

To ensure that both types of input are seen frequently enough during training, we duplicate the speech translation corpus so that it grows to 30% the size of the parallel text corpus (66 duplicates). The concatenation of text and audio examples is then traversed in random order. For the direct system, the same optimization and regularization techniques are applied as in the NMT system described in Section 5.

8. Experimental Evaluation

We participated in the speech translation task of the IWSLT 2018 evaluation, the translation direction was English→German. All NMT models are trained on the filtered bilingual data as described in Section 3.1, no monolingual data was used. For the fine-tuning experiments, we used the TED talk part of the bilingual data together with the test sets TED.tst2010,2013,2014 (which were not used for tuning or evaluation). The TED talk part was also included in the baseline system. For the experiments with the confusion networks, we ran the ASR system to recognize the speech of the 170K TED training set and the test sets TED.tst2010,2013,2014 and used the resulting CNs with the corresponding German translations as (additional) training data.

We shuffled the training samples before each epoch and removed sentences longer than 75 and 100 sub-words in the attention RNN-based and the Transformer setup, respectively. We evaluate our models almost every 10K iterations and select the best checkpoint based on perplexity on the validation set. NMT 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 used TED.dev2010 consisting of 888 sentences as our validation set and evaluated our models on TED.tst2015 test set with 1080 segments. The systems were evaluated using case-sensitive BLEU [30] and normalized case-sensitive TER [31].

8.1. Baselines

First we trained a model with standard preprocessing for written text described in Section 3.2 and evaluated its quality on the correct transcript with punctuation marks of the TED.tst2015 set, as shown in Table 1. We observed a slightly better BLEU score for the Transformer architecture (line 2) as compared to the recurrent architecture (line 1). We also made a comparison to the best single system of RWTH Aachen University on this set from the IWSLT 2017 evaluation. With our baseline system we improved upon that result by 1.9% to 2.5% absolute.

We then trained a model with speech-like preprocessing of the English side of the parallel corpus as described in Section 3.2. This model not only translates English words to German, but also predicts punctuation marks. To match this condition, we applied the same preprocessing to the correct English transcript of TED.tst2015, removing the punctuation marks. The evaluation included punctuation marks. Because of the dual task (translation and punctuation prediction), the MT quality is lower, but only by 1% BLEU (line 3 of Table 1). Here, the recurrent architecture outperforms the transformer architecture (line 4) by a significant margin. Because of this, most of our subsequent experiments were based on the recurrent model.

8.2. Effects of ASR errors and Punctuation Prediction

When we translate the first-best ASR output for TED.tst2015, which has an ASR word error rate of

Table 2: Results measured in BLEU [%] and TER [%] for the individual systems for the English→German speech translation task, translation of ASR output.

#	System	TED.dev2010		TED.tst2015		tst2018	
		BLEU	TER	BLEU	TER	BLEU	TER
1	speech translation baseline (RNN)	26.5	55.2	26.6	60.0	–	–
2	+ fine-tuning on TED corpus	27.1	54.2	27.5	57.5	–	–
3	+ 2 additional encoder layers	27.3	54.7	27.6	57.5	–	–
4	+ fine-tuning on TED with noise	27.1	54.1	28.0	56.5	21.1	64.1
5	fine-tuning of 1) on TED CNs only	26.6	55.7	26.9	58.3	–	–
6	fine-tuning of 1) on TED correct + CNs	26.6	55.5	27.0	58.5	20.3	66.5
7	fine-tuning of 1) on TED correct+noise + CNs	26.2	55.9	27.0	57.9	20.2	66.7
8	speech translation baseline (Transformer)	26.1	55.6	25.8	59.5	–	–
9	+ fine-tuning on TED corpus	27.0	54.4	27.0	57.7	–	–
10	Ensemble of 2, 3, 4, 9	27.9	53.7	28.3	56.7	21.4	64.2
11	Ensemble of 2, 3, 4, 6	27.3	55.6	28.0	58.1	21.2	64.4
12	Ensemble of 2, 3, 4, 5, 6, 7	27.5	54.2	28.3	56.7	21.5	64.1

10.9%, we observe a significant degradation of MT quality. For example, the BLEU score goes down from 31.4% to 26.6%, cf. line 3 of Table 1. This means that the NMT system is sensitive to ASR errors. Otherwise, the differences between architectures are similar when compared on the ASR first-best output as opposed to correct transcript.

8.3. Confusion Network Translation

For the subsequent experiments, we start with the RNN speech translation baseline. Table 2 presents the results on the ASR output for the `TED.dev2010` validation set and `TED.tst2015` test set. For the lines where confusion networks are mentioned, they were used as input to the NMT system as described in Section 6. The CNs were pruned based on the threshold of 0.0001 for the posterior probability; a maximum of 20 arcs per slot with highest probability were kept. The average density of the final CNs on the training and validation sets was 1.8 and 2.2, respectively.

Fine-tuning on the TED corpus (using the correct transcript as the English side of the parallel corpus) improves the result on the test set by 0.9% BLEU absolute, as shown in line 2 of Table 2. We fine-tuned our models with a small learning rate of 0.00001 for the Transformer model and the models using CNs, which is additionally decayed by a factor ranging from 0.8 to 0.9 after each half an epoch. For the attention RNN-based models which do not use CNs as input, the learning rate was set to 0.0001 with decay rate of 0.9. We also tried fine-tuning using a model with 6 encoder layers instead of 4, but have not obtained any further improvements as compared to the fine-tuned model with 4 encoder layers.

Next, we duplicated the TED parallel corpus and introduced noise into the duplicate. The level of noise was selected to be similar to the ASR word error rate on the development set, and the noise itself was created as described in Section 6. Line 4 of Table 2 shows that after fine-tuning on both correct and noisy TED corpus, we obtain an improvement of 0.5% BLEU and 1.0% TER when translating the first-best ASR output for the `TED.tst2015` set, as compared to fine-tuning without the noise (line 2). Thus, to some extent the NMT system was able to learn how to cope with noise that is based on common ASR errors.

Because we could only run ASR on the 170K sentences from the TED corpus, for which the speech was well-aligned with reference translations, we decided to use confusion networks for fine-tuning of the NMT system only. To make it possible, we replaced the original English word embeddings of the model with the linear combination of the embeddings of CN slot alternatives, as described in Section 6. The fine-tuning was done on a random mix of correct TED talk transcripts and ASR CNs for these transcripts at the same time.

Lines 5-7 of Table 2 list the results of three fine-tuning experiments which include CNs in the source-side training data. We either continued training of the model on 170K CNs (and the reference translations of the corresponding transcripts), or on CNs plus correct transcripts of the same set, or on CNs plus correct transcripts and transcripts with noise that was inserted as in the experiment in line 4 of Table 2. In all three cases we used a lower learning rate, a smaller batch size, and continued fine-tuning for 5-6 epochs. Unfortunately, the BLEU/TER scores go down as compared to the best result in line 4 when translating first-best ASR output.

Detailed analysis of the NMT output from line 6 (best result on the validation set) showed that when translating confusion networks, the model is able to recover from some recognition errors. For example, the `TED.tst2015` utterance `you throw the ball but you're hit right as you throw` is translated by the system in line 4 of Table 2 as `Sie werfen den Ball, aber das ist Ihr Thron` because of the ASR error `as you throw` → `is your throne`. The system that translates the corresponding ASR confusion network, however, is able to produce a correct translation: `Sie werfen den Ball, aber Sie werden getroffen.` In another anecdotal example there is no error in the first-best ASR output, but the NMT system is

not able to disambiguate the meaning of the word picking. The English source is: see over there somebody is kind of picking their nose. The translation of the first-best ASR output is: Da drüben, da ist jemand in der Nase (Over there, someone is in the nose). When the corresponding CN is translated, the translation preserves the original meaning: Sehen Sie, da ist jemand, der sich in die Nase bohrt. We looked at the CN alternatives for the word picking. It had a posterior probability of 0.77, and the top competing hypotheses were taking (0.14), making (0.06), shaking (0.02), sticking (0.004). The embeddings of these words seem to have helped the NMT system to correctly infer the meaning of picking in the given context.

In many cases, however, multiple alternatives, especially to an empty slot, sometimes confused the system to a point where translation quality was adversely affected. In a final contrastive experiment, we used the system from line 7 in Table 2 to translate not the CNs, but the first-best ASR output for the same test set. The result – BLEU of 28.0% and TER of 56.5% on TED.tst2015 – was nearly identical to line 4 in the table and showed that the system was fine-tuned well to the TED domain, and the noise in the CNs made the NMT system more robust against the errors which the ASR system could not avoid to make. However, the system does not generalize well to unseen CNs and may make errors by encoding meaning from alternatives to correctly recognized words, even if their posterior probability is low. Nevertheless, we think that the approach is promising and can benefit from a better training strategy, more CNs in training, and a better, adaptive weighting of CN alternatives. We plan to implement these improvements in our future work.

8.4. Direct Speech Translation

The direct translation system trained on the 170K segments of the TED corpus where both the audio files and their translations are available and well aligned yields a BLEU score of 15.6% when translating the speech of the TED.tst2015 set. For comparison, a standard text-only attention RNN model trained on the same corpus using the correct English transcript reaches the BLEU/TER scores of 18.5% on the first-ASR output for the same set. Although the results are much worse than for the NMT systems in Table 2 trained on large amounts of data, we see that the direct system can still produce results only moderately worse than a system trained on the same data, but on English text instead of speech. When we try to improve the direct system by multi-task learning using all of the text parallel data as described in Section 7, we obtain a BLEU score of 17.1% after many days of training that has not converged, neither until the evaluation nor paper submission deadline. Thus, although in preliminary tests multi-task learning seems to work correctly and bring improvements, the approach requires a faster implementation and a better training/optimization strategy.

8.5. Final Results

The Transformer model, even when fine-tuned on the TED corpus, did not result in additional improvements over the attention based RNN model (lines 8 and 9 of Table 2). However, it contributed to the ensemble of several systems (line 10). The RNN model that uses CN input can potentially be ensembled with the Transformer NMT model (using either first-best or CN ASR output as input). However, we did not have time to implement the necessary changes for such model combination. Therefore, for our primary evaluation submission we combined only the RNN models which translate either first-best ASR input (models from lines 2, 3, 4 of Table 2) or confusion networks (models from lines 5-7 of Table 2). The BLEU of this ensemble system in line 12 is only marginally better than of the best single system from line 4.

For the 2018 evaluation data, we first used the acoustic sentence segmentation of the ASR system. Because it was too fine-grained and unreliable due to many pauses of the speakers and could lead to context loss for the NMT system, we ran the punctuation prediction algorithm described in Section 3.3 on the first-best ASR output, but used its results only to define new segment boundaries at time points when a period or question mark was predicted after a word. We then re-ran the recognition to generate first-best and CN ASR output using the segmentation obtained in this way. The performance on the 2018 evaluation data is reported in the last column of Table 2. The BLEU and TER scores were provided by the organizers for our primary and contrastive submissions. We observed similar tendencies here as on the TED.tst2015 set: unfortunately, using ASR confusion networks as input to NMT results in worse scores (e.g. by 0.8% absolute in BLEU) as compared to the best single system translating ASR first-best output. Our primary submission from line 12 obtains the best results also on the 2018 evaluation set, but the improvement due to ensembling of multiple systems is not significant.

9. Conclusion

AppTek participated in the speech translation task of the IWSLT 2018 evaluation, achieving the BLEU score of 21.5% on the 2018 English to German evaluation data with the primary submission. Our best setup used an ensemble of attention RNN MT models which translate either first-best ASR output or ASR confusion networks, generating target language text with punctuation marks. We proposed a novel scheme for encoding CNs in NMT and showed that the negative effect of some ASR errors can be reduced when CNs are translated, although further improvements in training strategy are necessary to obtain significant improvements in speech translation quality. Preliminary experiments with direct speech translation with a single sequence-to-sequence model showed promising improvements due to a novel multi-task learning scenario that allows for exploitation of text-only parallel MT training data.

10. References

[1] 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

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[3] H. Ney, "Speech translation: Coupling of recognition and translation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, USA, Mar. 1999, pp. 517–520.

[4] E. Matusov, B. Hoffmeister, and H. Ney, "ASR word lattice translation with exhaustive reordering is possible," in *Interspeech*, Brisbane, Australia, Sept. 2008, pp. 2342–2345.

[5] N. Bertoldi, R. Zens, and M. Federico, "Speech translation by confusion network decoding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honululu, HI, USA, Apr. 2007, pp. 1297–1300.

[6] M. Sperber, G. Neubig, J. Niehues, and A. Waibel, "Neural lattice-to-sequence models for uncertain inputs," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1380–1389.

[7] M. Sperber, J. Niehues, and A. Waibel, "Toward robust neural machine translation for noisy input sequences," in *International Workshop on Spoken Language Translation (IWSLT)*, 2017.

[8] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

[9] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Low-resource speech-to-text translation," *arXiv preprint arXiv:1803.09164*, 2018.

[10] V. Vandeghinste, C.-K. Leuven, J. Pelemans, L. Verwimp, and P. Wambacq, "A comparison of different punctuation prediction approaches in a translation context," in *21st Annual Conference of the European Association for Machine Translation*, p. 269.

[11] P. Lison and J. Tiedemann, "Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles," 2016.

[12] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, 2007. [Online]. Available: http://aclweb.org/anthology/P07-2045

[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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016. [Online]. Available: http://aclweb.org/anthology/P/P16/P16-1162.pdf

[14] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration." in *Interspeech*, 2016, pp. 3047–3051.

[15] H. Bourlard and C. J. Wellekens, "Links between Markov models and multilayer perceptrons," in *Advances in Neural Information Processing Systems I*, D. Touretzky, Ed. San Mateo, CA, USA: Morgan Kaufmann, 1989, pp. 502–510.

[16] 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

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," San Diego, CA, USA, May 2015.

[18] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney, "RASR/NN: The RWTH neural network toolkit for speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Florence, Italy, May 2014, pp. 3313–3317.

[19] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney, "RETURNN: the RWTH extensible training framework for universal recurrent neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, New Orleans, LA, USA, Mar. 2017, pp. 5345–5349.

[20] L. Mangu, E. Brill, and A. Stolcke, "Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks," *Computer, Speech and Language*, vol. 14, no. 4, pp. 373–400, Oct. 2000.

[21] D. Hakkani-Tür and G. Riccardi, "A general algorithm for word graph matrix decomposition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, Apr. 2003, pp. 596–599.

[22] A. Zeyer, T. Alkhouli, and H. Ney, "RETURNN as a generic flexible neural toolkit with application to translation and speech recognition," in *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, 2018, pp. 128–133. [Online]. Available: https://aclanthology.info/papers/P18-4022/p18-4022

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[24] 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

[25] P. Bahar, J. Rosendahl, N. Rossenbach, and H. Ney, "The RWTH Aachen machine translation systems for IWSLT 2017," in *14th International Workshop on Spoken Language Translation*, 2017, pp. 29–34.

[26] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016, version 1.

[27] 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.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[29] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, "Regularizing neural networks by penalizing confident output distributions," *CoRR*, vol. abs/1701.06548, 2017. [Online]. Available: http://arxiv.org/abs/1701.06548

[30] 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.

[31] 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.

The Sogou-TIIC Speech Translation System for IWSLT 2018

Yuguang Wang, Liangliang Shi*, Linyu Wei*, Weifeng Zhu*, Jinkun Chen*
Zhichao Wang*, Shixue Wen*, Wei Chen*, Yanfeng Wang*, Jia Jia†*

*Voice Interaction Technology Center, Sogou Inc., Beijing, China
{wangyuguang,shiliangliang}@sogou-inc.com
†Tiangong Institute for Intelligent Computing, Tsinghua University, Beijing, China
{jjia}@tsinghua.edu.cn

Abstract

This paper describes our speech translation system for the IWSLT 2018 Speech Translation of lectures and TED talks from English to German task. The pipeline approach is employed in our work, which mainly includes the Automatic Speech Recognition (ASR) system, a post-processing module, and the Neural Machine Translation (NMT) system. Our ASR system is an ensemble system of Deep-CNN, BLSTM, TDNN, N-gram Language model with lattice rescoring. We report average results on tst2013, tst2014, tst2015. Our best combination system has an average WER of 6.73. The machine translation system is based on Google's Transformer architecture. We achieved an improvement of 3.6 BLEU over baseline system by applying several techniques, such as cleaning parallel corpus, fine tuning of single model, ensemble models and re-scoring with additional features. Our final average result on speech translation is 31.02 BLEU.

1. Introduction

We have participated in the Speech Translation of lectures and TED talks from English to German task. The goal of this task is to translate fully un-segmented talks or lectures from English to German.

A pipeline approach is employed in our work. It consists of segmentation of audio data, ASR system, punctuation restoration and NMT system. A two pass decoding is used in the ASR system. In the first pass, we use several different neural network, such as Deep-CNN [1] [2], BLSTM and TDNN [3] to generate ensemble results. Then the decoding lattices of the ensemble system are sent to a second pass decoder for lattice rescoring. In order to bridge the gap between the output of ASR system and training data of NMT system, punctuation restoration, disfluency detection and inverse text normalization are necessary in our pipeline. Our NMT system is based on the Transformer architecture [4], which is based solely on attention mechanisms. Several techniques are adopted to improve our system, such as parallel corpus cleaning, fine tuning, model ensembling and re-scoring with additional features.

The rest of this paper is structured as follows. Section 2 describes the details of our ASR system, and Section 3 describes our NMT system. Our results in the speech translation task are presented in Section 4. We conclude this paper in Section 5.

2. Automatic speech recognition

2.1. Audio Segmentation

In this evaluation, the test set is provided without manual sentence segmentation, thus automatic segmentation of the final test set is essential. We utilize an approach to automatic segment audio data based on the signal energy. We set a threshold to split the audio between 8 and 15 seconds and then concatenate utterances that are shorter than 8 seconds to its neighboring utterances.

2.2. Audio Data Preparation and Feature Extraction

2.2.1. Data Cleaning

Our acoustic data comes from two sources. The first is the TED-LIUM [5], which contains 340 hours of well transcribed data. The second part comes from Speech-Translation TED corpus, which is about 270 hours of data with some bad segments, e.g. music or transcriptions not comparing the wav files. We follow the way in kaldi toolkit [6] to do the cleaning[1]. This aimed to cut the bad part off and only retrain the segments that can be compared with the transcripts. And we got about 220 hours in this part.

2.2.2. Dereverbration

For speech dereverbration, we calculate the RT60 [7] of the speech firstly. The speech whose RT60 is longer than 400ms is filtered with the Kalman filtering algorithm [8] to dereverberate the speech. Thus we get about 11,000 kalman filtered utterances and add them to the original data.

2.2.3. Speed Perturbation

Speed perturbation is done with 1.1 and 0.9 times for all the data above. Finally we obtain about total 1700 hours acoustic data to get robust performance in the end.

2.2.4. Feature Extraction

Our acoustic feature engineering is not complicated. The system is built using several different features including 39-dimensional MFCC for GMM, 40-dimensional static MFCC and 80-dimensional filter banks for neural networks. These features can be augmented with i-vectors to train speaker

[1]https://github.com/kaldi-asr/kaldi/blob/master/egs/ami/s5b/local/run_cleanup _segmentation.sh

adapted networks. The dimension of i-vectors is chosen to be 200 which is extracted from every 50 frames. I-vectors and features are combined in the input layer for TDNNs and BLSTMs. While for CNN, input layer only contains filter bank and i-vectors are combined with the fully-connected layer before the output. We also extracted fMLLR transformed feature on 340hr TED-LIUM data and found fMLLR features contribute no significant improvement compared to i-vector augmented system. So we only use i-vector to train our final speaker adapted systems.

2.3. Acoustic Modeling

We use DNN-HMM hybrid acoustic model for all our ASR experiments. All NN systems were trained using Lattice-free MMI (LF-MMI) [9] loss function with low frame rate (LFR) equals to 3 to predict context dependent phones (bi-phone). We mainly used three different types of neural net architecture, including deep convolutional neural network (DCNN), bidirectional LSTM (BLSTM) and time delayed deep neural network (TDNN). In GMM-HMM part, we use 13-dimension mel frequency cepstral coefficient (MFCC) with first and second derivatives with 500 hours data without speed perturbation. The dictionary provided in the TED-LIUM dataset is used for our GMM training. The final GMM has totally 150,193 Gaussian mixtures, correspond to 4056 states. This 500hr GMM-HMM was used to align all the 1500h data to generate state alignments for clustering bi-phone labels used in LF-MMI training. After clustering we finally get 3144 bi-phones which equal the output nodes number of all our neural networks.

2.3.1. Deep CNN

We were inspired by the VGG net [10] and the deep CNN architecture used in [1] [2] to design our CNN model. We train our DCNN model with 80 dimension filter bank feature without first and second derivatives. We use batch normalization (BN) and ReLU nonlinear activations following each convolution layer. We stack 31 layer of such conv-BN-ReLU block with residual connections around every two of them. Most of the two dimensional time-frequency convolution kernels are all set to 3 x 3 with stride 1x1. We set kernel size to 5 x 5 with stride 2 x 1 in the 6th, 12th, 24th, 30th convolution layer to reduce the frequency dimension from 80 to 5. Every time we reduce the frequency dimension we double our kernel number. So as we go deeper, the kernel number is set to 32, 64, 128, 256, and 384. We train such DCNN with all the 1500h data to obtain system *dcnn*.

2.3.2. BLSTM

Our BLSTM model consists of 5 layers that has two unidirectional LSTM with 1024 cells and 512 projections. 256 of the projections are recurrent units and the other 256 projections are non-recurrent ones. 40 dimensional static MFCC feature is extracted for BLSTM training. By concatenating the two previous and the two following frames of MFCC, we use 40 * 5 = 200 dimension feature to train two BLSTM with different random seeds using all of the 1500h data. We call the two BLSTM with *blstm1* and *blstm2*.

2.3.3. TDNN

For TDNN neural acoustic model, we use factored form of TDNN [3] to design our own network. The factorized TDNN (TDNN-F) is reported to beat common TDNN with deeper architecture [3], in order to identify some hyper parameter configuration, we first train TDNN-F models with only the TED-LIUM data and decode with a relatively small n-gram language model. We summarize the intermediate result in table 1. Here we only report average WER of tst2013, tst2014 and tst2014.

As can be seen from the table, it is beneficial to use i-vector or fMLLR transformed feature to train speaker adapted networks. Comparing the third row and the forth row, we find WER of fMLLR system is 15.09 which is worse than i-vector augmented system of 14.23. As we gradually increase the number of layers from 16 to 26, a steady performance improvement is obtained. TDNNs that is deeper than 26 may decrease in performance as the 31 layer net is worse than 26 layer net. The 200 dimensional i-vector augmented 26 layer TDNN reaches a WER of 13.93. Additional discriminative training (DT) also help, it helps to decrease WER from 14.03 to 13.62. When adding the cleaned 220 hours of data, we lower the WER from 14.03 to 13.35.

Table 1: TDNN results on TED-LIUM corpus

#layer	configuration	average
16	40MFCC	15.76
16	40MFCC + 100ivec	15.53
21	40MFCC + 100ivec	14.23
21	40MFCC + fMLLR	15.09
26	40MFCC + 100ivec	14.03
26	40MFCC + 200ivec	13.93
26	40MFCC + 100ivec + DT	13.62
31	40MFCC + 100ivec	14.3
26	40MFCC + 100ivec + 220h data	13.35

Conclude from Table 1, our final TDNN use a 26 layers TDNN architecture. We abandon fMLLR and use 200 dimensional i-vector augmented to MFCC to train speaker adapted net. Each hidden layer contains 1024 units and 160-dimension bottleneck. The input to TDNN is 5 frames of 40-dimension static MFCC. The other TDNN layer has an input context equals to 3 which has different time stride. We constructed 3 consecutive layers with time stride 1, 4 consecutive layers with time stride 2 and 15 consecutive layers with time stride 3. Each of these consecutive layers with the same time stride is followed by a fully connected layer. We train two of them with different random seeds with total data, and the third TDNN with 80% data. After TDNN training we got *tdnn1*, *tdnn2* and *tdnn3.*

2.4. Language Model

2.4.1. Data Preparation and the Vocabulary

For the data preparation, number normalization and lowercasing are adopted to formatting the all-corpora. Next, punctuations are removed and the paragraphs are split into sentences. We choose 152217 English words to build the vocabulary and replace all the out-of-vocabulary (OOV) words in the corpora with the symbol “<unk>”.

2.4.2. *N-gram Language Models*

The constrained all-corpora consists of various text resources, such as news, TED subtitles, film subtitles, Europarl dataset and some web crawled materials. The Table 2 shows the details of the cleaned sub-corpora and their interpolation coefficients in n-gram language modeling. We estimate a series of sub-corpora 5-gram language models using the SRILM toolkit [11] with the modified Kneser-Ney smoothing. And then, the development datasets are used for the perplexities and the interpolation weights tuning. By linearly interpolating the different sub-corpora 5-gram models, the final back-off language model is estimated and adopted to the speech recognition system. The perplexities of the development datasets are listed in the Table 3.

Table 2: English language modelling datasets and interpolation coefficients.

Text corpus	**# Words**	**Interpolation**
TED	5.747 M	0.131
OpenSubtitles	144.1 M	0.064
Para WIT	3.263 M	0.029
ParaCrawl + Common crawl	765.1 M	0.048
News discussions	4638 M	0.397
News articles	4004 M	0.331

Table 3: The perplexities (PPL) of the English dev corpora.

Dev set	**5-gram LM**
tst2013	112.31
tst2014	143.11
tst2015	121.80

2.4.3. *LSTM based Neural Language Model*

To improve the computation efficiency in the neural language model, the vocabulary needs to be downsized. We select the top 30000 frequent words from the cleaned corpora to construct a small vocabulary, and replace the out-of-vocabulary words in the cleaned corpora with the symbol "<oos>" according to the customized vocabulary.

The LSTM based language model are trained with TensorFlow. The model contains two stacked dropout wrapped LSTM layers [12] with the hidden size of 256. The word embedding size is 256 and the initial learning rate is 0.1. After the training, we apply the LSTM based language model in the lattice rescoring and n-best rescoring with the Kaldi toolkit [6]. The pruned lattice-rescoring algorithm in [13] helps to achieve lower word error rate (WER) in ASR. Both in the lattice rescoring and n-best rescoring stages, interpolating the 5-gram language model with the LSTM based language model can further improve the ASR accuracies.

2.5. System combination

In the first pass, we use 6 neural network systems described in section 2.3, e.g. *dcnn, blstm1, blstm2, tdnn1, tdnn2, tdnn3* and 5-gram language model described in section 2.4.2. We combine the system in the posterior level and generate the first pass ensemble results. We also select the best single network system *tdnn1* to perform discriminative training, but found no performance gain when combine with the above 6 systems. The decoding lattices of the ensemble system are sent to a second pass decoder for lattice rescoring.

3. Neural Machine Translation

In this section, some post-processing details of ASR output and the architecture of our neural machine translation system are described.

3.1. Punctuation Restoration

The automatic speech recognition system only generates a stream of words without any punctuation symbols. In our work, we model the punctuation using the sequence to sequence architecture. Our punctuation restoration model is based on the Transformer architecture, which is based on attention only. In our work, given a sequence of words as our inputs, we label each word based on the punctuation after the word. Specifically, we label each word with comma, period, question mark, exclamation mark and non-punctuation.

The training dataset contains 41.5M sentences in total. Sentences were encoded using byte-pair encoding [15] with source vocabulary of about 30k tokens. We evaluate the performance of our punctuation restoration model by precision, recall and F1 score. We present the results in table.

Table 4: The result of our Punctuation Restoration model

Dev set	Precision	Recall	F1 value
tst13	88.01%	82.18%	85.00%
tst14	88.62%	84.28%	86.40%
tst15	91.51%	86.26%	88.81%
average	89.38%	84.24%	86.73%

3.2. Disfluency Detection and Inverse Text Normalization

Since the automatic speech recognition outputs often contain various disfluencies. In this paper, a simple but efficient detection approach is employed to identify and repair these disfluencies. At first, we remove the filled pauses, such as "uh" and "um". Then we define a window to identify and remove the repetitions in the output of ASR system.

After disfluency detection, the inverse text normalization is necessary for machine translation, because the corpus of machine translation are in written form, but the output of the automatic speech recognition are generally in spoken form, especially in figure, data and the amount of money. As shown in Figure 1, the word stream generated by ASR system is transformed into the standard form after punctuation restoration, disfluency detection and inverse text normalization.

Figure 1: Post-processing for ASR output

The output of ASR system:
and the results from the **twenty twenty two *uh* point five million** sentences we selected **sixteen point eight** and which let us to throw like **twenty two percent** of the corpus

After our post-process:
and the results from the **22.5 million** sentences, we selected **16.8** and which let us to throw like **22%** of the corpus.

3.3. Data Preparation and Cleaning for NMT

The parallel text data consists of four parts: Speech-Translation TED corpus, TED corpus (Web Inventory of Transcribed and Translated Talks, WIT), WMT2018 and OpenSubtitles2018.

We tokenize both the English and Germany text data by the Moses tokenizer[2]. Then the English data is transformed to lower case. To simplify the post processing of translation, we do not transform the German data to lower case. Finally, we use BPE subword segmentation tool to process the English data and German data.

We have observed some noise data, which cause a lot of translation errors. In order to improve the quality of parallel text data, we have cleaned the data.

- The samples whose number of tokens are over 100 will be removed.
- For one sentence pair, if the length rate of source/target is less than 1/2 or large than 2, they will be removed.
- We use SRILM Toolkit [11] to train an English ngram language model and a German ngram language model respectively with the parallel text data. The two LMs are used to evaluate the perplexity (PPL) for the sentences. For one source sentence (English side) and target sentence (German), they are removed if they meet the following two conditions: (1) we use source LM to calculate PPL. The PPL of source sentence is larger than that of target sentence; (2) we use target LM to calculate PPL. The PPL of target sentence is larger than that of the source sentence.

3.4. NMT Architecture

Our model follows the Transformer architecture which is solely based on attention mechanisms [4]. In our setup, the encoder has six layers. Each layer is consist of two parts: multi-head self-attention network and position-wise fully connected feed-forward network. The two parts employ both residual connection and layer-normalization. In the decoder, we employ masking to ensure that the prediction for the current word only depends on the previous words.

The dimension of word embedding is set to 512. The hidden state size is set to 1024. The vocabulary sizes of English and German are set of 60,000.

The sentences which have the similar number of tokens are grouped together. During training, the batches of size is set by the number of tokens which is set to 8000. We use the Adam optimizer to train the model.

3.5. Fine-tune

A large part of the training data comes from WMT, whose domain is news. But the test sets come from oral domain. After the systems are trained, we continue to train the systems by 5000 steps with the WIT parallel text data.

3.6. Ensemble

It's common to avoid over-fitting by using ensemble of several systems. There are two methods we have adopted. For one system training, we always average all of parameters across the last 20 checkpoints. For several system trainings, we compute the output tokens' possibilities by averaging the systems' output possibilities.

In the final system, we choose six systems to apply the second ensemble.

3.7. Re-scoring with NMT Variants

In order to get better translation result, we test different NMT variant models in re-scoring n-best list.

Target right-to-left NMT Model: When the target words are decoded by the NMT system, the later words will depend on the previous words decisions in the beam search decoder. So the word decision at time step t is much harder than that of time step t-1[16]. In order to alleviate this imbalance problem, a variant NMT model, which decodes the target words from right-to-left (R2L), is trained. The R2L model is used to re-score the n-best list which produced by the main NMT model. The scores represents the conditional probabilities of the reversed translations given the source sentences.

Target-to-source NMT Model: Moreover, the translations may be inadequate: the translations may repeat or miss out some words [17]. In order to cope with the inadequateness, we also test the target-to-source (T2S) model, which is trained with the source and target swapped.

We first produce one n-best list with an ensemble of serval models. Then we do force decoding with target right-to-left, target-to-source NMT models. We treat each models scores as an individual feature. We use k-batched MIRA [18] to tune weights for all the features. In order to get more diverse n-best list, we also try to increase the size of beam from 10 to 100 for re-scoring.

4. Results

4.1. Results for ASR

Table 5 shows our systems built for the ASR submission. In the first pass, we use 6 neural network system described in section 2.3, e.g. *dcnn, blstm1, blstm2, tdnn1, tdnn2, tdnn3* and 5-gram language model described in section 2.4.2. We combine the system in the posterior level and generate the first pass ensemble results. The decoding lattices of the ensemble system are sent to a second pass decoder for lattice rescoring.

Table 5: The WER result of our ASR model

System	tst2013	tst2014	tst2015	average
dcnn	11.15	8.86	7.77	9.26
blstm1	8.65	7.84	8.02	8.17
blstm2	8.78	8.07	8.03	8.29
tdnn1	8.5	7.35	6.24	7.36
tdnn2	8.52	7.42	6.15	7.36
tdnn3	8.47	7.55	6.18	7.4
+ensemble	8.01	7.08	6.54	7.21
+rescoring	7.49	6.76	5.95	6.73

4.2. Results for NMT

Table 6 shows the machine translation results on validation sets. All the results are cased BLEU evaluate by multi-bleu.perl script in Moses[3]. Our data cleaning technique

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

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

improves the baseline by 0.74 BLEU. Due to the domain of training data is not very consistent with that of test data, we continue training the system with the WIT parallel text data. This fine-tune technique get an improvement of 1.43 BLEU. In order to get more diverse models and better ensemble results, we train 6 models independently with different random initializations. The ensemble result gives an improvement of 0.53 BLEU over best single system. By increasing the beam size from 10 to 100 during decoding, we achieve another improvement of 0.05 BLEU. We add six right-to-left and six target-to-source NMT models as re-scoring features. It improved the system by 0.85 BLEU. The test2013 set is used as development set to tune the weights of re-scoring features.

Table 6: The English→Germany NMT results on three development sets. Submitted system is the last system.

system	tst2013	tst2014	tst2015	average
baseline	34.73	29.09	33.02	32.28
+data cleaning	35.4	30.03	33.62	33.02
+fine-tune	37.13	31.28	34.93	34.45
+ensemble	37.79	31.56	35.58	34.98
+beam(10 → 100)	37.92	31.32	35.86	35.03
+rescore(6*R2L,6*T2S)	38.90	32.36	36.38	35.88

4.3. Results for Speech Translation

Table 7 shows the final speech translation results on three test set. In order to tune the ASR and NMT system individually. We first segment the full utterance, and then align the utterance into segments with the correct English text segments and German translations. The transcript of the best ASR system was then passed to disfluency detection, Punctuation Restoration and text normalization module. Finally, ASR outputs with punctuations were translated into German. The average result of three test set for our Speech Translation is 31.02 BLEU.

Table 7: The English→Germany speech translation results on three sets.

system	tst2013	tst2014	tst2015	average
final system	32.95	28.28	31.82	31.02

5. Conclusions

This paper describes our pipeline system for the IWSLT 2018 Speech Translation task from English to German. The whole pipeline are consist of the wav utterance segmentation module, the ASR system, the punctuation restoration and the NMT system.

As for the ASR system, we adopted an ensemble system of Deep-CNN, BLSTM, TDNN, n-gram Language model with lattice rescoring. According to our experiments, TDNN achieved the lowest WER among these three acoustic modeling network for this task. For our tdnn acoustic modeling, we found adding layers, i-vector, cleaned data are effective. We have achieved average WER of 6.73 over three test sets using the combination system. For the NMT system, we also use an ensemble of Transformer system with n-best rescoring. And we use various techniques in our system, such as data cleaning, fine-tune, ensemble of models and n-best rescoring. These techniques help our system achieve 3.6 BLEU better than baseline. We use the outputs of the best ASR system as input of our NMT system, and we achieved average BLEU score of 31.02 over three development sets.

How to use document-level information to improve the ASR and NMT system performance and build a robust NMT system will be our future work.

6. References

[1] Zhang Y, Pezeshki M, Brakel P, et al. Towards end-to-end speech recognition with deep convolutional neural networks[J]. arXiv preprint arXiv:1701.02720, 2017.

[2] Qian, Yanmin, and Philip C. Woodland. "Very deep convolutional neural networks for robust speech recognition." Spoken Language Technology Workshop (SLT), 2016 IEEE. IEEE, 2016.

[3] Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohamadi, M., & Khudanpur, S. (2018). Semi-orthogonal low-rank matrix factorization for deep neural networks. INTERSPEECH 2018.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. CoRR, 2017.

[5] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the TED-LIUM Corpus with Selected Data for Language Modeling and More TED Talks", in Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), May 2014.

[6] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Silovsky, J. (2011). The Kaldi speech recognition toolkit. In IEEE 2011 workshop on automatic speech recognition and understanding (No. EPFL-CONF-192584). IEEE Signal Processing Society.

[7] Nakatani, T., Yoshioka, T., Kinoshita, K., Miyoshi, M., & Juang, B. H. (2008, March). Blind speech dereverberation with multi-channel linear prediction based on short time Fourier transform representation. In Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on (pp. 85-88). IEEE.

[8] Schwartz, B., Gannot, S., & Habets, E. A. (2015). Online speech dereverberation using Kalman filter and EM algorithm. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 23(2), 394-406.

[9] Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., & Khudanpur, S. (2016, September). Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. In Interspeech (pp. 2751-2755).

[10] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[11] A. Stolcke, "SRILM-an extensible language modeling toolkit," in Proceedings of Interspeech, September 2002, pp. 901–904.

[12] Y. Gal, and G. Zoubin, "A theoretically grounded application of dropout in recurrent neural networks," in Advances in neural information processing systems, pp. 1019-1027. 2016.

[13] H. Xu, T. Chen, D. Gao, Y. Wang, K. Li, N. Goel, Y. Carmiel, D. Povey and S. Khudanpur, "A Pruned

RNNLM Lattice-Rescoring Algorithm for Automatic Speech Recognition," in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on. IEEE, 2017.

[14] 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.

[15] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In Proceedings of ACL 2016.

[16] Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on Target-bidirectional Neural Machine Translation. In NAACL HLT 16, San Diego, CA.

[17] Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2016a. Neural machine translation with reconstruction. arXiv URL: https://arxiv.org/abs/1611.01874.

[18] Colin Cherry and Gorge Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation, In NAACL, 2012.

Samsung and University of Edinburgh's System for the IWSLT 2018 Low Resource MT Task

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

[1]Samsung R&D Institute, Poland
[2]School of Informatics, University of Edinburgh
{m.chochowski,p.przybysz}@samsung.com
{pwillia4,bhaddow}@inf.ed.ac.uk, {rico.sennrich,a.birch}@ed.ac.uk

Abstract

This paper describes the joint submission to the IWSLT 2018 Low Resource MT task by Samsung R&D Institute, Poland, and the University of Edinburgh. We focused on supplementing the very limited in-domain Basque-English training data with out-of-domain data, with synthetic data, and with data for other language pairs. We also experimented with a variety of model architectures and features, which included the development of extensions to the Nematus toolkit. Our submission was ultimately produced by a system combination in which we reranked translations from our strongest individual system using multiple weaker systems.

1. Introduction

This paper describes the joint submission to the IWSLT 2018 Low Resource MT task by Samsung R&D Institute, Poland, and the University of Edinburgh. We built several multilingual systems using the Tensor2Tensor[1] and Nematus[2] toolkits, ultimately choosing to use a system combination in which we reranked translations from our strongest individual system using multiple weaker systems.

As there was so little in-domain Basque-English data available, we experimented with the use of out-of-domain data, with the addition of synthetic data via back-translation, and with the incorporation of data for other language pairs. To support multilingual translation, we followed the single-model approach of [1] and simply prepended each source sentence with a token specifying the target language.

We experimented with a variety of model architectures and features. This involved the development of several extensions to the Nematus toolkit, including support for multi-GPU training, label smoothing, and mixtures of softmaxes. We have contributed our code to the public Nematus repository.

[1]https://github.com/tensorflow/tensor2tensor:1.6.3
[2]https://github.com/EdinburghNLP/nematus

2. Training Data

In brief, we used all of the provided in-domain parallel training data along with parallel data from OpenSubtitles and the Open Data Euskadil Repository. We also produced synthetic data by back-translating from English into Basque. Table 1 lists the individual parallel corpora that made up our training data. Note that not all of our systems used all of the data. We will indicate differences when describing the individual systems.

2.1. In-Domain Data

We used all of the available in-domain data, which we filtered in order to remove the TED talks covered by the devset. The task organisers had already removed the devset talks from the Basque-English training corpus, but the talks were present in the training data for all other language pairs. Since we were evaluating on the devset during system development, we filtered the in-domain data to avoid being misled by artificially strong results. In preliminary multilingual Nematus systems that used only the in-domain data, this filtering had a significant impact, reducing the BLEU score from 16.25 to 9.96.

For the Basque-Spanish and Spanish-English pairs, we used the excised training data to create supplementary devsets. Since the Basque-English devset only contained 1,140 sentence pairs, these additional devsets gave us greater confidence when evaluating system changes.

More generally, we noticed that the in-domain corpora contained many of the same talks, in effect reducing the amount of available in-domain Basque data.

2.2. Out-of-Domain Data

We added out-of-domain data for the Basque-English, Basque-Spanish, Spanish-English, and French-English language pairs. For all four, we used OpenSubtitles2018 data from the OPUS corpus. In order to avoid making our training data too unbalanced, we undersampled from the large Spanish-English and French-English corpora. This was done arbitrarily: we simply used the first N-million sentence pairs

Pair	Corpus	Sentence Pairs	eu src	es src	en tgt	es tgt
eu-en	In-domain	5,623	5,623	-	5,623	-
	OpenSubtitles2018	805,780	805,780	-	805,780	-
	Synthetic in-domain	277,097	277,097	-	277,097	-
	Synthetic OpenSubtitles2018	4,000,000	4,000,000	-	4,000,000	-
	Synthetic TED2013	1,904,674	1,904,674	-	1,904,674	-
eu-es	In-domain	5,546	5,546	-	-	5,546
	OpenSubtitles2018	793,593	793,593	-	-	793,593
	Euskadil	926,941	926,941	-	-	926,941
es-en	In-domain	277,097	-	277,097	277,097	-
	OpenSubtitles2018	10,000,000	-	10,000,000	10,000,000	-
es-fr	In-domain	277,278	-	277,278	277,278	-
eu-fr	In-domain	5,815	-	5,815	5,815	-
fr-en	In-domain	287,137	-	-	287,137	-
	OpenSubtitles2018	10,000,000	-	10,000,000	10,000,000	-
Total		29,566,581	8,719,254	20,847,327	27,840,501	1,726,080

Table 1: Statistics for the parallel training corpora used in our submission (note that not all individual systems use all of the corpora). Since we are interested in Basque-to-English translation (primarily) as well as Basque-to-Spanish and Spanish-and-English, we break down the corpus sizes for those source and target languages.

occurring in the full corpora. For Basque-Spanish, we also used the parallel data from the Open Data Euskadil Repository.

At the outset, we assumed that translation from Basque into Spanish would be easier than into English due to the greater availability of in-domain data. We contemplated pivoting from Basque to English via Spanish and therefore when selecting data from OpenSubtitles, we made an effort to include sufficient data to support high-quality Basque-Spanish and Spanish-English translation. However, translation quality for Basque-Spanish and Basque-English (as measured by BLEU) proved to be very similar and we therefore focused on direct translation from Basque to English.

As with the in-domain data, we noticed that there is a high degree of content overlap between the multilingual OpenSubtitles corpora. For OpenSubtitles2018, between 70% and 90% of the Basque side is common for Basque-English, Basque-Spanish and Basque-French making the effective size of Basque data seen by the system relatively smaller.

2.3. Synthetic Data

Basque is a language isolate spoken by less than 1 million people and as such there are few readily available parallel resources. One of the simplest ways to get more parallel data is to generate it syntheticaly through back-translation. In [2] it was shown that even poor quality synthetic corpora can improve translation quality. We used all available training data to train an English-to-Basque back-translation system for synthetic data generation (see Section 3.2 for details of the back-translation system).

In addition to back-translating the English side of the in-domain Spanish-English corpus, we back-translated the English talks from the OPUS TED2013 corpus, after filtering out the dev and test set talks. We also selected 4M pseudo in-domain sentences from OpenSubtitles using the filtering approach proposed in [3].

3. Tensor2Tensor Systems

In preliminary experiments, we tried training Transformer models [4] using both Tensor2Tensor and Marian, eventually choosing the former as the BLEU was higher. In all experiments we used the hyperparameters for *transformer_base*, setting the hidden layer size to 512, filter size to 2048, warmup steps to 16,000 and number of heads to 8. While training the back-translation model we set the *layer_prepostprocess_dropout* parameter to 0.1, while in the base systems it was set to 0.2. Each training was run on 8 GPUs for up to 300,000 training steps, with a batch size of 100 sentences per GPU.

3.1. Preprocessing

We relied on the preprocessing implemented in Tensor2Tensor for tokenization and wordpiece segmentation. For each corpus configuration we defined a new T2T problem inheriting from default TranslateProblem. We set the subword vocabulary size to 32k for all training runs, either bilingual or multilingual. The only additional preprocessing we did was punctuation normalization using the Moses toolkit and prepending the `<2xx>` tag at the beginning of source sentences, where `xx` was the code for the target language.

System	en-eu	es-eu	en-es
EnFrEs2EuFrEs	13.26	14.89	41.92

Table 2: BLEU scores for the Tensor2Tensor systems on dev2018 (eu-en) and on eu-es and es-en versions of the dev set (extracted from the training data). Since this was a back-translation system, we inverted the devsets to evaluate translation in the opposite direction. This system was used for back-translation of monolingual English corpora

3.2. Back-Translation System

For back-translation, we used all of the in-domain data listed in Table 1, along with the OpenSubtitles corpora for Basque-English and Basque-Spanish, and the Euskadil corpus for Basque-Spanish. We also used 1M sentence pairs of Spanish-English, making 5.5M sentence pairs in total.

We trained both Nematus and Tensor2Tensor systems on the same dataset, obtaining results of 12.14 BLEU and 13.26 BLEU respectively on the inverted Basque-English devset (see Table 2). We chose the better-performing Tensor2Tensor system for synthetic corpus generation.

3.3. Base System

For Basque-to-English translation, we experimented with different language pair and corpus selections (Table 3). We started with a bilingual model trained only on in-domain data. Next we trained a multilingual model adding all directions for in-domain data and oversampling the Basque-English data by a factor of 20 (to better balance the larger in-domain Spanish-English corpus). This resulted in a significant improvement of almost 7 BLEU points. After adding out-of-domain and synthetic corpora we got another 5 BLEU points. Next we experimented with removing the French data from the multilingual setting as it had the least Basque sentences, giving little additional input for that language and adding complexity be adding another language into the model. We observed that removing the French parallel corpora gave significantly better results, improving Basque-English translation by 1 BLEU point on dev2018. For the final submission we used the EuEs2EnEs model for n-best list generation.

4. Nematus Systems

Nematus [5] implements a GRU-based attentional encoder-decoder. Originally based on the model in [6], the toolkit has been extended to support features such as deep architectures and input factors. Our system was based on the configuration used in University of Edinburgh's WMT17 submissions [7]. To this we added several further extensions, which we describe below.

System	eu-en	eu-es	es-en
bilingual in-domain only	11.72	-	-
EuFrEs2EnFrEs in-domain only	18.41		
+ out-of-domain	22.26	22.28	42.76
+ back-translation	23.45	17.81	43.03
EuEs2EnEs	25.09		

Table 3: BLEU scores for the Tensor2Tensor systems on dev2018 (eu-en) and on eu-es and es-en versions of the dev set (extracted from the training data). The last system EuEs2EnEs was used to produce the 20-best list for further rescoring.

4.1. Preprocessing

All of our Nematus systems used a common preprocessing pipeline, consisting of five steps: normalization, tokenization, corpus cleaning, truecasing, and BPE segmentation. [8] We used scripts from the Moses toolkit [9] to perform the first four steps and subword-nmt[3] to perform the last.

The Moses tokenizer includes language-specific rules, which we opted to use.[4] However, we trained a shared truecasing model for all languages. The corpus cleaning script removes empty sentences and sentence pairs with length ratios greater than 9:1.

We trained a single joint BPE model over the full multilingual corpus, using 40,000 merge operations. Character sequences were only merged if they were observed 50 times in the training data.

4.2. Base System

Our base Nematus system used all of the data in Table 1 except for the synthetic data (which was added later for the final systems) and the French-English OpenSubtitles data. For Spanish-English we used 1M sentence pairs of OpenSubtitles rather than 10M.

4.2.1. Network Configuration

We used a word embedding size of 512 and hidden layer size of 1024. Both the encoder and decoder used a deep transition architecture [10], with 4 layers in the encoder and 8 in the decoder. We used layer normalization [11].

We tied the weights of the target-side embedding and the transpose of the output weight matrix [12]. Since the source and target sides used the same vocabulary, we also tied the source-side and target-side embeddings.

4.2.2. Training

We used the Adam [13] optimization algorithm with a learning rate of 0.0001 and a batch size of 80 (except where

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

[4] Moses does not include Basque-specific tokenization rules, so it fell back to generic tokenization for that language

noted). Training was stopped when the validation cross-entropy failed to reach a new minimum for 10 consecutive save-points (saving every 10,000 updates). The save-point used in the final model was selected based on the BLEU score of the validation set.

To speed up training, we excluded sentences in which either the source or target sentence contained more than 50 tokens.

In preliminary experiments, we found that it was important to use dropout (giving improvements of around 2 BLEU). The dropout rate was set to 0.1 for source and target word tokens and to 0.2 for embedding and hidden layers.

4.3. Extensions

4.3.1. Multi-GPU Support

Training the base model was already pushing the 12GB memory limit of our GPUs, restricting our ability to add new features. Since we did not want to risk compromising model quality by reducing the network size, we opted to implement multi-GPU training in order to reduce the per-GPU batch size, while maintaining (or increasing) the effective total batch size.[5] We added support for synchronous training in which the batch is split between multiple GPUs (on the same server), each running a full replica of the model, and then the gradients of the sub-batches are averaged. Unlike asynchronous training, this method does not affect translation quality compared to single-GPU training (assuming the batch size is constant).

4.3.2. Source language factors

As already mentioned, our training data contains tags to indicate the target language. In preliminary experiments, we found that it was beneficial to also specify the source language, which we did through the use of token-level factors. Our intuition was that the factors would help to disambiguate subword units that occur in multiple languages, but serve language-specific roles.

Since Nematus already included support for factors [14], this was simply a case of annotating the training and dev/test data with language tags and adjusting the network's word embedding settings: of the 512 source embedding units we reserved 12 for the source language factor tag and the remaining 500 for the BPE token embedding.

A contrastive experiment showing BLEU scores on dev2018 with and without source language factors can be found in Table 4.

4.3.3. Label smoothing

We implemented label smoothing [15], a regularization technique which has been shown to be effective for self-attention-based translation models [4], and, more recently, for RNN-based models similar to ours [16]. Following prior work, we set the ϵ parameter to 0.1. See Table 4 for results of a contrastive experiments with and without label smoothing.

[5] An alternative would have been to use delayed updates on a single GPU – or of course to buy GPUs with more memory.

System	eu-en	eu-es	es-en
Base	19.99	20.45	39.74
Base + source language factors	20.12	20.86	40.16
Base + label smoothing	20.46	20.65	40.16
Base + mixture of softmaxes	20.02	21.02	40.14
Base + fine-tuning	20.75	1.86	39.94

Table 4: BLEU scores for Nematus systems on dev2018 (eu-en) and on eu-es and es-en versions of the dev set (extracted from the training data). These systems use all of the parallel training data except for the synthetic data.

4.3.4. Mixture of Softmaxes

Like all standard neural translation models, our base model uses a softmax function to output a probability distribution over the target vocabulary for each timestep. For language modelling, [17] show that performance can be improved by using a combination of multiple softmax components. We reimplemented their method within Nematus and experimented with using a mixture of three softmax components. See Table 4 for results of a contrastive experiments with and without a mixture of softmaxes.

4.3.5. Fine-tuning

Since our system was trained on data drawn from multiple domains and covering several language pairs, we anticipated that there would be a benefit to fine-tuning on in-domain Basque-English data. After selecting the best model (according to validation set BLEU), we resumed training using only the in-domain Basque-English data (5,623 sentence pairs). See Table 4 for results of a contrastive experiment with and without fine-tuning.

4.4. Final Systems

Our final Nematus systems used all of the training data from Table 1, with the exception of the synthetic TED2013 corpus (since training was started before the filtered corpus was produced) and the French-English OpenSubtitles corpus. We used a 1M sentence pair version of the Spanish-English OpenSubtitles corpus. As in the Tensor2Tensor system, we oversampled the in-domain Basque-English corpus by a factor of 20. We experimented with removing the French training data but, unlike the Tensor2Tensor system, this did not improve performance (possibly because we had used less French data to start with).

We used all of the extensions just described. We trained two such systems, one using two GPUs with a total batch size of 80 and one using three GPUs with a total batch size of 160. Finally, we fine-tuned these systems giving a to-

System	dev2018	tst2018
Nematus (batch size 80)	22.56	23.18
Nematus (batch size 80, fine-tuned)	23.22	23.65
Nematus (batch size 160)	22.94	23.56
Nematus (batch size 160, fine-tuned)	23.86	24.12
Tensor2Tensor	25.09	25.40
+ reranking (default length penalty)	25.40	25.97
+ tuned length penalty	25.60	26.21

Table 5: BLEU scores on the official dev and test sets. The first five rows show the results for the individual Nematus and Tensor2Tensor systems used in the final system combination. The bottom two rows show the results of reranking the 20-best list from the Tensor2Tensor system with the Nematus systems and then tuning the length normalization parameter. The system in the bottom row is our submitted system.

tal of four Nematus systems. Unlike our base system, fine-tuning using only the in-domain data did not improve translation quality, possibly due to the oversampling of this data in the training set. Instead, we used a fine-tuning corpus that combined the genuine in-domain data with the synthetic in-domain data (which was back-translated from the English side of the Spanish-English corpus). Results with and without fine-tuning are given in Table 5.

5. System Combination

Of the individual systems, we achieved the best performance on the devset using the Tensor2Tensor EuEs2EnEs system. We used that system to generate a 20-best list, which we then rescored using the four final Nematus systems. After rescoring, we renormalized the individual scores for sentence length, optimising the length penalty (i.e., the alpha value in [18]) on dev2018, setting it to 1.5 in our submission (in all previous systems, the length penalty was set to the default value of 1.0). Finally, we reranked the list according to the sum of the five renormalized scores and used the resulting 1-best translations in our submission.

Table 5 gives BLEU scores on the dev and test sets for the five component systems and the reranked system, both with and without length penalty tuning.

6. Conclusions

For this task, we focused on supplementing the very limited in-domain Basque-to-English training data with out-of-domain data, with synthetic data, and with data for other language pairs. Through data alone, we improved translation quality from 11.72 to 25.09 BLEU.

Although our Nematus systems underperformed the Tensor2Tensor systems, we were able to narrow the gap through extensions to the base model, including label smoothing and source language factors. When evaluated on tst2018, our best Nematus system was 1.3 BLEU behind our best Tensor2Tensor system.

Despite the Nematus systems being weaker, we were able to further improve performance by reranking a 20-best list from the Tensor2Tensor system using the four final Nematus systems. Tuning the length penalty also boosted performance slightly. Our submitted system scored 26.21 BLEU on tst2018, outperforming the individual Tensor2Tensor system by 0.81 BLEU.

7. References

[1] M. Johnson, M. Schuster, Q. V. 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.

[2] 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.

[3] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 355–362. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145474

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.

[5] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, V. A. 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*. Association for Computational Linguistics, 2017, pp. 65–68.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of ICLR*, 2015.

[7] 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: Association for Computational Linguistics, September 2017, pp. 389–399.

[8] 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.

[9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07. Morristown, NJ, USA: Association for Computational Linguistics, 2007, pp. 177–180.

[10] 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*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 99–107.

[11] L. J. Ba, R. Kiros, and G. E. Hinton, "Layer Normalization," *CoRR*, vol. abs/1607.06450, 2016.

[12] O. Press and L. Wolf, "Using the output embedding to improve language models," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, 2017, pp. 157–163.

[13] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *The International Conference on Learning Representations*, San Diego, California, USA, 2015.

[14] 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.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.

[16] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, M. Schuster, N. Shazeer, N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, Z. Chen, Y. Wu, and M. Hughes, "The best of both worlds: Combining recent advances in neural machine translation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2018, pp. 76–86.

[17] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank RNN language model," in *Proceedings of ICLR*, 2018.

[18] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," *CoRR*, vol. abs/1609.08144, 2016.

The AFRL IWSLT 2018 Systems: What Worked, What Didn't

Brian Ore, Eric Hansen, Katherine Young, Grant Erdmann and Jeremy Gwinnup

Air Force Research Laboratory

{brian.ore.1, eric.hansen.5, katherine.young.1.ctr, grant.erdmann, jeremy.gwinnup.1}@us.af.mil

Abstract

This report summarizes the Air Force Research Laboratory (AFRL) machine translation (MT) and automatic speech recognition (ASR) systems submitted to the spoken language translation (SLT) and low-resource MT tasks as part of the IWSLT18 evaluation campaign.

1. Introduction

As part of the evaluation campaign for the 2018 International Workshop on Spoken Language Translation (IWSLT18) [1], the AFRL Human Language Technology team applied and improved techniques from previous workshops [2] and Conference on Machine Translation efforts [3] to the Spoken Language Translation and Low-Resource Machine Translation tasks.

2. Spoken Language Translation

2.1. Automatic Speech Recognition

This section describes the ASR systems that were developed for the baseline condition of the Speech Translation task. We trained two different English systems and performed system combination to obtain the final hypothesis for translation. Section 2.1.1 describes that language models (LMs) that were used for decoding and rescoring. Section 2.1.2 discusses the Kaldi ASR system, and Section 2.1.3 describes the Hidden Markov Model ToolKit (HTK) Tensorflow system. Finally, Section 2.1.4 describes how we segmented the test data and performed system combination.

2.1.1. Language Models

LMs were estimated on the provided TED data and subsets of News Crawl 2007-2017 and News Discussions versions 1-3. The subset of each news corpus was selected using cross-entropy difference scoring [4] with TED as the in-domain text, and selection thresholds were chosen to use 1/8 of each corpus to train N-gram LMs, and 1/16 of each corpus to train a recurrent neural network (RNN) LM. Interpolated bigram, trigram, and 4-gram LMs were estimated using the SRILM Toolkit,[1] and a RNN maximum entropy LM was trained using the RNNLM Toolkit.[2] The RNN included 160 hidden units, 300 classes in the output layer, 4-gram features for the direct connections, and a hash size of 10^9. The LM vocabulary included 100,000 words that were chosen using the select-vocab tool from SRILM.

Table 1: Kaldi WER. Decoding was performed using a trigram LM trained on TED.

Acoustic Training Data	dev2010	tst2010	tst2013
Speech-Translation TED	19.8	19.6	30.5
TEDLIUM	16.9	14.8	22.3
Combined	16.6	15.1	23.6

2.1.2. Kaldi System

The acoustic training data available for this year's evaluation included the Speech-Translation TED corpus and the TEDLIUM corpus. Based on a preliminary analysis of the Speech-Translation TED corpus, we removed all segments longer than 15 seconds from this corpus. The devtest and off-limit talks were sequestered from TEDLIUM, and a third data set was created by searching the Speech-Translation TED and TEDLIUM corpora for non-overlapping time segments. Next, an initial set of ASR systems were trained on each data set using the Kaldi open source speech recognition toolkit [5]. All Kaldi models discussed in this paper are based on the chain time delay neural network (TDNN)-rectified linear unit (ReLU) setup using i-vectors.[3] Standard data augmentation methods were applied during the Mel frequency cepstral coefficient (MFCC) feature generation stage, such as speech and volume perturbation. Each system was decoded using the same trigram LM, which was estimated from the provided TED data using the SRILM toolkit. Table 1 shows the word error rate (WER) obtained on dev2010, tst2010, and tst2013.

Based on the results in Table 1, a Kaldi ASR system was trained on TEDLIUM using the interpolated bigram LM described in Section 2.1.1. This model was then used to decode all of the audio from the Speech-Translation TED corpus (including segments longer than 15 seconds), and the ASR derived transcripts were folded in with the TEDLIUM data, as in a semi-supervised training scenario, to build the final Kaldi ASR system. This data set is referred to as TEDLIUM+ASR

[1] http://www.speech.sri.com/projects/srilm

[2] http://www.fit.vutbr.cz/$\sim$imikolov/rnnlm

[3] http://github.com/kaldi-asr/kaldi/tree/master/egs/swbd/s5c/local/chain

Table 2: Kaldi WER. Decoding was performed using an interpolated bigram LM, and rescoring was applied using an interpolated 4-gram and RNN LM.

ASR System	dev2010	tst2010	tst2013
Kaldi TEDLIUM	14.0	11.9	17.7
Kaldi TEDLIUM+ASR	13.5	11.4	17.0

in the remainder of this paper.

The test data was decoded as follows. First, the recognition lattices from the Kaldi bigram system were rescored with the 4-gram LM. Next, 1000-best lists were extracted from each lattice and rescored with the RNN LM. The final LM scores were obtained by linearly interpolating the log probabilities from the 4-gram and RNN LM. Interpolation weights of 0.25 for the 4-gram and 0.75 for the RNN were chosen based on results from previous experiments. Table 2 shows the final WER obtained with each system. Based on these results, we used the TEDLIUM+ASR system in all remaining experiments.

2.1.3. HTK-Tensorflow System

A hybrid neural network hidden Markov model (HMM) speech recognition system was developed using Tensorflow [6] and a version of HTK[4] that we modified according to the method of [7]. First, a Gaussian mixture model (GMM)-HMM system was trained on TEDLIUM. Phonemes were modeled using word-position-dependent state-clustered across-word triphones, and the final HMM set included 6000 shared states with an average of 28 mixtures per state. The feature set consisted of 12 perceptual linear prediction (PLP) coefficients, plus the zeroth coefficient, with mean and variance normalization applied on a per talk basis. Delta, acceleration, and third differential coefficents were appended to form a 52 dimensional vector, and heteroscedastic linear discriminant analysis (HLDA) was used to reduce the feature dimension to 39. Speaker adaptive training (SAT) was applied using constrained maximum likelihood linear regression (CMLLR) transforms, and the models were discriminatively trained using the minimum phone error (MPE) criterion.

A residual network (ResNet) was trained on the TEDLIUM+ASR data set described in Section 2.1.2. This network is based on the 18-layer network described in [8], with the batchnorm and ReLU activations moved to utilize full pre-activation residual units described in [9], and an additional fully connected layer for i-vector input. Figure 1 shows the ResNet structure. A context window of 17 was applied to the feature input, which included 40 log filterbank outputs normalized to zero mean and unit variance on a per talk basis. The 100 dimensional i-vectors were extracted on a per-talk basis with an i-vector extractor that was trained on

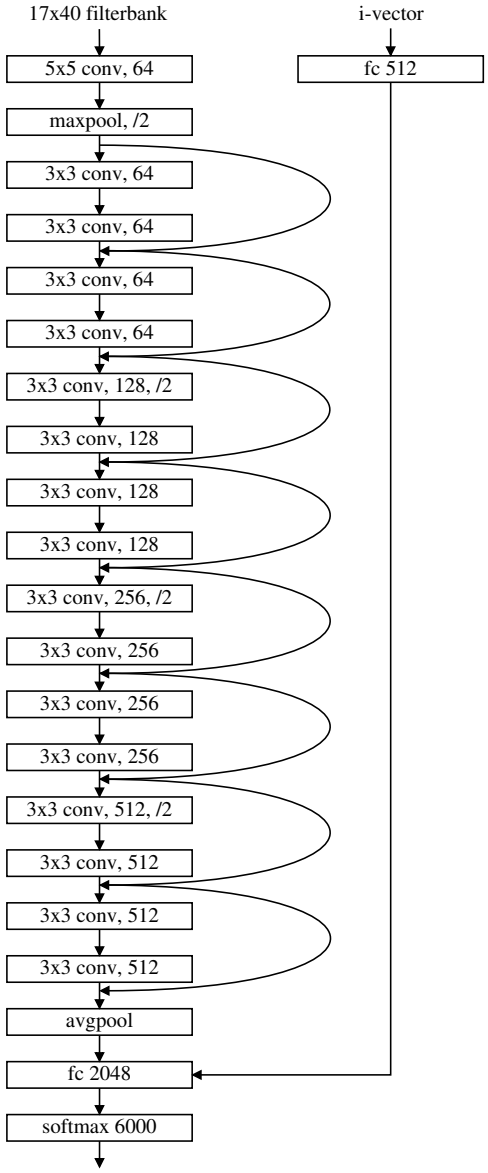


Figure 1: ResNet architecture based on [8, 9] with convolutional (conv), max pooling (maxpool), average pooling (avgpool), and fully connected (fc) layers. $H \times W$ is the filter size and $/2$ indicates that a stride of 2 was applied.

TEDLIUM using the same procedure as our IWSLT 2015 system [10]. Cross entropy training was performed using a mini-batch size of 512 and an initial learning rate of 0.0005 that was adjusted according to the QuickNet newbob algorithm.[5]

Recognition lattices were produced using HDecode with the interpolated trigram LM described in Section 2.1.1, and then rescored with the 4-gram and RNN LM using the same procedure as the Kaldi system. Next, confidence scores were estimated at the acoustic frame level by aligning the 20-best hypotheses for each utterance and counting the number of matching HMM states. An adapted ResNet was estimated

[4] `http://htk.eng.cam.ac.uk`

[5] `http://www.icsi.berkeley.edu/Speech/faq/nn-train.html`

Table 3: HTK-Tensorflow WER. Decoding was performed using an interpolated trigram LM, and rescoring was applied using an interpolated 4-gram and RNN LM.

ASR System	dev2010	tst2010	tst2013
ResNet	15.4	12.9	17.5
ResNet-Adapted	15.3	12.6	16.1

for each talk using frames that had a confidence score of 0.9 or higher and a single epoch of cross-entropy training with a learning rate of 0.0000625. Finally, the test set was decoded a second time and LM rescoring was reapplied. Table 3 shows the WER on dev2010, tst2010, and tst2013.

2.1.4. Test Segmentation and System Combination

The WER results reported in the previous sections were obtained by evaluating each ASR system on the automatically derived segments from the baseline implementation.[6] It was discovered that the segment boundaries did not always align with non-speech; therefore, we decided to use an alternative segmentation method.

A neural network based speech activity detector (SAD) was developed using Tensorflow. The SAD was trained on 40 hours from the TEDLIUM corpus using the automatically generated phoneme alignments from the HTK GMM-HMM system to define the speech/non-speech boundaries. The network included a context window of 41 frames on the input, a hidden layer of 1024 neurons with rectified linear activation functions, and 2 output units corresponding to speech and non-speech. The feature set consisted of 40 log filterbank outputs that were normalized to zero mean and unit variance. Automatic segmentation of the test data was performed by evaluating the SAD, applying a dynamic programming algorithm to choose the best sequence of states, and defining utterance boundaries at the midpoint of each non-speech segments longer than 0.5 seconds. Lastly, non-speech segments longer than 1.0 second were trimmed from each utterance.

The final hypothesis was selected by applying N-best recognizer output voting error reduction (ROVER) to the output from the Kaldi TEDLIUM+ASR and HTK-Tensorflow ResNet-Adapted system. Table 4 shows the WER obtain using the updated segmentation. Comparing Table 4 with the results in Table 2 and 3, we can see that the updated segmentation method provided a substantial improvement in WER.

2.2. ASR Postprocessing

We employed the provided SLT.KIT punctuator component to re-punctuate our ASR output before applying a truecaser model to induce the most common case for an English word before translating with the Marian section described in the next section.

[6] `http://github.com/isl-mt/SLT.KIT`

Table 4: WER using the updated test segmentation method. The final ASR hypothesis was obtained using N-best ROVER.

ASR System	dev2010	tst2010	tst2013
Kaldi TEDLIUM+ASR	9.5	7.7	12.8
ResNet-Adapted	11.2	8.6	11.2
N-best ROVER	9.5	6.9	9.8

Table 5: English-German cased BLEU scores for the SLT task. For comparison purposes, this table includes the scores obtained with the reference English source text.

English Transcripts	dev2010	tst2010	tst2013
Reference	27.10	27.40	28.83
ASR	18.48	17.20	18.40

2.3. Machine Translation

Lastly, a Marian [11] neural machine translation system was employed to translate the repunctuated text from English into German. This system was trained on the 41 million lines of preprocessed data provided by the WMT18 organizers for the news-translation shared task[12]. The data was truecased for uniformity, then a byte-pair encoding (BPE) [13] model was trained jointly on the source and target data with 90k merge operations.

As described in our WMT18 news-task efforts[3], we used the same parameters in training our Marian transformer [14] model:

- We used an encoding depth of 6 layers and a decoding depth of 6 layers.
- We used 8 transformer heads.
- We held the vocabulary size constant during training to 90k entries each for source and target.
- We held the word embedding dimensionality to 512 for all models.
- We used 1024 units in the hidden layer (where appropriate).
- We exclusively used the WMT `newstest2014` test set for validation.

2.4. Results

Results of scoring our repunctuated, translated ASR output and various references are shown in Table 5.

3. Low-Resource Machine Translation

For the low-resource translation task, we tried a variety of approaches with Marian [11], and Moses[15] toolkits. We

Table 6: Corpus size for each language pair in training corpus

Lang. Pair	Lines
Basque–English	5,623
French–English	288,366
Spanish–English	278,297
Total corpus	572,286

tried additional approaches with stemming and morphological processing, but systems trained with data processed in this manner were not ready in time for evaluation submission.

3.1. Common Training Corpus

For many of the experiments across different toolkits and systems, we constructed a common training corpus with uniform preprocessing in order to reduce variables when comparing different conditions.

Using the provided parallel Basque–English, French–English, and Spanish–English TED corpora [16], we construct a training corpus containing all three language pairs. Sizes of each portion of the training corpus are listed in Table 6. A joint BPE model was trained with 89,500 merge operations on the combination of all languages in the training data, then applied to the unified training corpus.

A similar corpus for use in backtranslation was constructed from the provided English–Basque, French–Basque, and Spanish–Basque corpora. Due to the small size of each of these component corpora, we also add the Basque–English portion of the OpenSubititles Corpus[7]. Sizes of each portion of this backtranslation training corpus are listed in Table 7. The BPE model from the 'forward' was used to segment the source and target data.

For some Marian experiments, we also constructed monolingual Basque and English corpora for use in constructing pretrained word embeddings. We use 50 million lines from the English monolingual CommonCrawl corpus selected for use in backtranslation from our WMT17 news-task efforts [17]. Additional monolingual Basque data was taken from the Commoncrawl website[8] and language-filtered using a modified C implementation[9] of the algorithm outlined in [18], yielding a Basque monolingual corpus of 38 million lines. We then apply BPE to each of these corpora with the same model as above and use word2vec [19] to generate 512-dimension word embeddings compatible with our settings in Marian.

3.2. Marian Systems

We spent the bulk of our efforts building systems with the Marian toolkit, experimenting with a variety of settings along two major categories: Sentence-weighting and backtranslated systems.

[7] http://www.opensubtitles.org
[8] http://www.commoncrawl.org
[9] https://github.com/saffsd/langid.c

Table 7: Corpus size for each language pair in backtranslation training corpus

Lang. Pair	Lines
English–Basque	5,623
French–Basque	6,948
Spanish–Basque	6,668
Basque–English OpenSubtitles	458,380
Total corpus	477,619

3.2.1. Sentence-Weighted training

We used the "forward" corpus outlined in Section 3.1 to train Marian systems with the same network parameters as outlined in the SLT translation system in Section 2.3. In Table 8, we note our baseline system (#1) scored 11.11 cased BLEU on dev2018. Next, we utilize the sentence-weighting feature of Marian that allows each sentence to be assigned a "weight" to determine how much of an effect each will have during training. A score of 1.0 is assigned to sentences from the Basque–English portion of the training corpus, French–English and Spanish–English sentences are assigned a score of 0.5. The system trained with these weights (#2) shows a +2.41 increase in BLEU.

Using the same data as system #2, we train a system that uses BEER [22] as the validation metric. While we have seen performance gains using this tactic in other work, here the resulting system(#3) performs -0.75 BLEU worse than the previous system.

Next, we consider averaging and ensembling of models. We take the 4-best model checkpoints from system #2 and average them into a single model, resulting in system #4's +0.87 BLEU gain over system #2.

Lastly, we decode with an ensemble of system #4 and a model averaged from the four best checkpoints of system #3, resulting in a BLEU score of 15.45. This system (#5) was then submitted as our entry to the low-resource MT task.

3.2.2. Backtranslated training corpus

As a contrast, we use the "backtranslation" corpus to train a shallow "s2s" Marian system that translates English, Spanish, and French into Basque. We then translate a 2 million line portion of the English monolingual corpus described in 3.1 into something resembling Basque and then use the combination of the two in conjunction with the small amount of provided Basque–English data to train two Marian "bi-deep" [20, 21] systems, both using BEER [22] as the training validation metric. These systems are listed as #6 (without pretrained word embeddings) and #7 (with pretrained word embeddings) in Table 8. We note that pretrained word embedding system scores -1.46 cased BLEU lower than the equivalent system without the pretrained embeddings, counter to

Table 8: Results for various MT systems decoding Basque–English dev2018 measured in cased BLEU. Our submission system is highlighted in bold text.

#	System	BLEU
1.	marian-eseufr-trans	11.11
2.	marian-eseufr-trans-weight	13.52
3.	marian-eseufr-trans-weight-beervalid	12.77
4.	marian-eseufr-trans-weight-avg4	14.39
5.	**marian-eseufr-trans-weight-avg4X2**	**15.45**
6.	marian-bt-bideep-beervalid	11.23
7.	marian-bt-bideep-preembed-beervalid	9.77
8.	moses-bt-bpe	14.06

our experience with our WMT18 systems.

3.3. Moses System

Using both the provided Basque–English data and the back-translated corpus outlined in Section 3.2.2 we train a Moses system in a similar vein to the one employed in our WMT18 submission: This system employed a hierarchical reordering model [23] and 5-gram operation sequence model [24]. The 5-gram English language model was trained with KenLM on the constrained monolingual corpus from our WMT15 [25] efforts. Our uniform BPE model used was applied to the parallel training data, but the language modelling corpus used the Russian–English joint BPE model from our WMT18 submission, possibly degrading performance due to this BPE mismatch. System weights were tuned with the Drem [26] optimizer using the "Expected Corpus BLEU" (ECB) metric.

This system, listed as #8 in Table 8 performs better than the two other Marian-based backtranslation systems (#6 and #7).

3.4. Results

Results of various systems described in the above sections are listed in Table 8. Our final submission system (#5) is highlighted in bold text.

4. Conclusions

Our experimentation this year show positive results in spoken language translation, especially our ASR component. However, for the low-resource MT task, we note that various approaches we have previously employed with great effect in high-resource conditions need further adaptation and refinement when scaling down to extremely low-resource conditions.

Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government. Cleared for public release on 4 Oct 2018. Originator Reference Number: RH-18-118975 Case Number: 88ABW-2018-4946.

5. References

[1] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, R. Cattoni, and M. Federico, "The IWSLT 2018 Evaluation Campaign," in *Proceedings of the 15th International Workshop on Spoken Language Translation (IWSLT 2018)*, Bruges, Belgium, October 2018.

[2] 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," in *Proc. of the 13th International Workshop on Spoken Language Translation (IWSLT'16)*, Seattle, Washington, December 2016.

[3] J. Gwinnup, T. Anderson, G. Erdmann, and K. Young, "The afrl wmt18 systems: Ensembling, continuation, combination," in *Proceedings of the Third Conference on Machine Translation*. Brussels, Beligum: Association for Computational Linguistics, October 2018.

[4] R. Moore and W. Lewis, "Intelligent selection of language model training data," in *Association Computational Linguists 2010 Conference Short Papers*, Uppsala, Sweden, July 2016.

[5] 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 *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, 2011.

[6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[7] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 30–42, January 2012.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, Nevada, June 2016.

[9] ——, "Identity mappings in deep residual networks," in *Proceedings of the 14th European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands, October 2016.

[10] 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 *Proc. of the 12th International Workshop on Spoken Language Translation (IWSLT'15)*, Da Nang, Vietnam, December 2015.

[11] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. F. Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, "Marian: Fast neural machine translation in c++," in *Proceedings of ACL 2018, System Demonstrations*. Association for Computational Linguistics, 2018, pp. 116–121. [Online]. Available: http://aclweb.org/anthology/P18-4020

[12] O. Bojar, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, and C. Monz, "Findings of the 2018 conference on machine translation (WMT18)," in *Proceedings of the Third Conference on Machine Translation*. Brussels, Belgium: Association for Computational Linguistics, October 2018.

[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)*. Association for Computational Linguistics, 2016, pp. 1715–1725.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[15] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proc. of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07, 2007, pp. 177–180.

[16] 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.

[17] J. Gwinnup, T. Anderson, G. Erdmann, K. Young, M. Kazi, E. Salesky, B. Thompson, and J. Taylor, "The afrl-mitll wmt17 systems: Old, new, borrowed, bleu," in *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 303–309. [Online]. Available: http://www.aclweb.org/anthology/W17-4728

[18] M. Lui and T. Baldwin, "Cross-domain feature selection for language identification," in *Proceedings of 5th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, 2011, pp. 553–561. [Online]. Available: http://www.aclweb.org/anthology/I11-1062

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *International Conference on Learning Representations (ICLR) Workshop*, 2013.

[20] 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*. Association for Computational Linguistics, 2017, pp. 99–107. [Online]. Available: http://aclweb.org/anthology/W17-4710

[21] 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: Association for Computational Linguistics, September 2017, pp. 389–399. [Online]. Available: http://www.aclweb.org/anthology/W17-4739

[22] M. Stanojević and K. Sima'an, "Fitting sentence level translation evaluation with many dense features," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, October 2014, pp. 202–206. [Online]. Available: http://www.aclweb.org/anthology/D14-1025

[23] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '08, 2008, pp. 848–856.

[24] N. Durrani, H. Schmid, and A. Fraser, "A joint sequence translation model with integrated reordering," in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics (ACL '11)*, Portland, Oregon, June 2011, pp. 1045–1054.

[25] J. Gwinnup, T. Anderson, G. Erdmann, K. Young, C. May, M. Kazi, E. Salesky, and B. Thompson, "The afrl-mitll wmt15 system: There's more than one way to decode it!" in *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, September 2015, pp. 112–119. [Online]. Available: http://aclweb.org/anthology/W15-3011

[26] G. Erdmann and J. Gwinnup, "Drem: The AFRL submission to the WMT15 tuning task," in *Proc. of the Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal, September 2015, pp. 422–427.

KIT's IWSLT 2018 SLT Translation System

Matthias Sperber, Ngoc Quan Pham, Thai Son Nguyen, Jan Niehues,
Markus Müller, Thanh-Le Ha, Sebastian Stüker, Alex Waibel

Institute for Antrophomaics and Robotics, Karlsruhe Institut of Technology

firstname.lastname@kit.edu

Abstract

This paper describes KIT's submission to the IWSLT 2018 Translation task. We describe a system participating in the baseline condition and a system participating in the end-to-end condition. The baseline system is a cascade of an ASR system, a system to segment the ASR output and a neural machine translation system. We investigate the combination of different ASR systems. For the segmentation and machine translation components, we focused on transformer-based architectures.

1. Introduction

The Karlsruhe Institute of Technology participated in the IWSLT 2018 Evaluation Campaign with systems for English→German Speech Translation task. We submitted system to both conditions: the baseline condition and the end-to-end condition.

The submission to the baseline condition is based on the cascaded approach described in [1]. In this evaluation campaign, we investigated the combination of different ASR systems. Furthermore, we investigated the use of transformer-based models.

This paper is structured as follows. In Section 2, we describe different speech recognition systems we employed in the campaign and how we combined them. Afterwards, we give a detailed description of the segmentation approach in Section 3 and the machine translation system in Section 4. Finally, in Section 5 we describe the end-to-end speech translation model. At the end of the paper we report the results and finish with a conclusion.

2. Speech Recognition

In this year's evaluation, we built three different types of automatic speech recognition systems. All the systems were trained using the data from the `TED-LIUM` Corpus version 2 [2].

2.1. Hybrid Model

Different from previous years, this year we built only one single HMM-based hybrid model for the speech recognition task. The hybrid acoustic modelling is constructed by stacking 5 LSTMs layers of 320 units, a projection layer of 200 units and a softmax layer to classify 8000 context-dependence phone (CD-Phone) states. As traditional approach, we used Viterbi forced alignment to provide CD-Phone state labels for the training data and the acoustic model was trained using only cross-entropy loss function.

For model training, we use SGD with an initial learning rate of 0.004 for 8 epochs and degrade it with a factor of 0.8 for other 8 epochs. We use a momentum term of 0.9 while dropout is set to 10%. Only 40 features of Mel-filterbank coefficients are fed into the LSTMs network every timestep, we did not employ any further speaker adaptation features.

After the model was successfully trained, we performed the traditional beam search decoding with the employment of the 4-gram language model. We used Janus Recognition Toolkit (JRTK) [3] as the decoding framework while the language model is built by Cantab research group [4] from WMT data.

2.2. CTC Model

Our CTC-based [5] ASR model is similar to the system described by [6]. The input to the model are 40-dimensional Mel-filterbank coefficients. We used every third speech feature of our input sequence and randomly chose the start offset during training, which has the advantage of a lower input sequence length. We trained the model to predict Byte-Pair Units, also referred to as Byte-Pair Encoding (BPE) [7].

The CTC-based model consists of four bidirectional LSTM layers with 400 units in each direction followed by a softmax layer. The size of the softmax layer depends on the number of BPE units we created. We used a dropout rate of 0.25 for all LSTM layers. We trained two models based on BPE units with 300 (small model) and 10,000 (big model) merges, respectively.

We used SGD with a learning rate of 0.0005 and a momentum term of 0.9 for training. The learning rate is halved whenever the validation token error rate does not decrease by more than 0.1%. We first trained the small model and initialized the parameters of the big model's BiLSTM layers using the smaller model's ones. We decoded the model by greedily selecting the most likely output at each time step.

2.3. Encoder-Decoder Model

Our attentional ASR model follows the listen-attend-spell [8] architecture and is similar to the system described by [9]. The model is implemented with XNMT. Compared to a conventional neural machine translation architecture, we replace the encoder with a 4-layer bidirectional pyramidal encoder with a total downsampling factor of 8. The layer size is set to 512, the target embedding size is 64, and the attention uses an MLP of size 128. Input to the model are Mel-filterbank features with 40 coefficients. For regularization, we apply variational dropout of rate 0.3 in all LSTMs, and word dropout of rate 0.1 on the target side [10]. We also fix the target embedding norm to 1 [11]. For training, we use Adam [12] with initial learning rate of 0.0003, which is decayed by factor 0.5 if no improved WER is observed. To further facilitate training, label smoothing [13] is applied. For the search, we use beam size 20 and length normalization with the exponent set to 1.5.

2.4. Rover

To combine the outputs of the different ASR systems, we used ROVER [14]. It operates on the final system output, the CTM-files. Multiple merging strategies exist to combine the outputs based on a majority vote. It is, e.g. possible to take confidences into account to further fine-tune the merging process. The key idea of ROVER is that different systems tend to produce different errors, but that no two systems produce the same error. The more different the systems those outputs are combined are, the better the result will be. Systems being very similar on the other hand will not benefit much from the system combination as they are likely to generate the same errors.

We here combined three different architectures: a traditional HMM-based ASR system, a RNN/CTC based one and an encoder-decoder based one. Hence, combining the outputs improved the WER due to the diversity of the system architectures.

3. Segmentation

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 conditions. To create segments and better match typical MT training conditions, we use a monolingual NMT system to add sentence boundaries, insert proper punctuation, and add case where appropriate before translating [15].

The idea of the monolingual machine translation system is to translate from lower-cased, unpunctuated text into text with case information and punctuation. Since we do not have any information about the sentence boundaries when inserting the punctuation and case information, we also remove them from the training data. Therefore, in the first step of the pre-processing, we randomly segment the source corpus of the training data into chunks of 20 to 30 words. Based on this randomly segmented corpus, we build the input and output data for the monolingual translation system.

For the input data, we remove all punctuation marks and lowercase all words. Since we will get lower-cased input, we cannot use the same byte-pair encoding [7] as for the machine translation system. Therefore, we train a separate byte-pair encoding on the lower-cased source data with a code size of 40k. To summarize, the source sequence consists of lower-cased BPE units without any punctuation.

For the target side, we do not want to change the words in the output sentence, but only add case and punctuation information. Therefore, we replace the sentence by features indicating case with punctuation attached. Every word is replaced by a letter *U* or *L*, whether it is upper-cased or lower-cased. Furthermore, punctuation marks following the word are directly attached to the letter.

At test time, we follow the sliding window technique described by [16]. Therefore, we created a test set with segments of length 10 starting with every word on the input data. This means, that except for the beginning and the end of the document, every word occurs ten times, at all positions within the segment. This of course dramatically increases the number of sentences in the test data. In the second step, we generate the target features by applying the monolingual translation system. In a post-processing step, we case the word as it most frequently occurs in the output. We insert punctuation marks, if there is at least one punctuation mark after the word in one of the 10 segments containing this word. If different punctuation marks are predicted, we take the most frequent one. Finally, if the punctuation mark is an end of sentence punctuation mark {*"."*,*"!"*,*"?"*}, we also start a new segment. The segmented test data with case and punctuation information is passed on to the machine translation system.

This year, we used a transformer-based NMT system to generated the punctuation marks.[1]. For the encoder and decoder we used 12 layers each using a hidden size of 512 and an inner size of the transformer model of 1024. We applied dropout and trained the models using adam. We first trained the system on the source side of the parallel data. We used the EPPS corpus, NC corpus and a filtered version of the paraCrawl corpus. In a second step, we fine-tuned the model on the TED corpus.

4. Machine Translation

Data preprocessing Our training data, while consisting the TED Talks provided by the evaluation campaign, also includes the following corpora: Europarl (1.8M sentences), News Commentary(280K), Rapid (1.2M), Common crawl (2.2M), the backtranslation data from University of Edinburgh (3.M) and the Paracrawl data (30M). The Paracrawl data is filtered by training a translation model to identify sentences with low likelihood. The final data size is around 36M

[1] https://github.com/isl-mt/NMTGMinor

sentences, with basic preprocessing steps being truecasing, tokenizing, and BPE splitting with BPE size of 40K. The development set is newstest2013 to newstest2016 from the WMT datasets for training the big models. In the adaptation phase, we use the TED talks of dev2010 to validate our models.

Modeling Our translation models are constructed with self-attention encoders and decoders, known as the Transformer networks, following the work of [17]. In this work, we extend the depth of the standard Tranformers, thanks for the residual design combined with layer normalization schemes allowing gradients to flow smoothly. Thanks to the huge amount of data as shown above, we were able to train models up to 32 blocks and still yield meaningful improvement.

Our hyper-parameters of the Transformer models (except depth) follow the *Base* configuration of the original work. The layer size for hidden layers is 512, while the the inner size of feed-forward network inside each block is 2048. The attention layers (including self-attention and attention between decoders and encoders) are multi-head attention layers with 8 heads. We also added label smoothing to regularize the cross-entropy loss. For the network depth, we trained models consisting of 4, 8, 6, 12, 16 and 32 blocks (for both encoder and decoder). Not only are deep models very demanding in terms of computation, they also consume a considerate amount of memory. In order to make training feasible, we used the checkpointing technique [18] by employ re-computation of the network activations during the backward pass to reduce the memory cost for the models.

Training procedure We group mini-batches to fill up our GPU's memory depending on the network size. 12-layer models can fit the memory with batch size containing 2048 words, while deeper models requires batch size reduction to avoid out-of-memory. For updating the networks' parameters, we accumulate gradients up to 25000 target words before doing an update. The learning rate is scheduled as in [17] but we doubled the initial learning rate and extend the warm up duration to 8000 steps. All models including the $32-layer$ config train with 100000 updates. Each model, except the 32-layer one has an additional variation with dropout (added to the residual connection and the inner feed-forward hidden layers).

Domain adaptation After training on all datasets, we further fine tune each model on the TED Talks specifically. Such technique is known to improve the model's performance greatly on the specifically adapted domain [19].

Noise adaptation Since the ASR output is fundamentally different than the collected natural data, we apply a noise model [20] on the TED training data which randomly replace words by sampling. The model is further fine-tuned on the noisy data in the same fashion as domain adaptation.

Final models The output is generated from the ensemble of five models: 12-layer, 12-layer with dropout, 16-layer, 16-layer with dropout and 32-layer.

5. End-to-End Models

We extend the attentional ASR described under Section 2.3 to perform translation by replacing the source-language target tokens by tokens from the target language. We use a refined encoder that performs downsampling to make memory requirements manageable and adds depth for improved accuracy, following one of the variants described by [21]: we stack several blocks consisting of a bidirectional LSTM, a network-in-network (NiN) projection, and batch normalization. After the last block, we add a final bidirectional LSTM layer. NiN denotes a simple linear projection applied at every time step, performing downsampling by concatenating pairs of adjacent projection inputs.

For better results and to be able to include the TEDLIUM corpus in the training, we devise a multi-task training strategy that trains auxiliary models on related tasks while sharing a subset of the parameters with the main ST model. Precisely, besides the main ST task we include an ASR task that shares encoder and attention, an MT task that shares attention and decoder, and an transcript auto-encoder task that shares only the attention. The ASR task is trained on the TEDLIUM corpus, whereas the other tasks are trained the respective subset of the 3-way TED corpus provided for the IWSLT 2018 evaluation. It should be noted that we did not put any efforts into cleaning this data, despite it being relatively noisy.

6. Experiments

We evaluated the models presented in the last section on the provided test sets. Note that for all experiments we used the audio segmentation tool [22] provided in the IWSLT docker container.

6.1. Cascaded

Our experiments involve different configurations regarding three main components in the cascade. For the ASR component, we present three different setups: the ROVER combination of two CTC and one Encoder-Decoder systems (dubbed as ROVER-1) and finally the ROVER combination of CTC,Encoder Decoder and Hybrid systems (dubbed as ROVER-2).

For the text segmenter, we showed the models trained on two data sizes (small and large), together with the larger models being adapted with domains and adapted with noise. Similarly, we showed the Translation models with additional adaptation towards domain and noisy inputs.

ASR	WER
ROVER-1	21.2%
HYBRID	17.6%
ROVER-2	16.7%

Table 1: Word-Error Rate of different ASR configurations on the tst2014 English set.

ASR	SEG	MT	BLEU
ROVER-1	Small	Transformer	13.77
ROVER-1	Small	D. Adapted	16.26
ROVER-1	Large	D. Adapted	18.2
ROVER-1	D.Adapted	D. Adapted	19.15
ROVER-1	N.Adapted	N.Adapted	19.32
HYBRID	D.Adapted	N.Adapted	21.33
HYBRID	N.Adapted	N.Adapted	21.41
ROVER-2	D.Adapted	N.Adapted	**22.61**
ROVER-2	N.Adapted	N.Adapted	21.35

Table 2: SLT English→German results. We report the BLEU scores (after re-segmentation) on the tst2014 test data. Note: Noise-adapted models (N.Adapted) were already adapted to the TED domain (D.Adapted) previously.

Regarding the whole cascade performance, as can be seen from Table 2, the score dramatically increased with the help of domain adaptation (2.5 BLEU points improved from adapting the translation model, and an additional 3.2 points from having a stronger adapted segmentation model (Large into D.Adaptation). Additional noise-adaptation on the segmentation and translation models improves the result by 0.2. We can also see that the HYBRID ASR model is much better than the ROVER-1 configuration, thanks to 17% improvement in word error rate on the English speech input. As a result, the whole cascade is improved by significant 3 BLEU points. The best ASR configuration - ROVER-2 - finalized the best result at 22.61. Notably, we have to use the segmentation model configuration without noisy-adaptation to achieve this, the counterpart fell short by 1 BLEU point.

6.2. End-to-End

We conduct preliminary experiments on the well-established Fisher Spanish-English Speech Translation Corpus [23] to confirm the model's accuracy. We obtain 35.3 BLEU points on Fisher/Test, 2.8 points better than a cascaded model using a similar architecture and the same training data. We then train the end-to-end model described in Section 5 on TEDLIUM2 for the ASR task, and on IWSLT2018's provided end-to-end TED data for the remaining tasks. We test the resulting model on the tst2013 dataset and obtain 10.3 BLEU points without casing, and 9.3 BLEU points case-sensitive scoring. This is still much worse than the cascaded models described above, despite the promising preliminary results. As potential reasons we identify the rather noisy provided training data, mismatch between the manual segmentation at training time and the automatic segmentation at test time, and the lack of additional data (beyond TED) included in the training.

7. Conclusions

In this evaluated we build a cascaded speech translation model as well as an end-to-end model for the English to German Speech Translation task.

For the cascaded approach, we see that the combination of several ASR systems reaches the best performance. Furthermore, we get the best single performance by using a hybrid model.

For the end-to-end model, we cannot achieve the same performance as the cascaded approach. One challenge is the integration of the significantly larger data available for the cascaded models.

8. Acknowledgements

9. References

[1] T. Zenkel, M. Sperber, J. Niehues, M. Müller, N.-Q. Pham, S. Stüker, and A. Waibel, "Open Source Toolkit for Speech to Text Translation," *The Prague Bulletin of Mathematical Linguistics*, vol. 111, pp. 125–135, October 2018. [Online]. Available: https://ufal.mff.cuni.cz/pbml/111/art-zenkel-et-al.pdf

[2] A. Rousseau, P. Deléglise, and Y. Esteve, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks," in *LREC*, 2014.

[3] 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.

[4] W. Williams, N. Prasad, D. Mrva, T. Ash, and T. Robinson, "Scaling recurrent neural network language models," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5391–5395.

[5] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification," in *Proceedings of the 23rd international conference on Machine learning - ICML 2006*. ACM Press, 2006.

[6] T. Zenkel, R. Sanabria, F. Metze, and A. Waibel, "Subword and crossword units for ctc acoustic models," *Interspeech*, 2018.

[7] 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)*. Association for Computational Linguistics, 2016.

[8] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, mar 2016.

[9] G. Neubig, M. Sperber, X. Wang, M. Felix, A. Matthews, S. Padmanabhan, Y. Qi, D. S. Sachan, P. Arthur, P. Godard, *et al.*, "Xnmt: The extensible neural machine translation toolkit," Boston, USA, 2018.

[10] 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.

[11] T. Nguyen and D. Chiang, "Improving lexical choice in neural machine translation," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018.

[12] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, Banff, Canada, 2015.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.

[14] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*. IEEE, 1997.

[15] E. Cho, J. Niehues, and A. Waibel, "NMT-based segmentation and punctuation insertion for real-time spoken language translation," in *Interspeech 2017*. ISCA, aug 2017.

[16] ——, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system," in *Proceedings of the Ninth International Workshop on Spoken Language Translation (IWSLT)*, 2012. [Online]. Available: https://isl.anthropomatik.kit.edu/pdf/Cho2012.pdf

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[18] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," *arXiv preprint arXiv:1604.06174*, 2016.

[19] 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 *Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT), Seattle, WA*, 2016.

[20] M. Sperber, J. Niehues, and A. Waibel, "Toward Robust Neural Machine Translation for Noisy Input Sequences," in *International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, 2017.

[21] Y. Zhang, W. Chan, and N. Jaitly, "Very Deep Convolutional Networks for End-to-End Speech Recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[22] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization," in *Interspeech*, 2013.

[23] 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. [Online]. Available: http://cs.jhu.edu/ gkumar/papers/post2013improved.pdf

Alibaba Speech Translation Systems for IWSLT 2018

Nguyen Bach, Hongjie Chen*, Kai Fan*, Cheung-Chi Leung*,*
Bo Li, Chongjia Ni*, Rong Tong*, Pei Zhang*,*
Boxing Chen, Bin Ma, Fei Huang

Machine Intelligence Technology Lab, Alibaba Group
firstname.lastname@alibaba-inc.org

Abstract

This work describes the En→De Alibaba speech translation system developed for the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2018. In order to improve ASR performance, multiple ASR models including conventional and end-to-end models are built, then we apply model fusion in the final step. ASR pre- and post- processing techniques such as speech segmentation, punctuation insertion, and sentence splitting are found to be very useful for MT. We also employed most techniques that have proven effective during the WMT 2018 evaluation, such as BPE, back translation, data selection, model ensembling and reranking. These ASR and MT techniques, combined, improve the speech translation quality significantly.

1. Introduction

In this paper we describe the Alibaba speech translation system that was built as part of the Speech Translation Task in IWSLT 2018. The task involved translating English audio to German text in which English audio are from lectures and TED talks. Our system employs a pipeline approach that includes an automatic speech recognition system (ASR) and a machine translation (MT) system.

The paper is organized as follows: Section 2 presents the ASR system used, along with a description of conventional, end-to-end, and fusion systems. Section 3 focuses on the MT system in which we describe preprocessing, data augmentation, noisy input translation, ensembling, and reranking components in detail. We present our concluding remarks in Section 4.

2. Automatic Speech Recognition

In a pipeline-based speech translation system, ASR is the most front-end module. In order to get reliable translation of quality, it is critical to obtain ASR transcription as accurate as possible. To start, we build several conventional pipeline-based ASR systems using deep neural network/hidden Markov model (DNN/HMM) framework. In the DNN-HMM framework, we employ several DNNs with different structures including fully-connected DNN (FDNN), time-delay deep neural network (TDNN) [1, 2], and latency-controlled bidirectional long short-term memory (BLSTM) [3].

Our final goal is to build end-to-end speech translation system, that is, we need to simplify the model-building process of conventional pipeline-based ASR system by constructing complicated modules with a single DNN architecture or in a data-driven learning method. Thus we also employ an end-to-end ASR system which is based on a hybrid connectionist temporal classification (CTC) [4, 5] and attention based encoder-decoder [6] architecture. Finally, we combine the output of different ASR systems to boost the final ASR performance.

During the development of our system, all acoustic models are trained on TED dataset together with the training datasets provided by the organizer. We noticed that the organizer's segmentation of the talks/lectures is not quite accurate, e.g. some sentences are not properly split. Therefore, we employ our own model-based voice activity detection (VAD) module to split the talks/lectures into utterances before ASR decoding. For the model-based VAD, the recurrent neural network (RNN) model is used to train and classify each frame into non-speech or speech. The RNN based VAD model was trained by using TED and other speech corpus, and by using Alibaba's VAD segmentation, it can get better performance when comparing with Organizer's segmentation. Table 2 listed the comparsion.

Table 1: Configuration of DNNs in DNN-HMM acoustic models.

System	Input feature	#Dim	Network context
FDNN	FBank+Pitch	80	{-5, 5}
TDNN	MFCC+ivector	40+100	{-13, 9}
BLSTM	FBank+Pitch	80	{-8, 8}

#Dim: number of feature dimension
{-L, R}: L frames in the left context and R frames in the right context

* Equal contribution

Table 2: WER (%) of different ASR systems on IWSLT 2013 and 2015 dataset using organizer's and Alibaba's segmentation.

System	Organizer's segmentation			Alibaba's segmentation		
	tst2013	tst2015	Avg.	tst2013	tst2015	Avg.
CTC (baseline)	26.1	37.6	31.9	-	-	-
1. FDNN/HMM	22.9	31.6	27.3	19.4	28	23.7
2. TDNN/HMM	13.6	25.5	19.6	11	23.1	17.1
3. BLSTM/HMM	13.6	26.3	20.0	10.6	22.3	16.5
4. CTC/Attention	26	33.1	29.6	23	29.8	26.4
1+2+3+4	-	-	-	10.3	22.3	16.3
2+3+4	-	-	-	**8.6**	**21.7**	**15.2**

2.1. Conventional ASR system

In the conventional DNN-HMM framework, DNNs are designated to predict the alignments derived from a GMM-HMM based acoustic model, given different input features. As shown in Table 1, FDNN and BLTSM take 80-dimensional filter bank feature vectors (FBank) and pitch feature as input feature while TDNN take 40-dimensional Mel-frequency cepstral coefficients (MFCCs) appended with 100-dimensional iVector as input feature. These DNNs take different lengths of context. FDNN takes one frame together with 5 frames from its left and right context as the input window, i.e. context span is {-5, 5}. TDNN follows the setting in [2] which takes {-13, 9} as the context span. BLSTM takes context span {-8, 8}.

We train a 4-gram LM using all the allowed text. The 4-gram LM obtained by linearly interpolating a number of LMs that are trained using all the TED transcripts provided by the organizer, all the text in the WMT18 CommonCrawl corpus, some sentences selected from the WMT18 news, WMT18 news discussion and OpenSubtitles2018 corpora. We use cross-entropy based data selection to select sentences from the corpora that are close to the TED transcripts. The LM interpolation weights are optimized on all development data.

2.2. End-to-end ASR system

We employ a hybrid CTC/Attention architecture [6] in our end-to-end ASR system. This system consists of a BLSTM-based encoder, a CTC output layer and an attention decoder. The CTC output layer and the attention decoder takes the output of encoder and predicts the corresponding letters. That is, the end-to-end system is charactor level system. Please refer to [6] for more system details. And in Table 2, there are large differences between CTC/attention based ASR system and TDNN or BLSTM based ASR system. The reasons lie in that (1) the scale of speech training data. The CTC/attention based ASR need more large scale of traning data to get better performance comparing with TDNN or BLSTM based approach; and (2) the weak language model for CTC/attention based ASR system.

2.3. Fusion of ASR output

With all the aforementioned ASR systems, we have a set of ASR output of each test set. The WER of the ASR systems are summarized in Table. 2. We utilize ROVER [7] to fuse the output from different ASR systems. By enumerating all combinations of each ASR system, we select the output from the combination of the TDNN/HMM, BLSTM/HMM and hybrid CTC/attention based end-to-end systems, since this combined output gives the lowest WER. There are so big differences between them in model structure and used features for TDNN, BLSTM and CTC/attention systems, and therefore, after fusing between them, it can get better performance.

3. Machine Translation

3.1. MT baseline

In this section, we describe how our MT system has been developed. All our models are based on the transformer architecture in [8]. We start with the TED corpus, speech-translation TED corpus, and WMT18 data that are relevant to the speech translation domain. The total size of the bilingual corpus is 6.3 million sentence pairs. We use Marian toolkit for all experiments [9] and our development set includes dev2010 and tst2010-2015. The baseline architecture of Marian mainly follows the default setting for transformer NMT except for a 6-layer transformer encode-decoder, a 0.1 label smoothing, and 0.1 dropout between transformer layer. For parameter optimization, we use synchronized ADAM [10] with learning rate 0.0003, and set up the number of noam warm-up steps as 16,000.

3.2. Preprocessing

The training data is preprocessed following standard procedure. We first use the scripts in moses toolkit[1] for punctuation normalization, tokenization and lowercasing. Afterwards, we jointly learn and apply the byte pair encoding[2] for English and German together. Figure 1 shows a detailed

[1] https://github.com/moses-smt/mosesdecoder
[2] https://github.com/rsennrich/subword-nmt

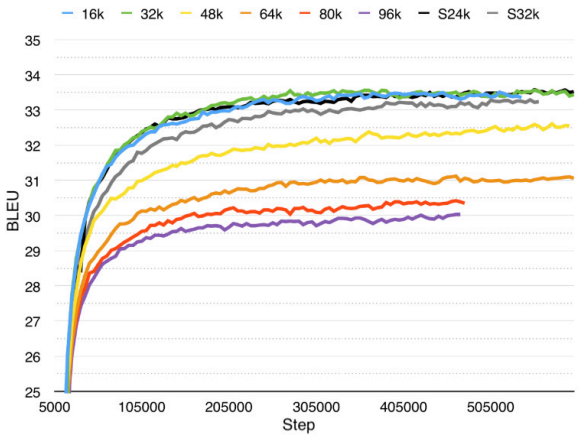


Figure 1: Performance of BPE operations including joint-bpe 16k, 32k, 48k, 64k, 80k, 96k, and shared-bpe 24K and 32K on our development set.

comparison of different BPE operations on our development set. We observe that joint bpe with 32K vocabulary performs best in this case and it is our final BPE code size. In the end, sentences longer than 100 tokens are removed.

Generally, the ASR raw output is a long stream of words with no punctuation, capitalization or segmentation markers. [11] shown that using various types of text segmenters between ASR and MT modules can improve speech translation quality. Our ASR system performed audio segmentation, punctuation and capitalization prediction. We use spaCy[3] to segment processed ASR transcription into shorter chunks. By using text segmentation we observed BLEU score gains between 0.2 and 1.0, depending on different ASR and MT systems.

For sentence boundary detection and punctuation insertion model, we experiment with a 3 layers LSTM for sequence tagging, and 3-gram KenLM[12] for additional scoring. In prediction phrase, we store some token in buffer as the foregoing context which is useful in real time prediction. Silence time in ASR is also used for sentence boundary detection. We also trained single layer self attention sequence tagging model and a bi-directional self attention LM. And we found it got much higher comma F1 score, but a litter lower period F1 score. This approach was not used in the final result and we will do more experiments in our future work.

3.3. Data Augmentation

In order to obtain a high quality domain related training corpus, we exploit the algorithm described in [13, 14], aiming at selecting sentence pairs from large out-domain corpus that are similar to the target domain. In our experiments, the 200K TED talks data is considered as in-domain corpus, and all the other parallel corpora provided are combined as a large out-domain corpus. Two 3-gram language models are trained over the source and target side of the in-domain corpus, respectively. Then, another set of two 3-gram language models are trained, whose training data is randomly selected from the out-domain corpus, with size being similar. Thus, each sentence pair from out-domain corpus is scored by the bilingual cross-entropy difference model. Finally, we sort all sentence pairs and select top ranked sentences pairs. Our experiments show that with different amount of additional data, we obtain BLEU gain on the development set between 0.4 to 1.8.

[3] https://spacy.io/

3.4. Noisy input translation

Though our transformer translation system is applied to the post-processed speech recognition outputs, the insertion, deletion and substitution errors still cannot be removed. Following the idea in [15, 16], we use the corrupted inputs to train a robust neural machine translation model. Since we observe that the insertion error is rare in our ASR system, only the deletion and substitution noises are considered for the source sentences of the regular parallel training data. In this way, the gap between training data and testing ASR output will become potentially smaller.

Basically, we first randomly delete the token of the source sentence with a small probability (0.01 and 0.02 are selected in our experiments by cross-validation). Specifically, we also pre-define a functional-words list including 120 tokens with number of letters less than 5, and assign a doubled deletion rate for them. Notice that we train several deletion-noisy models alone without any other corrupted strategy for further ensemble. In our experiments, the single model trained with deletion noise can increase the case insensitive (CI) BLEU for at least 0.5 BLEU point over the baseline on the ASR output.

Additionally, we attempt to introduce the substitution noise by randomly replacing the token with its pronunciation-like candidates with a small probability. By trying different substitution rate, we empirically found that the substitution noise model achieved no significant improvement over deletion noise model. One possible solution is to use the adaptive substitution rate of each token, estimated with the maximum likelihood in the ASR model. We will leave this as the future work.

The third strategy for noisy training is punctuation simulation. We randomly spare 30% of the regular parallel corpus and remove all punctuations from the source side, then annotate/re-generate the same data with the tool used in the ASR post-processing. We add the punctuation noisy corpus back and train our model, and empirically observe an improvement of at least 0.5 BLEU point gain compared with baseline as well. The punctuation simulation and deletion/substitution noise are typically not combined, since the underlying true noise comes from the ASR system and our two manually designed noising systems may have a large bias. Therefore, we decide to apply them separately to increase the diversity of our models in ensemble.

3.5. Refinements

3.5.1. Ensemble decoding

Model ensemble is a widely used technique to boost the performance of a MT system, which is to combine the prediction of multiple models at each decode step. We adopt the ensemble method GMSE (Greedy Model Selection based Ensemble) detailed in [17].

The candidate single models are first sorted as a list according to their performance on the development dataset. Another two model lists are maintained during the ensemble, named "keep", "redemption". For each iteration, a candidate model could be either drawn from the beginning or the end of the candidates list with probability p or p_{reserve}, or the "redemption" list with probability p_{redempt}, where $p + p_{\text{reserve}} + p_{\text{redempt}} = 1$. Then, the selected model is temporarily concatenated to the "keep" list. If the evaluation of the current model ensemble achieves a better BLEU score, the model is permanently added to the "keep" list. Otherwise, it will be put into the "redemption" list. Notice that one model from the "redemption" list can only be redeemed once, after which it is withdrawn permanently from the candidates.

In order to achieve better ensemble performance, we increase the diversity of our candidate models by introducing another 8 training schedules and further obtain about 200 single checkpoints. The greedy nature of the GMSE algorithm makes the search feasible in an acceptable time frame. In summary, we list the different training schedules as follows.

1. 10 million training corpus is selected by BPE level language models.
2. 8 million training corpus is selected by word level language models.
3. Adding another 3 million back-translation data to the original parallel corpus.
4. Training with deletion/substitution noise.
5. Training with punctuation simulation noise.
6. Fine-tuning with the 200K TED talks in-domain data.
7. Fine-tuning with the punctuation simulation data.
8. 7-layer transformer NMT model.

Due to the time limitation, we cannot do all the ablative experiments on listed strategies. However, we can still report the best single model and best ensemble result on our ASR output in Table 3. Note that our development set is the combination of tst2013 and tst2015, we did not test our model on these two dataset separately.

Table 3: Improvement with ensemble

	tst2013 + tst2015
best single model	22.96
best ensemble	23.84

3.5.2. Reranking

Besides building multiple ensemble systems with different random initialization and configurations, we also build and optimize the n-best list reranker. We follow the approach in [17] in which several neural machine translation models and language models have been experimented with. The n-best list reranker involves the following steps

- Build and optimize neural MT models including single models, ensemble models, and models with different configurations such as beam size. To improve diversity, we also use the right-to-left and target-to-source models.
- Build ngram language models from in-domain and out-of-domain data using the data selection method similar to [17, 18].
- Apply the the greedy feature selection based reranking method in [17] to train the reranker. To deal with overfiting, we use the tuning set that contains both manual transcription and our ASR transcription.

Our experiments show that depending on different settings the reranker typically obtains improvements between 0.1 to 0.4 BLEU point over the best system.

Additionally, we experiment with the multi ASR inputs for reranking. The main motivation is to directly exploit strengths of different ASR systems into the reranking system. Out initial results show that the multi ASR inputs does not outperform the input for the ASR fusion output.

3.5.3. Recaser

Since the lowercased corpus are applied to train our MT system, an additional post-processing recaser model is necessary to obtain the truecased (or capitalized) German translation output. In principle, we exploit the combination of the moses[4] SMT recaser model and the Char-RNN based neural recaser model [19]. The SMT recaser model essentially trains a word-to-word translation model and a cased language model without reordering. Unlike the word-level approach, the neural recaser model restores the case information at the character-level, reducing the recasing problem to a sequential binary classification task. No special treatment is required for mixed cased words.

The final combination rule is that for every single word, if the SMT recaser and the neural recaser reach a consensus

[4] http://www.statmt.org/moses/?n=Moses.SupportTools

on the final recasing output, we will accept it; if they have a disagreement, we will always capitalize that word. This strategy will result in a combined recaser model which is slightly better than any other one.

4. Conclusions

The IWSLT 2018 Speech Translation task provided the opportunity to compare different speech translation approaches using shared datasets and standardized evaluation metrics. Table 4 shows our submission results on the IWSLT 2018 official test set.

Table 4: IWSLT 2018 final evaluation results on contrastive and primary submissions

	con.1	con.2	primary	con.3
BLEU	22	22.16	22.36	22.5
TER	63.44	63.52	63.03	63.03
BEER	52.47	52.69	51.77	52.64
CharTER	59.56	57.54	69.26	57.76
BLEU(ci)	23.97	24.14	24.23	24.3
TER(ci)	60.44	60.41	60.22	60.15

Our participation in this task revealed three important aspects of speech translation that we regard as important for the future.

First, our experiments indicated that the speech segmentation and transcription post processing, by themselves, can make a big differences on transcription quality as well as translation quality. Furthermore, these components not only improve WER and BLEU scores, in live speech translation scenario they also greatly improve user experience.

The second aspect relates to the importance of noisy inputs. There are many types of noises in speech translation scenario. For example noise come from speech audio, transcription errors, and the nature of spoken language. Our experiments show that by modeling transcription errors directly in the neural MT (NMT) training, we obtained consistent improvement. It indicates that the NMT model becomes more robust against errors of our ASR system. Also, if we compare statistical machine translation (SMT) technique, we find SMT is generally more robust to noises than NMT. It is probably because SMT models are built on probability distributions estimated from many occurrences of words and phrases, therefore any unsystematic noise in the training only affects the tail end of the distribution.

The third aspect is the importance of model engineering. In the statistical machine learning, the best model is typically from through several rounds of feature engineering. In the NMT context, we see that our best model is also from many steps of model engineering and refinements. Given the availability and good scalability of ASR and MT toolkits today, it is tempting to throw as much model configurations as possible and let the built-in mechanisms of these learning algorithms figure out which one is the best. However, the strategy has its own limitations, and, in conjunction with the limited availability of the labeled data, can easily produce models that are under-performing on blind test sets.

5. Acknowledgments

We thank the support from Shanbo Cheng, Jun Lu for their help in reranking and recasing.

6. References

[1] A. H. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, no. 3, pp. 328–339, 1989.

[2] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Interspeech*, 2015.

[3] S. Xue and Z. Yan, "Improving latency-controlled BLSTM acoustic models for online speech recognition," in *Proc. ICASSP*, 2017, pp. 5340–5344. [Online]. Available: https://doi.org/10.1109/ICASSP.2017.7953176

[4] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: first results," in *Proc. NIPS*, vol. abs/1412.1602, 2014.

[5] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, 2014, pp. 1764–1772.

[6] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE J. Sel. Topics Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[7] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997, pp. 347–352.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.

[9] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Fikri Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, "Marian: Fast neural machine translation in C++," in *Proceedings of ACL 2018, System Demonstrations*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 116–121.

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

[11] V. K. R. Sridhar, J. Chen, S. Bangalore, A. Ljolje, and R. Chengalvarayan, "Segmentation strategies for streaming speech translation," in *Proc. of the Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, 2013, pp. 230–238.

[12] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable modified Kneser-Ney language model estimation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria, August 2013, pp. 690–696.

[13] 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*. Association for Computational Linguistics, 2011, pp. 355–362.

[14] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2013, pp. 678–683.

[15] M. Sperber, J. Niehues, and A. Waibel, "Toward robust neural machine translation for noisy input sequences," in *Proc. of the International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, December, 14-15 2017.

[16] 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 the International Workshop on Spoken Language Translation (IWSLT)*, Tokyo, Japan, December, 14-15 2017.

[17] Y. Deng, S. Cheng, J. Lu, K. Song, J. Wang, S. Wu, L. Yao, G. Zhang, H. Zhang, P. Zhang, C. Zhu, and B. Chen, "Alibaba's neural machine translation systems for wmt18," in *Proc. of the Conference on Machine Translation*. Brussels, Belgium: Association for Computational Linguistics, November 2018.

[18] B. Chen and F. Huang, "Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data," in *Proc. of the Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2016, pp. 314–323.

[19] R. H. Susanto, H. L. Chieu, and W. Lu, "Learning to capitalize with character-level recurrent neural networks: An empirical study," in *Proc. of the Empirical Methods in Natural Language Processing)*. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 2090–2095.

CUNI Basque-to-English Submission in IWSLT18

Tom Kocmi *Dušan Variš* *Ondřej Bojar*

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
<surname>@ufal.mff.cuni.cz

Abstract

We present our submission to the IWSLT18 Low Resource task focused on the translation from Basque-to-English. Our submission is based on the current state-of-the-art self-attentive neural network architecture, Transformer. We further improve this strong baseline by exploiting available monolingual data using the back-translation technique. We also present further improvements gained by a transfer learning, a technique that trains a model using a high-resource language pair (Czech-English) and then fine-tunes the model using the target low-resource language pair (Basque-English).

1. Introduction

Despite becoming the current dominant approach in the field of machine translation (MT), neural machine translation (NMT) [1] systems still perform poorly in certain scenarios. One of them is learning to translate between language pairs where only a small amount of parallel data is available. Under these circumstances the NMT model quickly overfits and its performance plummets when translating sentences not seen during training. As observed in [2], with small parallel data, NMT performs much worse than the previous approach of phrase-based MT.

There are situations where the ability to learn an MT model of a reasonable quality given only a small amount of training data can be crucial. For example, when a crisis occurs in a region where an under-resourced language is spoken, a quick deployment of an MT system translating from or to that language can make a huge difference in the impact of the provided support [3].

In this paper, we describe the CUNI submission to the IWSLT Low Resource task for translating from Basque-to-English in the domain of TED talks. Our submission is based on the recently introduced self-attentive network architecture called Transformer [4]. We improve the performance of this model by exploiting the English in-domain monolingual data using the back-translation technique [5]. We achieve further improvements via transfer learning. Transfer learning [6, 7] consists of training a "parent" (high-resource) model first and then continuing the training on the "child", low-resource, parallel data as a means of model adaptation. Furthermore, we combine several models saved at training checkpoints by simply averaging the weights ("model averaging") as a substitute of model ensembling.

The structure of the paper is the following. In Section 2, we describe the method of transfer learning followed by the description of back-translation in Section 3. The model description is presented in Section 4 and the dataset overview in Section 5. Section 6 details the results achieved by our systems. Section 7 discusses other works in the area of low-resource translation systems. And finally Section 8 concludes the paper.

2. Transfer learning

Transfer learning is based on the observation that neural machine translation model that is first trained on the parallel data of a high-resource language pair can be adapted to a lower-resource language pair. The two languages can have a linguistic relation, however, transfer learning works even for unrelated languages [7].

The method starts by first training the parent model until it reaches the best possible performance or until a fixed number of gradient updates is performed. This model is then adapted by switching the training dataset from the parent pair to the low-resource child pair. During this transition, we do not change any hyperparameters nor the learning rate.

The transfer learning method does not need any modification of the existing NMT pipeline. The method only relies on a single condition: the vocabulary has to be shared across all the languages in the parent as well as child language pairs.

We construct the shared vocabulary using subword tokens, namely wordpieces [8], instead of words. This way, we are able to handle words not seen during training by splitting them into subwords, which are present in the vocabulary. We learn the subword segmentation using concatenated source and target sides of both the parent and child language pairs. To avoid bias in the vocabulary towards the high-resource language pair, [7] suggest to sample a subset of the sentences from the high-resource language pair that has a size similar to the low-resource language pair dataset, calling this approach "balanced vocabulary". They also showed, that a significant portion of this balanced vocabulary is relevant only for the child model, as it never appears in the parent training data.

Unlike [7], we also experiment with additional vocabulary setups, using either only the parent (or only the child) training data to generate the vocabulary. We call these restricted setups "parent vocabulary" and "child vocabulary", respectively. The idea behind the use of "child vocabulary" is that there will be more child-specific wordpieces which can lead to a better performance of the child model. On the other hand, the reasoning behind the "parent vocabulary" is that we can use only a single parent for the training of several different child models and therefore save the time of training parent models for each child separately.

3. Back-translation

The organizers of IWSLT 2018 provided participants with a vast amount of English monolingual data to use in their system submissions, both in-domain and out-of-domain. We exploit the English in-domain TED talks monolingual data for creation of the synthetic data as described by [5].

The key idea is to use an MT system trained to translate in the opposite direction (English-to-Basque) and use it to translate the monolingual data. These synthetic outputs, when paired with the input monolingual data, can be then used as additional parallel data for the original (Basque-to-English) direction. Even though the source side is noisy, the additional training examples help the decoder to learn a more fluent target side language model.

We use this method to back-translate only the in-domain TED talks data because it is the target domain of the Low Resource task.

To create the synthetic parallel data from the English monolingual corpus, we used a Transformer model and transfer learning. We first trained on the English-to-Czech corpus and then adapted the model using English-to-Basque corpus. This was based on our previous experiments where transfer learning resulted in a model with a better translation performance and therefore a better quality of synthetic data.

4. Model description

We use the self-attentive neural network architecture called Transformer [4]. We chose this network architecture due to its reported state-of-the-art results [9, 10],[1] making it a strong baseline for our experiments.

The architecture follows the encoder-decoder paradigm where the encoder creates hidden representations of the source language tokens and the decoder outputs the target sequence conditioned on that source language representations and the representations of the already decoded tokens.

The self-attentive encoder contains several layers each consisting of two sublayers: the first one applies a self-attention and the second one a feed-forward network. The decoder is similar, including an additional attention-over-encoder layer between its own self-attention and the feedforward layers. The self-attention layer is the key component of the Transformer architecture, effectively modeling the context of each token and thus substituting other methods such as the recurrent hidden units [1, 11] or convolutional networks [12]. The absence of recurrent units makes the training much faster due to a possible parallelism while requiring a lower number of layers when compared to the convolutional network.

We use the Transformer implemented in Tensor2Tensor [13],[2] version 1.4.2. Our models are based on the "big single GPU" configuration as defined in the paper. We use the default setup, only changing the batch size to 2300 and a maximum sentence length to 100 wordpieces in order to fit the model to our GPUs (NVIDIA GeForce GTX 1080 Ti with 11 GB RAM).

We use Noam learning rate decay scheme with the starting learning rate of 0.2 and 32000 warm-up steps. The decoding uses the beam size of 8 and length normalization penalty is set to 0.8.

Dataset	Sentences	Tokens EN	Tokens CS/EU
Genuine EN-EU	0.9 M	7.0 M	5.1 M
Genuine EN-CS	40.1 M	563.4 M	490.5 M
Synthetic EN-EU	0.3 M	5.3 M	3.6 M

Table 1: Sizes of the parallel corpora. The "synthetic" have Basque side back-translated from English.

5. Dataset

For Basque-English, we used all the available data that were allowed by the organizers of IWSLT 2018. The parallel corpora consist of only around 5,600 in-domain (TED) sentence pairs and around 940,000 out-of-domain sentence pairs.

In addition to the resources suggested by the organizers, we also used data from OPUS and WMT, which were also allowed. Specifically, corpora PaCo2 English-Basque and QTLeap Batches 1-3 from WMT.[3]

For English-Czech, we use all parallel data available for WMT 2018 except of the Paracrawl. The majority of the data is part of the CzEng 1.7 corpus (the filtered version, [14]).[4]

We also created synthetic Basque-English data using back-translation. We generate them by translating all English sentences from the TED talks data gathered across all language pairs provided for IWSLT 2018. The data do not contain sentences from talks in test set.

From all training sentences, we dropped sentence pairs shorter than 4 words or longer than 75 words on either source or target side. This results in a speedup of the training by allowing a larger batch size. A similar setup was used in [15] where the authors argue that in their experiments, the

[1] http://www.statmt.org/wmt18/translation-task.html

[2] https://github.com/tensorflow/tensor2tensor

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

[4] https://ufal.mff.cuni.cz/czeng/czeng17

Vocabulary	CS to EN (BLEU)	EU to EN (BLEU)
Child only	24.93	22.92
Parent only	27.81	23.29
Balanced	27.93	23.63
Baseline	–	19.09

Table 2: The results of transfer learning. The first column shows the performance of the parent model, the second column is the child model based on the corresponding parent. The baseline does not use transfer learning. The results are reported on the development set. Scores are comparable only within columns.

performance is not negatively influenced by the reduction of training data.

To evaluate the models during training we used the development data provided by IWSLT 2018 (Basque-English) and development data available for WMT (Czech-English), namely WMT 2011 Newstest.

6. Results

In this section we first compare results obtained when using the three types of vocabulary and then describe our systems submitted to the IWSLT evaluation.

6.1. Effect of vocabulary

We experiment with three types of shared vocabulary as described in Section 2. All setups use the exact same data (and a same layout of the transfer learning); they differ only in the vocabulary. First, we trained three models from Czech-to-English with different vocabularies for 1M steps and then we continued with the transfer learning of the child Basque-to-English models until their performance on the validation set stopped improving.

As seen in Table 6.1, the transfer learning for Basque-to-English improves the model performance significantly over the baseline, gaining over 4 BLEU points (19.09 vs. 23.63).

When we look at the parent-only or child-only vocabulary setups, both performed worse than the balanced vocabulary. With the balanced vocabulary, we obtain the best result on the Basque-to-English translation. We suppose that the same holds for other language pairs too, since there is no language specific restriction.

Still, it would be interesting to know whether the data ratio 50:50 is the best possible setup or whether other ratios could improve the results. We plan to investigate this in our future work. We assume that the exact ratio might be language specific, however, in general, using the balanced approach with an equal representation of the languages might still be an effective option.

Run	Transfer	Back-translation	Genuine	BLEU	NIST	TER
Primary	✓	✓	✓	22.86	6.01	60.31
Contrastive 1	–	–	✓	16.13	4.98	66.55
Contrastive 2	✓	✓	–	22.26	6.00	63.89
Contrastive 3	✓	–	✓	21.11	5.84	62.34

Table 3: Results of our submissions. Official evaluation on the test set.

6.2. Final results

We submitted several contrastive models for the final IWSLT evaluation. All our systems use the same balanced vocabulary. The synthetic data were generated by the English-to-Basque system. All final models are averaged over the last 5 checkpoints.

The primary system uses the transfer learning: the parent model is trained for 1M steps on Czech-to-English, followed by transfer learning using only the synthetic data for 405k steps, and completed by 60k steps on the genuine, original parallel data.

The run labelled "Contrastive 1" in Table 3 is the baseline trained only on the official parallel Basque-to-English data.

"Contrastive 2" uses transfer learning on the parent model Czech-to-English trained for 1M steps, followed by training on only the synthetic English-to-Basque data, without the use of genuine parallel data.

Finally, "Contrastive 3" also uses transfer from Czech-to-English as the primary, followed by genuine parallel English-to-Basque data, without the use of any synthetic data.

As clearly confirmed by three automatic metrics, the combination of the back-translation and transfer learning leads to the best performance.

7. Related work

In [16], Firat et al. propose zero-resource multi-way multilingual systems, with the main goal of reducing the total number of parameters needed to train multiple source and target languages. To prevent the network from forgetting the previously learned language pairs, they implement a special training schedule.

Another multilingual approach is proposed by [8] where Johnson et al. simply use all translation pairs at the same time and the choice of the target language happens at runtime by special token at the end of the input sentence. This forces the model to learn to translate between many languages, including language pairs without available 'direct' parallel data.

The lack of sufficient amounts of parallel data can be also tackled by unsupervised translation [17, 18]. The general idea is to train monolingual embeddings using large amounts of monolingual data and finding a projection from the source to target words that preserves the structure of embedding vector spaces [19]. Using these shared fixed bilingual embeddings an architecture with a shared encoder [18] or both

shared encoder and decoder [17] is then trained using multiple training objectives.

Aside from the common back-translation [5], simple copying of the target monolingual data back to the source-side has also been shown to improve translation quality in the low-resource setting [20].

The transfer learning we used could be also seen as a variant of the so-called "curriculum learning" [21, 22], where the training data are ordered from foreign out-of-domain to the in-domain training examples to speed up the training convergence.

8. Conclusion

In this paper, we presented our systems for IWSLT 2018 low-resource Basque-to-English translation task. We reached a significant improvement using transfer learning and back-translation. We compared three types of vocabularies used for the transfer learning and concluded, that the balanced vocabulary is the best option.

9. Acknowledgements

This study was supported in parts by the grants SVV 260 453, GAUK 8502/2016, GAUK 1140218, and 18-24210S of the Czech Science Foundation. This work has been using language resources and tools stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (projects LM2015071 and OP VVV VI CZ.02.1.01/0.0/0.0/16 013/0001781).

10. References

[1] 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 - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 3104–3112. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969173

[2] P. Koehn and R. Knowles, "Six Challenges for Neural Machine Translation," in *Proceedings of the First Workshop on Neural Machine Translation*. Vancouver: Association for Computational Linguistics, August 2017, pp. 28–39. [Online]. Available: http://www.aclweb.org/anthology/W17-3204

[3] W. Lewis, R. Munro, and S. Vogel, "Crisis mt: Developing a cookbook for mt in crisis situations," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, July 2011. [Online]. Available: https://www.microsoft.com/en-us/research/publication/crisis-mt-developing-a-cookbook-for-mt-in-crisis-situations/

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 6000–6010. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[5] 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

[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*. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 1568–1575. [Online]. Available: https://aclweb.org/anthology/D16-1163

[7] T. Kocmi and O. Bojar, "Trivial Transfer Learning for Low-Resource Neural Machine Translation," in *Proceedings of the 3rd Conference on Machine Translation (WMT)*, Brussels, Belgium, Nov. 2018.

[8] M. Johnson, M. Schuster, Q. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. a. Vigas, 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://transacl.org/ojs/index.php/tacl/article/view/1081

[9] M. Popel and O. Bojar, "Training Tips for the Transformer Model," *The Prague Bulletin of Mathematical Linguistics*, vol. 110, no. 1, pp. 43–70, 2018. [Online]. Available: https://content.sciendo.com/view/journals/pralin/110/1/article-p43.xml

[10] O. Bojar, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, and C. Monz, "Findings of the 2018 conference on machine translation (WMT18)," in *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*. Brussels, Belgium: Association for Computational Linguistics, October 2018.

[11] 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

[12] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 1243–1252.

[13] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. Gomez, S. Gouws, L. Jones, L. Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer, and J. Uszkoreit, "Tensor2Tensor for Neural Machine Translation," in *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*. Boston, MA: Association for Machine Translation in the Americas, March 2018, pp. 193–199. [Online]. Available: http://www.aclweb.org/anthology/W18-1819

[14] O. Bojar, O. Dušek, T. Kocmi, J. Libovický, M. Novák, M. Popel, R. Sudarikov, and D. Variš, "Czeng 1.6: enlarged czech-english parallel corpus with processing tools dockered," in *International Conference on Text, Speech, and Dialogue*. Springer, 2016, pp. 231–238.

[15] T. Kocmi, oman Sudarikov, and O. Bojar, "CUNI Submissions in WMT18," in *Proceedings of the 3rd Conference on Machine Translation (WMT)*, Brussels, Belgium, Nov. 2018.

[16] O. Firat, K. Cho, and Y. Bengio, "Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism," 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. 866–875. [Online]. Available: http://www.aclweb.org/anthology/N16-1101

[17] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised neural machine translation," in *Proceedings of the Sixth International Conference on Learning Representations*, April 2018.

[18] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," *CoRR*, vol. abs/1710.04087, 2017.

[19] M. Artetxe, G. Labaka, and E. Agirre, "Learning bilingual word embeddings with (almost) no bilingual data," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 451–462. [Online]. Available: http://aclweb.org/anthology/P17-1042

[20] A. Currey, A. V. M. Barone, and K. Heafield, "Copied monolingual data improves low-resource neural machine translation," in *Proceedings of the Second Conference on Machine Translation*, 2017, pp. 148–156.

[21] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.

[22] T. Kocmi and O. Bojar, "Curriculum Learning and Minibatch Bucketing in Neural Machine Translation," in *Recent Advances in Natural Language Processing 2017*, Sept. 2017.

Fine-tuning on Clean Data for End-to-End Speech Translation: FBK @ IWSLT 2018

Mattia Antonino Di Gangi[1,2], *Roberto Dessì* [1]*, *Roldano Cattoni*[2], *Matteo Negri*[2], *Marco Turchi*[2]

[1]University of Trento, Italy
[2]Fondazione Bruno Kessler, Italy
[1]roberto.dessi@unitn.it
[2]{digangi,cattoni,negri,turchi}@fbk.eu

Abstract

This paper describes FBK's submission to the end-to-end English-German speech translation task at IWSLT 2018. Our system relies on a state-of-the-art model based on LSTMs and CNNs, where the CNNs are used to reduce the temporal dimension of the audio input, which is in general much higher than machine translation input. Our model was trained only on the audio-to-text parallel data released for the task, and fine-tuned on cleaned subsets of the original training corpus. The addition of weight normalization and label smoothing improved the baseline system by 1.0 BLEU point on our validation set. The final submission also featured checkpoint averaging within a training run and ensemble decoding of models trained during multiple runs. On test data, our best single model obtained a BLEU score of 9.7, while the ensemble obtained a BLEU score of 10.24.

1. Introduction

End-to-end speech translation (that is, the direct translation of an audio signal without intermediate transcription steps) has recently gained increasing interest in the scientific community thanks to the recent advances of neural approaches in the related ASR and MT fields [1, 2, 3, 4, 5]. Effective approaches to the task can become a useful solution to deal with languages that do not have a formal writing system [6], as it is possible to create a collection of spoken utterances with their respective translations in a more common language. We can also expect that, in the future, end-to-end speech translation systems will overcome problems related to the cumulative effect of speech recognition errors introduced in pipelined architectures. FBK's submission to the IWSLT 2018 Speech Translation (ST) task relies on a single model that takes as input features extracted from an English audio signal and returns as output a written translation in German. As the input is not in raw wave form, one might argue that the "end-to-end" denomination does not fit in this formulation of the task. Nevertheless, since feeding the network with the input features released by the task organizers was allowed, we adhere to the looser definition of "end-to-end" implicit in this year's task formulation.[1]

Our system was trained using the state-of-the-art sequence-to-sequence model based on LSTMs and CNNs introduced in [2]. Considering the high number of experiments to run, and the high number of epochs needed to train a speech translation model (up to 87 in the case of our final submission), the model was implemented using the *fairseq*[2] [7] sequence-to-sequence learning toolkit from Facebook AI Research. The tool, which is tailored to NMT, was adapted to the ST task showing considerable reductions in training time compared to the same models implemented on other platforms (from hours to minutes in the processing of the same amount of training instances).

One of the main challenges we faced was how to maximize the usefulness of the available training data by weeding out noisy (and potentially harmful) instances. For this purpose, we developed the two data cleaning procedures described in Section 2. The architectural choices and the main implementation details of our system are described in Section 3. In Section 4, we report the results on our validation set, which were obtained by using different data conditions and hyper-parameters. Section 5 concludes the paper with final remarks.

2. Data Cleaning

Our submission was obtained by a model solely trained with the data released for the speech translation task. Before building the model, we devoted particular attention to the quality of the training material, aiming to reduce the possible impact that noise in the data can have on training time and model convergence. Indeed, the initial training set of 171,121 instances comprised elements featuring either a partial alignment between the audio signal and the correspond-

* Work performed during an internship at FBK

[1]Our work has been pursued during a summer project with the goal of gaining hands-on expertise in this new promising field with the simplification of a standardized data set.

[2]http://github.com/facebookresearch/fairseq

ing transcription, or a skewed ratio between the number of feature frames and the characters in the transcription. To identify and weed out such noisy and potentially harmful training items, we applied two cleaning procedures. Both the procedures take advantage of the available English transcriptions of the audio signals[3] and were run in cascade, after the removal of $1,000$ items to be used as our development set. As discussed in Section 4.1, though smaller in size, the resulting subsets of the original training corpus yielded performance improvements on development data, especially when used for fine-tuning a model trained on the original unfiltered corpus.

2.1. Cleaning Based on Alignment

Starting from the initial training corpus of $170,121$ instances (called "Parallel" henceforth), the first cleaning step was aimed to identify and remove the items featuring a poor alignment between the audio signal and the text. Assuming that the English and German texts are parallel, the potential noise introduced by such instances is represented by wrong transcriptions/translations (either totally inadequate or containing spurious words) of the original source signal. To identify them, our approach was to align each audio signal with the corresponding English transcription and then decide what to retain based on the alignment quality (i.e. considering unaligned words as evidence of noise). We performed the alignment on a sentence-by-sentence basis using Gentle,[4] a forced aligner based on Kaldi.[5] After the alignment, we removed all the training instances in which at least one word in the transcription was not aligned with the corresponding audio segment. This strict cleaning policy (due to time limitations, we did not experiment with less aggressive strategies) resulted in the removal of $24,240$ instances, which reduced the initial "Parallel" corpus to $145,881$ items. Henceforth, the corpus resulting from this first cleaning step will be referred to as "Clean 1".

2.2. Cleaning Based on Frames/Characters Ratio

The second cleaning step was aimed to identify and remove from "Clean 1" the training instances featuring a skewed ratio between the number of feature frames and the characters in the transcription. In this case, the potential noise is due to portions of the original speech that correspond to long silences, background noise (e.g. laughter and applause), or words that are not present in the transcription/translation. To identify such possible outliers, looking at the ratios reported in Figure 1, we decided to cut the distribution so to retain only the training instances belonging to ratio bins that contain at least $5,000$ items. The corresponding cutting values of 3.5 and 7.5 resulted in the removal of $29,898$ instances,

[3]Note that data cleaning is the only phase in which we used the English transcriptions. Being this step independent from the actual system training, our approach is still fully end to end.

[4]https://lowerquality.com/gentle/

[5]http://kaldi-asr.org/index.html

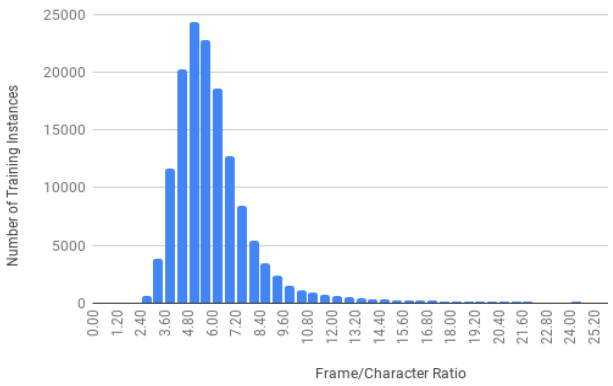


Figure 1: Distribution of the training instances in terms of the ratio between the number of feature frames and the characters in the transcription.

which further reduced our training corpus to $115,983$ items. Henceforth, the corpus resulting from our second cleaning step will be referred to as "Clean 2".

3. Seq2seq Speech Translation model

We re-implemented the seq2seq ST model introduced in [2], which uses an encoder-decoder-attention architecture based mainly on LSTMs [8]. The source-side input length is some order of magnitudes higher than the decoder side, and thus some reduction in the temporal dimension was performed using 2-D CNNs with stride (2, 2). The decoder is inspired by the early deep-transition decoder used in Nematus [9], which stacks two LSTM units in a way that the single LSTMs are not recurrent by themselves, while the stack of the two is globally recurrent. A schema of the model is depicted in Figure 2.

3.1. Encoder

The input to the encoder is a variable-length audio sequence with 40 features for each time step. At first, the input sequence is processed by two time-distributed densely-connected layers with size of 256 and 128 respectively, each followed by a *tanh* activation. The output of the densely-connected layers is then processed by two stacked 2-dimensional convolutional layers, each having a 3×3 kernel and stride $= 2$. Let n be the sequence length and f be the number of input features to the first convolutional layer. The output of the first convolution is of size $(16, n/2, f/2)$ and for the second convolution is of size $(16, n/4, f/4)$. The 16 filters are then flattened to obtain an output of size $(n/4, 4 \times f)$, which is subsequently processed by a stack of three bidirectional LSTM layers [10]. The initial state of the LSTM is initialized as a zero vector at the beginning of the training, but then it is optimized via back-propagation together with the rest of the network. We found that training the initial state gives a boost in performance and speeds up

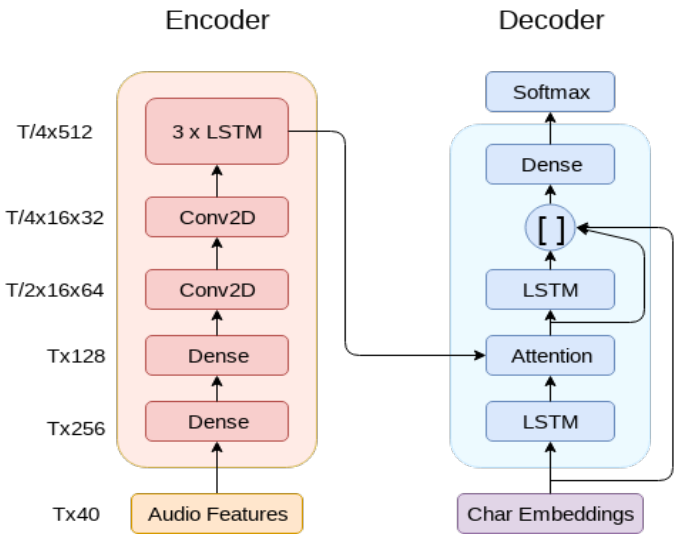


Figure 2: Schema of our end-to-end model architecture. The numbers on the left represent the dimensionality of each encoder layer's output. The batch size is not written.

the model convergence.

3.2. Decoder

The decoder consists of a two-layered deep-transition LSTM [9] followed by a deep output layer [11]. The input of the first layer is the character embedding of the last character. The output of the first layer is used as a query vector to compute an attention over the last layer of the encoder. The attention output is then used as input to the second LSTM layer. The hidden and cell states received as input by the two LSTM layers are, for every time step, the last hidden and cell states produced by the other LSTM layer. The last encoder output is averaged over the time dimension and this new tensor is passed as input to two different densely-connected layers with *tanh* non-linearity. The two functions compute the initialization of the hidden and cell states for the first LSTM layer. The deep output is a densely-connected nonlinear function, which takes as input the concatenation of the LSTM output, the attention output and the current symbol (character) embedding, and outputs a tensor of size 512. This tensor is finally multiplied by a second character embedding matrix to compute the scores over the whole vocabulary.

3.3. Attention

The attention layer computes a distribution of weights that sums up to 1 for the encoder output sequence (soft attention) with no positional information (global attention). The scores for each encoder position are computed according to their relevance with respect to the decoder state. The relevance score is computed using the general attention score proposed in [12].

3.4. Increased Regularization

Due to the small size of the training data, we found useful to apply some regularization tricks. The first and more common technique is the dropout applied to each layer [13]. Instead of variational dropout [14], we preferred to use the fastest implementation of LSTMs provided by the Pytorch library, which uses regular dropout.

Besides dropout, we applied weight normalization and label smoothing as additional techniques for regularization. Weight normalization [15] is a simple technique that decomposes the parameter matrices into their magnitude and direction components in order to easily produce a transformation that scales the weights and reduces the gradient covariance to zero. The result is a faster convergence and a limitation of the weight space, which has a regularizing effect.

Label smoothing [16] smooths the cross-entropy cost function by giving a weight of 0.9 to the probability of the correct symbol, and 0.1 to the sum of the probabilities of the other symbols. Label smoothing makes the model less confident on its predictions, producing a regularizing effect. In NMT, it has been observed that, despite the increased loss and perplexity usually obtained with this technique, the translations are usually better [17] and end up in improved BLEU [18] scores.

4. Experiments

In this section we summarize the experiments that motivated our choices for the final submission. Since the goal of our participation was to explore the potential of a single end-to-end model that can translate directly from audio signals, we used as training data only the Speech Translation TED Corpus that was released for the task. No pre-training has been performed on different types of data (such a pre-training would in fact rely on ASR data). All our models were trained using the Adam optimizer [19] with learning rate of 0.001, and values for β_1 and β_2 of 0.9 and 0.999. We applied dropout of 0.2 to all layers, including the input features. The norm of the gradients was clipped to 5. All the models have been trained until convergence according to the loss on a held-out set of $1,000$ sentences (see Section 2). The results achieved by each model on the validation set are reported Tables 1–4.

At first, we experimented with the reference implementation of the sequence-to-sequence model[6] that is based on Tensorflow [20]. However, with about 3.5 hours per epoch on a single NVIDIA GTX-1080 GPU, its training time resulted to be incompatible with the need of quickly testing a range of alternative solutions. To avoid this bottleneck, we re-implemented the same model within the *fairseq* toolkit, which is highly optimized to significantly reduce training time. Our re-implementation was indeed faster, with a reduction of the training time to about 30 minutes per epoch for the largest version of the training corpus ("Parallel"), and

[6] https://github.com/eske/seq2seq

Data	Val. BLEU
Parallel	8.54
Clean 1	8.98
Clean 2	8.54

Table 1: Results of the base model over the three different versions of the dataset.

Data	Val. BLEU
P $\rightarrow$ C1	9.55
P $\rightarrow$ C2	9.85
C1 $\rightarrow$ C2	9.89
P $\rightarrow$ C1 $\rightarrow$ C2	**10.14**

(a) Dataset fine-tuning

Strategy	Val. BLEU
Adam annealing	9.11
NAG annealing	8.74

(b) Restart strategy

Table 2: (a) Results for the base model in different fine-tuning conditions. P stands for Parallel, C1 for Clean 1 and C2 for Clean 2. Only the last row refers to a double step of fine-tuning. (b) Results with two different restart strategies for the model trained on Clean 2.

about 20 minutes per epoch for the smallest one ("Clean 2"). The wall clock time of a single training run was around 30 hours, with a maximum of 10 additional hours for the fine-tuning.

4.1. Dataset Selection

In the first round of experiments, we were interested in understanding the impact of the data cleaning procedures described in Section 2. To this aim, we trained the base system on the three different versions of the dataset (i.e. "Parallel", "Clean 1" and "Clean 2") and evaluated the resulting models on the same validation set. The results listed in Table 1 show that Clean 1 provides us with the best result, but Clean 2 leads to a result equivalent to Parallel despite using about 36% less data. Thus, we decided to use Clean 2 for the following experiments in order to have faster training cycles.

4.2. Dataset Fine-tuning and Restart Strategy

In this subsection we address two questions. The first one is whether it is useful to fine-tune a model trained on a larger dataset by using a smaller and cleaner subset of the same corpus. The second question is whether a restart strategy with learning rate annealing can improve the performance.

The first question was addressed by restarting the training of the model by using the new, smaller dataset as training set, but with the same training policy and hyper parameters. The results listed in Table 2a show that fine-tuning the model on cleaner data always helps. In particular, fine-tuning on Clean 2 (which is smaller but of higher quality) always results in better performance, especially in the case of a double step of fine-tuning (P $\rightarrow$ C1 $\rightarrow$ C2). Interestingly, also using only the clean data (C1 $\rightarrow$ C2) yields better results than training the initial model on the original Parallel corpus.

To address the second question, we used the model trained on Clean 2 and restarted the training on the same training set with a policy of learning rate annealing. To this aim, the learning rate was multiplied by 0.5 every time the validation loss did not improve over the best one computed so far [21]. We experimented using both Adam with annealing and SGD with Nesterov Accelerated Gradient (NAG) [22] with annealing. The results listed in table 2b show that, though Adam with annealing yields a better model, both the BLEU scores are at least 0.45 points less than the worse model with fine-tuning.

Model	Val. BLEU
AST Seq2Seq	8.54
+ Weight Normalization (WN)	8.69
+ Label Smoothing (LS)	8.74
+ Sigmoid Attention	8.44
+ WN and LS	**9.69**

Table 3: Results on the data cleaned with two cleaning steps.

4.3. Features Exploration

In this round of experiments we trained our base model on the Clean 2 dataset and compared its result with models that have weight normalization, label smoothing, sigmoidal attention instead of softmax attention, and weight normalization and label smoothing together. The results on the validation set, which are listed in Table 3, show that both weight normalization and label smoothing give a small contribution, while the sigmoidal attention slightly decreases the translation quality. Moreover, the joint addition of label smoothing and weight normalization gives a sensibly higher boost, suggesting that the models need high regularization. Considering the scarce amount of data, the need for high regularization was expected. However, it is interesting to note that by increasing the dropout to 0.3 the base model converges to a much worse point.[7] From now on, we call the model with weight normalization and label smoothing "full modell".

4.4. Experiments with Full Model

Once we found that the full model is clearly better than the others, we replicated the experiments on all the datasets with the new model. In the second column of Table 4a, we can see that this model is more sensitive to noise. In fact, training it with the "Parallel" set leads to poor performance in translation (4.66 BLEU), but this lower translation quality was not expected by looking at only the training and validation

[7] Observed in preliminary experiments, not reported here.

Data	BLEU
Parallel	4.66
Clean 1	9.69
Clean 2	9.69

(a)

Data	Best	Avg	Test
P → C1	10.26	10.46	-
C1 → C2	9.71	10.42	-
P → C2	10.63	10.90	9.70
P → C1 → C2	10.41	10.78	-
+ Adam annealing	10.50	10.59	-
Ensemble of 4	-	**11.60**	**10.24**

(b)

Table 4: Results using different versions of the dataset for training our model with weight normalization and label smoothing.

losses. Nonetheless, the fine-tuning of this model on cleaner data, whose results are listed in table 4b, leads to improvements ranging from 0.57 to 0.94 BLEU points with respect to the models trained only on the clean data.

Unfortunately, the score of 10.63 of the best model (P→C2) represents only a limited improvement when compared with the best model in the second column of Table 2a (P→C1→C2), which improved from 8.54 of the base model to 10.14. The fifth row of Table 4b shows the results when the last fine-tuning is performed using Adam with annealing instead of Adam with a fixed learning rate. Based on these results, we submitted our single best model (P → C2 Avg) as our contrastive submission.

4.5. Checkpoint Averaging and Ensemble Decoding

Checkpoint averaging consists in computing the average of different checkpoints of the same training. In [23], it has been shown that, in neural machine translation, it leads to a better translation quality than using a single model. For each model, we computed the BLEU score on the validation set for the last 10 checkpoints, and averaged the weights of all the models whose results are less than 0.5 BLEU points worse than the best one. The improvement can be observed by comparing the Best and Avg columns of Table 4b.

We also performed ensemble decoding of models trained in different runs. The ensemble involved all the Avg checkpoints listed in table 4b, except for "C1→ C2", which was trained using a different vocabulary. The ensemble of the 4 models obtained a result of 11.60 BLEU on the validation set.

4.6. Submitted Systems and Results

Based on the outcomes of the above experiments on development data, we opted for submitting the following systems:

- **Primary**: ensemble of 4 systems (Section 4.5).
- **Contrastive**: Checkpoint averaging of P→C2 (Table 4b).

The result of the primary system is 11.60 BLEU score on our validation set and 10.24 on the test set, whereas the contrastive system scored, respectively, 10.90 and 9.70 in the validation and the test set.

5. Conclusions

We described FBK's participation in the end-to-end speech translation task at IWSLT 2018. We have shown that data cleaning is useful in reducing the training time by discarding a good portion of the training data, while not hurting translation quality. We have also observed that fine-tuning a model using a cleaner dataset can bring improvements up to 1.6 BLEU points. Moreover, regularizing the model with normalization and label smoothing can produce an improvement of more than 1.0 BLEU point with clean datasets, but the same model fails to converge to a good point using all the data. In addition, using checkpoint averaging and ensemble decoding gave us another gain of 1.0 BLEU point. The final score on this year's test set is of 9.70 and 10.24 BLEU respectively for our best single model and for the primary submission based on ensemble decoding. In order to improve the competitiveness of this system, our next experiments will include ASR for pretraining the encoder [24] or for multi-task learning [4].

6. Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

7. References

[1] A. Bérard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," *arXiv preprint arXiv:1612.01744*, 2016.

[2] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," in *ICASSP 2018-IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.

[3] A. Anastasopoulos and D. Chiang, "Tied multitask learning for neural speech translation," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, vol. 1, 2018, pp. 82–91.

[4] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," 2017.

[5] A. Anastasopoulos and D. Chiang, "Leveraging translations for speech transcription in low-resource settings," in *Proceedings of Interspeech 2018*, 2018.

[6] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn, "An attentional model for speech translation without transcription," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 949–959.

[7] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International Conference on Machine Learning*, 2017, pp. 1243–1252.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] 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 *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, pp. 65–68.

[10] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[11] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[12] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[13] 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.

[14] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Advances in Neural Information Processing Systems*, 2015, pp. 2575–2583.

[15] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 901–909.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[17] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, N. Parmar, M. Schuster, Z. Chen, *et al.*, "The best of both worlds: Combining recent advances in neural machine translation," *arXiv preprint arXiv:1804.09849*, 2018.

[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*, 2002, pp. 311–318.

[19] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of ICLR 2015*, 2015.

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: a system for large-scale machine learning."

[21] 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*, vol. 108, no. 1, pp. 13–25, 2017.

[22] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[23] M. Junczys-Dowmunt, T. Dwojak, and H. Hoang, "Is neural machine translation ready for deployment? a case study on 30 translation directions."

[24] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," *arXiv preprint arXiv:1809.01431*, 2018.

The JHU/KyotoU Speech Translation System for IWSLT 2018

Hirofumi Inaguma[†]*, Xuan Zhang, Zhiqi Wang, Adithya Renduchintala, Shinji Watanabe, Kevin Duh*

Center for Language and Speech Processing, Johns Hopkins University, Baltimore, USA
[†]Graduate School of Informatics, Kyoto University, Japan
{xuanzhang,zwang132,adithya.renduchintala,shinjiw}@jhu.edu
inaguma@sap.ist.i.kyoto-u.ac.jp, kevinduh@cs.jhu.edu

Abstract

This paper describes the Johns Hopkins University (JHU) and Kyoto University submissions to the Speech Translation evaluation campaign at IWSLT2018. Our end-to-end speech translation systems are based on ESPnet and implements an attention-based encoder-decoder model. As comparison, we also experiment with a pipeline system that uses independent neural network systems for both the speech transcription and text translation components. We find that a transfer learning approach that bootstraps the end-to-end speech translation system with speech transcription system's parameters is important for training on small datasets.

1. Introduction

We report on our efforts on the IWSLT 2018 Speech Translation task. The goal of the 2018 task is to build and evaluate English-to-German speech translation systems on the domain of lectures and TED talks. We build two systems:

- Pipeline System: English (EN) speech transcription system using a joint CTC-attention model (Section 3.1), followed by a English-to-German (EN-DE) text translation system using a RNN-based sequence-to-sequence model (Section 3.3).
- End-to-End System: English-to-German (EN-DE) speech translation system using an RNN-based sequence-to-sequence model transferred from the joint CTC-attention model (Section 4).

The main challenge is to develop end-to-end neural systems that are trainable given the small amount of data (of English speech matched to German text). We find that bootstrapping the end-to-end system with the parameters of an English-only speech transcription system (i.e. ASR of English speech to English text) was helpful.

Generally, we are interested in comparing the relative merits of end-to-end vs. pipeline approaches. Currently, our pipeline system outperforms the end-to-end system, even when trained on the same number of utterances, suggesting that there is much room for future work in end-to-end models.

[†] Work carried out as a visiting scholar at JHU.

2. Data

We build our systems on the following provided corpora:

1. Speech-Translation TED corpus (ST TED): This data contains English speech (EN-s), the corresponding English transcription (EN-t), as well as the German translation (DE-t). We use this to train both pipeline and end-to-end systems. In particular, (EN-s,EN-t) is used to train the pipeline's speech transcription component (Section 3.1); (EN-t, DE-t) is used to train the pipeline's text translation component (Section 3.3); and (EN-s, DE-t) is used to train the end-to-end sytem (Section 4).
2. TED LIUM corpus (TEDLIUM2): This data contains English speech (EN-s) and their English transcription (EN-t). We use this as additional data to train the pipeline system's speech transcription component, which is also used to initialize the end-to-end system.
3. WMT 2018 data, filtered to the TED domain using Moore-Lewis data selection [1] (WMT-Filtered): We trained 5-gram Engish language models on TED (LM_{TED}) and a random sample of the WMT data (LM_{WMT}), then selected the top 1 million WMT bitext according to the perplexity difference between LM_{TED} and LM_{WMT}. Finally, we filtered all sentences that were longer than 100 tokens or had an out-of-vocabulary rate (with respect ST TED dictionary) of 10% or larger. This is used to augment the training data for the pipeline's text translation component.

For data preprocessing of transcriptions and translations in all languages, we normalized punctuation and performed tokenization using the Moses scripts[1]. For both the pipeline's speech transcription and the end-to-end speech translation, we used a fixed vocabulary of 5k or 10k wordpieces, which were composed from characters to words and generated using sentencepiece[2]. We used the same dictionary including both EN and DE wordpieces to capture the common words in both languages.

[1] normalize-punctuation.perl and tokenizer.perl in https://github.com/moses-smt/mosesdecoder
[2] https://github.com/google/sentencepiece

For the pipeline's text translation component, we experimented with different kinds of subword units. For simplicity, we did not use any truecase models for any systems and worked directly with the natural casing. Table 1 shows the data sizes in each corpus.

For feature extraction for speech transcription and translation, we extracted 80-channel log-mel filterbank outputs with 3-dimensional pitch features computed with a 25ms window and shifted every 10 ms using Kaldi [2]. The features were normalized by the mean and the standard deviation on the whole training set (excluding our development set). We removed utterances having more than 3000 frames or more than characters due to the GPU memory capacity.

corpus	#utterance	speech datasize
Speech-Translation TED	166,214	271 hours
TEDLIUM2	258,943	210 hours
WMT-Filtered	988,697	-

Table 1: Data size in each corpus.

3. Pipeline System

3.1. Speech Transcription Component

In this section, we briefly describe the joint CTC-attention framework for the speech transcription (i.e. ASR) component. Let $\mathbf{x} = (x_1, \ldots, x_T)$ be acoustic features and $\mathbf{y} = (y_1, \ldots, y_U)$ be the corresponding target sentence in the same language as $\mathbf{x}$.

3.1.1. Connectionist Temporal Classification (CTC)

Connectionist Temporal Classification (CTC) [3] is a latent variable model which directly maps the input sequence into the output sequence of shorter length. To compensate the differences of sequence lengths, CTC introduces an additional "blank" symbol. The CTC loss function is defined as the summation of negative log probabilities of all possible paths mapped from ground truth labels interleaved with blank labels.

$$\begin{aligned} L_{ctc} &= -\ln P(\mathbf{y}|\mathbf{x}) \\ &= -\ln \sum_{\pi \in B^{-1}(\mathbf{y})} P(\pi|\mathbf{x}) \end{aligned}$$

where π represents a CTC path, and B represents a collapse function which maps all the CTC paths into the unique ground truth labels $\mathbf{y}$ by removing all blank labels. Based on the conditional independence assumption, posterior probabilities $P(\pi|\mathbf{x})$ is factorized frame by frame as follows:

$$P(\pi|\mathbf{x}) = \prod_{t=1}^{T} P(\pi_t|h_t)$$

where h_t represents an activation of the top layer of the encoder. $P(\pi|\mathbf{x})$ is effectively calculated by the forward-backward algorithm.

3.1.2. Attention-based encoder-decoder

Attention-based encoder-decoder [4, 5] is another sequence labeling model which directly predicts output sequences. Unlike the CTC framework, this approach does not make any conditional independence assumptions, where the model predicts each token conditioned on all previous tokens.

$$P(\mathbf{y}|\mathbf{x}) = \prod_{u=1}^{U} P(y_u|y_1, \ldots, y_{u-1}, \mathbf{x})$$

Attention-based encoder-decoder model consists of two modules: the encoder and decoder. The encoder network maps input features $\mathbf{x}$ into high-level distributed representation $\mathbf{h}$, and the decoder network picks up a portion of $\mathbf{h}$ with a scoring function given encoder and decoder hidden states, which is called the attention mechanism. We used the location-aware scoring function, which takes previous attention weights into account. The loss function is designed as the negative log probabilities as follows:

$$L_{att} = -\ln P(\mathbf{y}|\mathbf{x})$$

3.1.3. Joint CTC-attention

We introduce the multitask learning (MTL) framework with the CTC objective in the training of the attention-based encoder-decoder model [6]. This approach has two advantages: 1) it encourages monotonic alignments in the encoder network, which leads to fast convergence and removes inappropriate alignments in long sequences, 2) it leads to sequence-level optimization. The loss function of the joint CTC-attention framework is designed as an interporation of L_{ctc} and L_{att} with a tunable parameter λ ($0 \leq \lambda \leq 1$):

$$L_{mtl} = \lambda L_{ctc}(\mathbf{y}|\mathbf{x}) + (1-\lambda) L_{att}(\mathbf{y}|\mathbf{x})$$

In addition, scores from CTC outputs are taken into account in the beam search decoding of the attention-based model during the inference stage [7, 8]. Because CTC is frame-synchronous, hyper-parameters tuning such as length penalty and coverage penalty are not neccesary any more in order to prune inappropriate hypotheses.

3.2. Evaluation of Speech Transcription Component

Preprocessing: We used the Speech-Translation TED corpus augmented with TED LIUM corpus, totaling 481h. With regard to the official development sets provided by the IWSLT organizers (*dev2010*, *tst2013* etc.), there is no segmentation information of the start and end time of utterances. Therefore, we sampled 4k utterances from the Speech-Translation TED corpus as the validation set, and removed them from the original training data. For evaluation, we segmented each audio file in the development sets with the LIUM SpkDiarization tool [9] first, then performed MWER segmentation with the toolkit from RWTH [10] as

in the baseline implementation provided by organizers[3].

Architecture: We built end-to-end ASR models with the ESPnet toolkit [11] with a pytorch backend [12]. For the encoder part, we used 2-layers CNN layers with max-pooling layers followed by 3 or 5-layers of 1024 dimensional bidirectional LSTM [13], resulting in 4-fold time reduction. For the decoder part, we used 2-layers of 1024 dimensional LSTM. We did not conduct regularization such as dropout, label smoothing [14, 15], scheduled sampling [16] for speech transcription.

Optimization: Our systems were optimized with the AdaDelta algorithm with epsilon annealing for 15 epochs. The weight for CTC loss λ was empirically set to 0.5.

Decoding: We conducted beam search decoding with beam width 20. LSTM language model of 2 layers with 650 hidden units trained on the same parallel corpus was used.

Results: Results for the TEDLIUM2 corpus are shown in Table 2. 5k wordpiece units are always better than 10k in this corpus. We also conformed the consistent improvements with deeper encoders (3 layers $\rightarrow$ 5 layers). Results for the official development sets are shown in Table 3. As in Table 2, 5k units are better than 10k units, but we cannot see improvements by adding encoder layers. We suspect that this is due to the quality of audio segmentation by the LIUM SpkDiarization tool and utterance matching by the RWTH MWER tool.

#unit	#layer	dev	test
10k	3	13.8	12.3
5k	3	12.1	11.1
10k	5	13.3	12.5
5k	5	**11.6**	**10.7**

Table 2: Word error rate (WER) evaluated on the TEDLIUM2 corpus. **#unit** represents the number of untis in the softmax layer. **#layer** represents the number of BiLSTM layers following CNN layers in the encoder network.

#unit	#layer	dev2010	test2010	test2013	test2014	test2015
10k	3	28.2	29.5	31.9	32.6	46.5
5k	3	**25.6**	**27.7**	**30.6**	**31.1**	**44.4**
10k	5	28.8	31.2	33.1	34.5	45.6
5k	5	27.4	30.8	32.0	33.7	47.9

Table 3: Word error rate (WER) evaluated on the official development sets.

3.3. Text Translation Component

Preprocessing: We built neural machine translation (NMT) systems for the English-German text translation component of our pipeline system. These systems were trained on the ST TED corpus, with English manual transcript for speech recognition on the source side and corresponding German translation on the target. Training data were tokenized and split into subwords using Byte Pair Encoding (BPE) [17]. We set the number of BPE merge operations to be 20k for the source side — same for the target side. The validation set used for early stopping consists of around 4k utterances, and they were randomly sampled from the corpus.

Architecture: The attention-based NMT models consist of two components: an encoder network, which is a recurrent neural network (RNN), that provides a representation of the input sentence, and a decoder network, which is also a RNN, that generates translation based on the input context with attention mechanism [5, 18] applied.

We trained our NMT systems with Sockeye [19]. In the model we used based on hyper-parameter tuning, the encoder and decoder both had 2 layers with 512 LSTM hidden units on each layer and we applied dot product attention for RNN decoders. Both the source and target embedding vectors were set to 512. We used word-count based batch of size 4096 words and maximum sequence length 100. For regularization, the RNN inputs and states dropout rates for both the encoder and the decoder were set to 0.1.

Optimization: Our systems employed the Adam optimizer to reduce the cross-entropy loss with an initial learning rate 0.0005. We made a checkpoint after every 2000 batch updates, and if the model had not improved in perplexity on the validation data for more than 8 checkpoints, we would perform early stopping for the training process. In general, it takes around 50 epochs (about 10 hours) for the model to converge.

#BPE merge ops	20k	30k	40k	50k
avg dev BLEU	23.83	24.18	24.28	23.77

Table 4: Effect of different number of BPE merge operations on average BLEU score on development sets. The initial learning rate was set to 0.0007.

Hyper-parameter Tuning: Hyper-parameters were tuned based on systems' average decoding performance (BLEU score) on *dev2010*, *tst2010*, *tst2013*, *tst2014* and *tst2015* set. We searched the number of BPE merge operations from 20k, 30k, 40k and 50k, word embedding size and the number of RNN hidden units from 512 and 1024, batch size from 4096 and 6000, initial learning rate from 0.0002 to 0.0007, dropout probability from 0.1 and 0.2[4].

When searching for a good hyper-parameter configuration, we found that a more complex model, in terms of RNN hidden size, was not necessarily needed to get better performance on this corpus: when we increased the number of

[3]https://github.com/isl-mt/SLT.KIT

[4]Due to time and computational resources limitation, we only tried a subset of all the possible combinations.

system	dev2010	tst2010	tst2013	tst2014	tst2015
Manual: NMT (ST TED) on EN reference	23.96	27.54	26.23	22.30	25.07
Pipeline: NMT (ST TED) on ASR output	15.47	20.54	16.68	14.35	16.21
Manual: NMT (ST TED + WMT-Filtered) on EN reference	28.07	30.59	29.47	26.04	26.70

Table 5: BLEU comparison of NMT translating English reference (Manual) or ASR ouput (Pipeline). BLEU scores are evaluated on development sets using multi-bleu.perl with the Moses tokenization.

RNN hidden units from 512 to 1024, the average BLEU score dropped from 24.71 to 24. Another interesting finding was that with other hyper-parameters fixed, when the number of BPE operations increased, the BLEU score on the development data tended to first go up and then decrease (see Table 4). Finally, the initial learning rate turned out to be an important hyper-parameter to tune. For example, we got 23.44 BLEU with initial learning rate 0.0003, but 24.18 BLEU with 0.0007.

3.4. Evaluation of Pipeline System

We show the main results of our pipeline systems in Table 5. For the purpose of comparison, we provided the NMT systems with either the manual transcripts (*Manual*) or the output of our ASR system (*Pipeline*), which is described in Section 3.1. As expected for error cascading in pipeline systems, BLEU scores drop substantially, by up to 36.4%, when translating noisy ASR outputs compared to translating the clean English transcript.

A paired permutation test shows that *Manual* outperforms *Pipeline* statistically significantly with p-value $< 1\%$. NMT systems trained with good manual transcripts might be intolerant to various ASR errors, and it is very likely they will propagate the errors during decoding.

Additionally, the final row in Table 5 we show the BLEU scores of the NMT system trained with additional WMT-Filtered data. There is a large improvement, for example from 23.96 to 28.07 on the *dev2010*. This confirms that adding more bitext helps.

While the corpus-level BLEU score of the pipeline is lower than the manual system, we did observe some interesting variances at the level of individual sentences: it is not the case that translations of ASR outputs are always worse than translations of manual, clean transcripts. Figure 1 compares the sentence-level BLEU scores in three different scatter plots. For each sentence in *tst2010*, we have the English transcript (EN-ref) and the resulting translation (DE-manual) by our NMT system; we also have the English ASR output (EN-ASR) and the resulting translation (DE-pipeline). Finally we have the correct German reference (DE-ref). We then computed three sentence-level BLEU scores (with add-one smoothing) as follows:

- Manual BLEU: BLEU of DE-manual vs. DE-ref
- Pipeline BLEU: BLEU of DE-pipeline vs. DE-ref
- ASR BLEU: BLEU of EN-ASR vs. EN-ref

Our goal is to compare ASR BLEU (which measures whether the English sentence was difficult to transcribe) with Manual/Pipeline BLEU. Our original hypothesis is that sentences with low ASR BLEU should result in a larger difference in Manual BLEU minus Pipeline BLEU.

Interestingly, as seen in Figure 1 (c), there are individual sentences where Manual BLEU is less than Pipeline BLEU. An example is shown in Table 6. The difference between the English reference and the ASR output is "where it gets" vs "what gets", which are arguably both correct. However, the NMT result is very different, one translating perfectly and the other not. It appears that since NMT output has high variance, i.e. it can output very different translations even when the inputs are semantically similar.

4. End-to-End System

In this section, we describe our end-to-end speech translation model and transfer learning from pre-trained ASR model.

4.1. Model for End-to-End Speech Translation

We used an attention-based encoder-decoder model for the end-to-end speech translation model. The architecture of the encoder is exactly the same as that in the ASR model in Section 3.1 (VGG-like CNN layers followed by stacked BiLSTM layers). The decoder includes two modification from the ASR decoder: (1) adopting input-feeding mechanism [18], and (2) adding scheduled sampling [16].

It is possible to integrate a language model during the decoding stage (i.e. *shallow fusion* [20]) and also training stage (i.e. *deep fusion* [21] and *cold fusion* [22]), but we did not use any language models for the speech translation task in this paper. We'll leave them to the future work.

4.2. Transfer learning from ASR

In our preliminary experiments, it took too much time to train end-to-end speech translation models from scratch, i.e. many epochs are required for convergence. Therefore, we explored methods to better initialize our end-to-end model.

Speech translation can be viewed as a combination of ASR and MT tasks, so we can treat the encoder and decoder networks as having roles in ASR and MT, respectively. Therefore, it is a natural choice to initialize the encoder with that of a pre-trained ASR model. Initialing the decoder with pre-trained MT model will be left to the future work.

Weiss et al. [23] shows improvements of BLEU scores

English Reference	But here's where it gets interesting.	**NMT Result (*Manual*)**	Aber hier ist das, was interessant wird.
ASR Output	But here's what gets interesting.	**NMT Result (*Pipeline*)**	Aber hier wird es interessant.

Table 6: An example where Pipeline system outperforms the Manual system (100 sentBLEU vs. 6.5 sentBLEU). The German reference for the utterance is *Aber hier wird es interessant .*

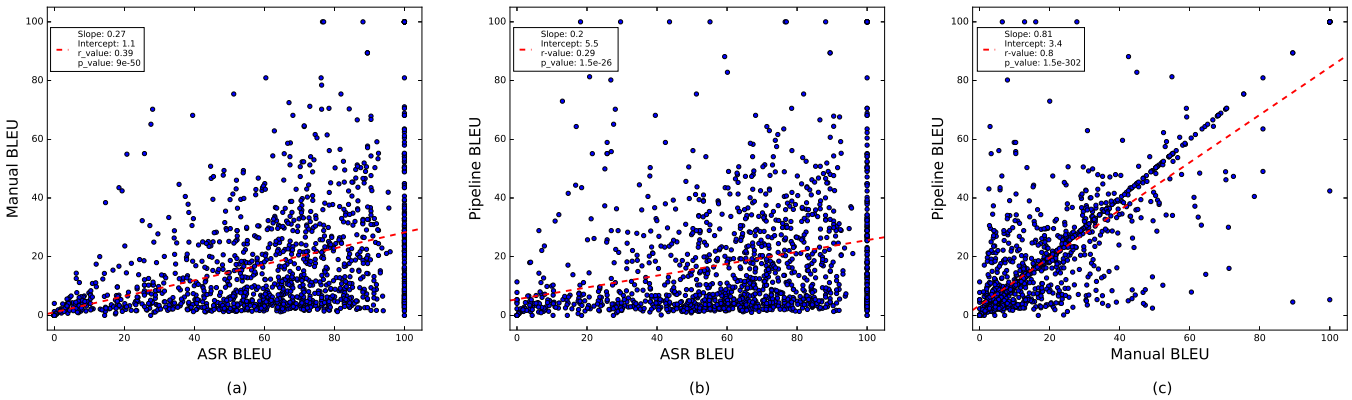


Figure 1: Sentence-level BLEU of *manual*, *pipeline* and *ASR* system on *tst2010*. A linear least-squares regression is calculated for each pair of systems.

by multi-task learning (MTL) with ASR task by sharing the encoder. Berard et al. [24] also shows both MTL and pre-training strategies lead to fast convergence and better results for end-to-end speech translation. Here we chose the pre-training strategy and transfer the weights of the encoder in the ASR model to the end-to-end model prior to training.

4.3. Experiments

We did the same data preprocessing as speech transcription in Section 3.2. We also built end-to-end speech translation models with the ESPnet toolkit with a pytorch backend. The differences of the architecture, optimization, and decoding from speech transcription models are as follows:

- We did not use the CTC framework due to its monotone assumption
- We used scheduled sampling with probability 0.2
- We ran for 30 epochs
- We did not perform beam search decoding (i.e. greedy decoding)
- We did not use any language models (due to time constraints)

We use the official scripts from the organizer and calculated case-sensitive BLEU scores with multi-bleu-detok.perl in the Moses toolkit after detokenization. We report BLEU scores in Table 7 for both pipeline systems and end-to-end speech translation models (E2E). There are several observations:

First, we can confirm that better ASR models led to better BLEU scores in the pipeline systems when comparing Table 3 and Table 7. The two ASR models with 5k units have the lowest WER scores, and the resulting two pipeline systems (b) and (d) also achieved the best BLEU scores. Second, it seems challenging to train an E2E speech translation model from scratch. Transfer learning with parameters from an existing ASR model gave consistent gains.

Finally, there are large differences between pipeline and end-to-end systems. For example, on *dev2010*, E2E trained from scratch achieved a BLEU of 4.44, E2E with transfer from ASR achieved a BLEU of 6.71, and the pipeline systems achieved BLEU in the range of 14. This may be due to data sparseness. Perhaps the explicit intermediate representation of transcripts in the language of the speech input is important for constraining the model complexity. Further, 10k wordpieces is a relatively large unit size for speech models and the data needs of an end-to-end model may be larger than that of a pipeline model.

We show some examples of the end-to-end speech translation model transferred from pre-trained ASR (system (f) in Table 7) in Table 8. Despite the low BLEU scores in general, the end-to-end model sometimes do generate reasonable sentences and correctly predicts keywords such as proper nouns and numbers. Our system was robust to misspelling because we used 10k units for the vocabulary. The official development sets include many long sentences, and it appears that our E2E model may be doing relatively worse compared to Pipeline systems on long sentences.

5. Discussion

We described our pipeline and end-to-end speech translation systems for IWSLT 2018. For the official evaluations, we submitted the pipeline system (a) in Table 7 as a contrastive

System	Configuration	dev2010	test2010	test2013	test2014	test2015
(a) Pipeline	ASR (10k unit, 3 layer); NMT (ST TED)	14.22	13.62	14.21	11.73	10.68
(b) Pipeline	ASR (5k unit, 3 layer); NMT (ST TED)	14.68	**14.70**	**15.08**	**12.23**	**11.59**
(c) Pipeline	ASR (10k unit, 5 layer); NMT (ST TED)	14.54	13.05	14.60	11.73	11.16
(d) Pipeline	ASR (5k unit, 5 layer); NMT (ST TED)	**14.87**	13.76	14.75	11.58	10.96
(e) E2E	train from scratch	4.44	4.10	3.57	3.52	2.42
(f) E2E	transfer learning from ASR parameters	6.71	6.21	6.01	5.08	4.51

Table 7: BLEU evaluated on the development sets using the official scripts provided by organizers. Note the results here are not comparable to Table 5 due to differences in the tokenization and evaluation scripts.

EN(Ref)	In the last five years we've added 70 million tons of CO2 every 24 hours – 25 million tons every day to the oceans.
DE (Ref)	In den letzten 5 Jahren haben wir 70 Millionen Tonnen an CO2 produziert alle 24 Stunden – 25 Millionen Tonnen jeden Tag in die Ozeane.
DE (Hyp)	In den letzten fnf Jahren haben wir die 70 Millionen Tonnen CO2 / h. Wir haben die Ostkste
EN(Ref)	But not just any mission, it's a mission that is perfectly matched with your current level in the game.
DE (Ref)	Aber nicht nur irgendeine Mission, sondern eine Mission, die perfekt zu Ihrem aktuellen Level im Spiel passt, richtig?
DE (Hyp)	Aber nicht nur die Mission, sondern nur eine Mission, die sich perfekt antreibt. Mit dem deren auf dem Spiel.

Table 8: Examples of the end-to-end speech translation model (system (f) in Table 7)

system and the E2E system (f) in Table 7 as the primary system; they were our best systems at the time of submission. Our main findings are that (1) pipeline systems can be very strong systems, and that (2) more work is needed to train end-to-end systems effectively, especially in small datasets.

For the official development sets, we had to use other tools to segment audio files before decoding and then match the number of references and hypotheses after decoding. We found that this affected WER and BLEU scores seriously due to misalignment. Therefore, the exact segmentation information for acoustic features is desired for the future evaluation in speech translation.

6. Acknowledgements

We would like to thank Pamela Shapiro for providing the Moore-Lewis data selection toolkit.

7. References

[1] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the ACL 2010 Conference Short Papers*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 220–224. [Online]. Available: http://www.aclweb.org/anthology/P10-2041

[2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *Proc. of ASRU*, 2011.

[3] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. of ICML*, 2006, pp. 369–376.

[4] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. of NIPS*, 2015, pp. 577–585.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. of ICLR*, 2015.

[6] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Preceedings of ICASSP*. IEEE, 2017, pp. 4835–4839.

[7] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," *arXiv preprint arXiv:1706.02737*, 2017.

[8] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[9] S. Meignier and T. Merlin, "Lium spkdiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, 2010.

[10] O. Bender, R. Zens, E. Matusov, and H. Ney, "Alignment templates: the rwth smt system." in *IWSLT*, 2004, pp. 79–84.

[11] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[12] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of CVPR*, 2016, pp. 2818–2826.

[15] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," in *Proc. of Interspeech*, 2017.

[16] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Preceedings of NIPS*, 2015, pp. 1171–1179.

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

[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] F. Hieber, T. Domhan, M. Denkowski, D. Vilar, A. Sokolov, A. Clifton, and M. Post, "Sockeye: A toolkit for neural machine translation," *arXiv preprint arXiv:1712.05690*, 2017.

[20] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," 2017.

[21] 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.

[22] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," *arXiv preprint arXiv:1708.06426*, 2017.

[23] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

[24] A. Bérard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, "End-to-end automatic speech translation of audiobooks," *arXiv preprint arXiv:1802.04200*, 2018.

Adapting Multilingual NMT to Extremely Low Resource Languages FBK's Participation in the Basque-English Low-Resource MT Task, IWSLT 2018

Surafel M. Lakew[†+]*, Marcello Federico*[⋆]

[†]University of Trento, [+]Fondazione Bruno Kessler, Trento, Italy
[⋆]Amazon AI, East Palo Alto, CA 94303, USA
lastname@fbk.eu, [⋆]marcfede@amazon.com

Abstract

Multilingual neural machine translation (M-NMT) has recently shown to improve performance of machine translation of low-resource languages. Thanks to its implicit transfer-learning mechanism, the availability of a highly resourced language pair can be leveraged to learn useful representation for a lower resourced language. This work investigates how a low-resource translation task can be improved within a multilingual setting. First, we adapt a system trained on multiple language directions to a specific language pair. Then, we utilize the adapted model to apply an iterative training-inference scheme [1] using monolingual data. In the experimental setting, an extremely low-resourced Basque-English language pair (i.e., $\approx$ 5.6K in-domain training data) is our target translation task, where we considered a closely related French/Spanish-English parallel data to build the multilingual model. Experimental results from an *i*) in-domain and *ii*) an out-of-domain setting with additional training data, show improvements with our approach. We report a translation performance of 15.89 with the former and 23.99 BLEU with the latter on the official IWSLT 2018 Basque-English test set.

1. Introduction

The amount and diversity of model training data have been shown to affect the performance of Neural Machine Translation (NMT) system [2]. The direct relation between dataset size and performance of NMT [3], calls for alternative approaches to improve low-resource language translation.

Multilingual models that constitute more than one language pair has been shown to improve the translation performance of the low-resources language direction [4, 5]. In it's simplified and most effective setting, building an M-NMT system requires only an additional "language-flag" on the data level. Then, the attentional encoder-decoder based NMT model can be trained with the aggregation of several language pairs. The flag functions as a mechanism to trigger and direct the generation of target tokens in a specific target language. Thus, when the training set is constructed with the merge of several language directions, the latent transfer-learning across languages within the conventional NMT architecture showed to improve low-resourced language pairs. However, M-NMT training mechanism is biased towards generating the language pair with the largest portion of training data [1]. This bias will consequently limit the expected level of improvement in translating low-resource language pairs.

In this work, we propose a progressive adaptation of a multilingual model to a single language pair. We cast the adaptation stage in iterative training-inference operations that utilize monolingual data. Assuming, the availability of a low-resource language pair and a high resource related/language pairs data, we specifically explore the following two mechanisms:

- Adapting a multilingual model trained with several language directions to a specific low-resourced language pair, with the aim to avoid ambiguities at the time of inference.
- Then, applying an iterative training-inference using monolingual data of the low-resourced pair, with the aim to acquire a more cleaner pseudo-parallel corpus for the next adaptation stage.

In our experimental setting, we apply the above two mechanisms for improving the extremely low-resourced (ELR) Basque(EU)-English(EN) language pair. Then, with the experimental results and discussion we present our participation of the IWSLT-2018[1] shared task on Low Resource MT of TED[2] talks from Basque to English direction. We evaluated our approach with the *i*) ELR training condition in a constrained in-domain data, and *ii*) by adding an out-of-domain training data in addition to the in-domain. We train both models in a similar language setting (i.e., the additional/related language pairs are French/Spanish-English).

For comparing our approach, we train a bilingual (Basque-English) baseline and multilingual baseline model by adding more data from the related language pairs. More specifically, to build the M-NMT model Basque-French and

(*) Work conducted while this author was at FBK.

[1] https://sites.google.com/site/iwsltevaluation2018/TED-tasks
[2] https://wit3.fbk.eu/

Basque-Spanish with a similar ELR condition, and French-English and Spanish-English with the relatively high resource data size are added to the bilingual model. All models share common configurations at training and inference time, unless stated differently. Models are trained following [4], preprocessing and training procedures using the Transformer model [6].

In the following sections, we begin by introducing NMT (§2). Following, we review the related work in multilingual models and transfer-learning (§3). In Section 4, we describe our model training approach, followed by dataset and preprocessing, experimental settings, and baseline models (§5). Finally, we give further analysis on the experimental results in Section 6.

2. Neural Machine Translation

A standard state-of-the-art NMT system comprises an encoder, a decoder and an attention mechanism, which are all trained with maximum likelihood in an end-to-end fashion [7]. Although there are different variants of the encoder-attention-decoder based approach, Recurrent variants being the predominant until recently [8], this work utilizes the "Transformer" model [6]. The encoder is purposed to encode a source sentence into hidden state vectors, whereas the decoder uses the last representation of the encoder to predict symbols in the target language. In a broad sense, the attention mechanism improves the prediction process by deciding which portion of the source sentence to emphasize at a time [9]. Nevertheless, in the Transformer architecture, the application of attention spans to the representation of encoder latent and decoder latent space.

The Transformer architecture works by relying on a self-attention (*intra-attention*) mechanism, removing all the recurrent operations that are found in the RNN approach. In other words, the attention mechanism is repurposed to compute the latent space representation of both the encoder and the decoder sides. However, with the absence of recurrence, *positional-encoding* is added to the input and output embeddings. Similarly, as the time-step in a recurrent network, the positional information provides the Transformer network with the order of input and output sequences.

In our work, we use the absolute positional encoding, but very recently the use of the relative positional information has been shown to improve performance [10]. The model is organized as a stack of encoder-decoder networks that works in an auto-regressive way, using the previously generated symbol as input for the next prediction. Both the decoder and encoder can be composed of uniform layers, each built of sub-layers, i.e., a multi-head self-attention layer and a position wise feed-forward network (FFN) layer. The multi-head sub-layer enables the use of multiple attention functions with a similar cost of utilizing attention, while the FFN sub-layer is a fully connected network used to process the attention sublayers; as such, FFN applies two linear transformations on each position and a ReLU [6].

3. Related Works

3.1. Multilingual NMT

Prior to the introduction of a shared attention mechanism [11], early works in multilingual NMT utilizes separate encoder, decoder and an attention mechanism to support the translation of either many-to-one [12], or one-to-many [13] language directions. Moreover, Firat et al. [11] introduced a many-to-many system, however, relying on separate encoder-decoder setup. In a simplified yet delivering better performance [4] and [5] introduced a "language-flag" based approach that shares the attention mechanism and a single encoder-decoder networks to enable multilingual models. In this work, we follow the Johnson et al. [4] approach for prepending a language-specific flag at the source side of the training and inference examples.

3.2. Transfer Learning and Model Adaptation

Zoph et al., (2016) [14], proposed how transfer-learning between two NMT models can improve a low-resourced MT task. In their approach, a language pair with the relatively large amount of data is first utilized to train a parent model, then the encoder-decoder parameters are transferred to initialize a child model for a low-resourced language pair. After initializing, in the fine-tunning stage, the parameters of the child decoder network is fixed. The main motivation behind updating only the encoder parameters is that the decoder language across the parent-child models stays the same. Similarly, the parent-child approach has been extended to analyze the effect of using related languages on the source side of the encoder-decoder network [15].

In a related way to benefit the low-resource language from the high resourced pair [16] proposed an alternative transfer-learning approach built on a component that allows to share lexical and sentence level representations of multiple source language to a single target language. In a prior work, a multi-source approach where two or more encoders shares an attention mechanism has been suggested in [17], to address the ambiguities of translating a source token to a single target language. Unlike [18] where a single multilingual model is used for several language translations [19] showed how adapting the multilingual model on a specific language pair improves performance. Recently [20] explored the advantage of initializing a low-resource language pair training using a pre-trained multilingual model showing a significant improvement over baseline approaches.

4. Adaptation from a Multilingual Model

This work aims to exploit the transfer-learning across languages, however, instead of the parent-child strategy [14], we rely on using a multilingual model as in [4] that allows to abstract the representation of several languages in a single attentional encoder-decoder model. We hypothesis if data is received both for the low and high resourced language

pairs, training a single model with the concatenation of all the data and progressively fine tunning it with the low-resource (*target-task*) language pair can avoid possible ambiguities between languages at the time of inference.

First, we train a model with all the available language pairs (including the target-task). Second, we adapt the best performing model to the target-task language pair. Unlike the recently proposed approach [20], we adapt using the same target-task data that has been utilized for training the baseline multilingual model. The main reason behind this is that the target-task data is already received at time of training the multilingual model. Then, the (latest) adapted model is used to perform back-translation [21] in a target $\rightarrow$ source direction or an iterative dual-inference in a $source \leftrightarrow$ target directions [1]. However, both inference approaches are used to create a source side synthetic data, the dual-inference requires an available monolingual data both from the source and target language. More importantly, the fact that we adapt from the multilingual to a bi-directional model allows us to avoid the use of auxiliary models (i.e., a separate model trained in a target $\rightarrow$ source direction) to perform the inference operations. After the inference stage, we continue training the model by combining the target-task and the newly formed source (synthetic) $\rightarrow$ target parallel data, consequently creating a progressive adaptation stages.

In the experimental section, the adaptation and progressive update of the multilingual model to the single language pair (Basque-English) target-task are evaluated in two settings:

- `iELR`, an extremely low-resource language pair trained and evaluated using an in-domain parallel and monolingual data.
- `oELR`, an extension of the `iELR` training condition with an additional out-of-domain parallel and monolingual data, as described in Section 5

In the following Section, the details of the experimental setup are given for the two evaluation scenarios.

5. Experiments

5.1. Dataset

The experimental setting covers the Basque (EU), English (EN), French (FR), and Spanish (ES) languages. The `ELR` language pair (EU-EN) and the related language pairs (FR-EU/EN, and ES-EU/EN) are categorized into the in-domain and out-of-domain settings. The in-domain data are extracted from the publicly released shared task dataset, WIT[3] TED corpus [22]. Where as the the out-of-domain dataset is collected from the WMT evaluation campaign PaCo corpus [23, 24], Opus corpus [25], and the Open Data Euskadi Repository (OpenData)[3]. Monolingual datasets for the Eu-EN pair are extracted from the TED, Opus18, and OpenData sources and preprocessed by removing the overlapping segments with the parallel data. Note; EN is the only available in-domain monolingual data, whereas the rest is collected from the out-of-domain sources based on availability. Table 1 summarizes the source and data size of each language direction.

	TED	Opus16/18	PaCo	OpenData
`EU-EN`	5623	856314	130359	-
`EU-FR`	5815	689358	-	-
`EU-ES`	5546	840458	-	926203
`FR-EN`	287134	-	-	-
`ES-EN`	277093	-	-	-
`EU-Mono`	-	-	-	741254
`EN-Mono`	242831	503970	-	-

Table 1: *Languages and dataset size of the training set. TED represents the in-domain data, whereas the Opus from the 2016 and 2018 (excluding the FR-EN and ES-EN pairs), PaCO for the EU-EN pair, and OpenData for the EU-ES pair represent the out-of-domain pairs.*

For evaluating the target-task (EU-EN) a development set of 1140 segments and for reporting the official submission results, the 2018 test set constituting 1051 source side segments are used from the TED talks in-domain data.

5.2. Preprocessing

We first tokenize the raw data and remove sentences longer than 70 tokens. As in [4], we prepend a "language-flag" on the source side of the corpus for all multilingual models. The internal sub-word segmentation [26] provided by the Tensor2Tensor library[4] is used before each training and inference. Note that prepending the "language-flag" on the source side of the corpus is specific to the multilingual models. Following the recommendation in [27], the number of segmentation rules is set to $16K$ for the in-domain data and $32K$ for the out-of-domain data.

5.3. Experimental Settings

All systems are trained using the Transformer [6] model implementation in the Tensor2Tensor library. For all trainings, we use the Adam optimizer [28], with an initial learning rate constant of 2 and a dropout [29, 30] of 0.2. The learning rate is increased linearly in the early stages ($warmup\_training\_steps$=$16,000$) and afterward it is decreased with an inverse square root of the training step.

Considering the two training scenario (i.e., `iELR` and `oELR`), we utilize two model configurations; *i*) for the in-domain data a 512 embedding and hidden units dimension, and 6 layers of self-attention encoder-decoder network, and *ii*) for an out-of-domain scenario the dimension is set to 1024. The training batch size is of $4,096$ sub-word tokens.

[3] http://hltshare.fbk.eu/IWSLT2018/OpendataBasqueSpanish.tgz

[4] https://github.com/tensorflow/tensor2tensor/tree/v1.6.2/tensor2tensor

	Round	NMT	M-NMT	iELR	oELR
Eu-En	I	3.10	13.37	12.96	22.48
	II	-	-	**15.65**	22.72
	III	-	-	15.15	**23.14**

Table 2: *BLEU results on the dev*2018 *using the EU-EN single language pair* NMT *and the multilingual* M-NMT *baseline models, as compared to the in-domain* iELR *and the out-of-domain* oELR *adapted multilingual models from three training rounds. The bold highlight shows the best performing training rounds.*

At inference time, we employ a beam size of 4 and a batch size of 32.

Following [6], iELR experiments are run upto 100k training steps, whereas oELR experiments are run upto 400K steps, i.e., all models are observed to converge within these steps. The consecutive adaptation converged in a variable training steps, however, to make sure a convergence point is reached, all restarted experiments are run for additional 50K steps. Then, the best performing checkpoint on the dev set is used in the next training stage. All models are trained on a Tesla V100-pcie-16gb with a single GPU for iELR and 4 GPU's for oELR.

5.4. Baseline Models

Baseline: models are trained as a term of comparison in two settings, *i*) using only the available in-domain EU-EN data, refereed to as NMT, and *ii*) by adding the related language (EU-FR/ES and FR/ES-EN) in-domain data on the EU-EN target-task. The latter forms a multilingual (M-NMT) baseline model. The following section, discusses the results and the comparison between the baselines and the adapted model types.

6. Results and Discussion

The baseline models (NMT and M-NMT) compared to against the adapted multilingual (iELR and oELR) models are reported in Table 2. The single language pair model trained with the in-domain (≈5.6K) training data showed a performance of 3.10 BLEU. As we expected, the poor performance is directly related to the small amount of training data. In case of the M-NMT, we observed an improvement of +10.27 over the NMT with a performance of 13.37 BLEU. As discussed in Section 1, the transfer-learning across languages, that arise from the additional EU-ES/FR and FR/ES-EN in-domain language pairs highly contributed for the observed improvement. Moreover, the experiments with our suggestion have been run for two consecutive rounds.

In-Domain Setting. In the first adaptation stage, the iELR model showed no significant difference with the baseline M-NMT. However, the adaptation stage helps to narrow the translation direction to the target task and avoid possible ambiguities for the inference stage. In the second round the iELR model showed a +2.28 BLEU improvement over the M-NMT (13.37 BLEU) baseline. The improvement is expected for the reason that the model is trained with the additional pseudo-parallel corpus from the back-translation step of the EN in-domain monolingual data to the EU target. In the consecutive round, however, the model performance degrades after the back-translation stage. This is likely caused by poorly generated source side synthetic EU from the EN monolingual data. Thus, for the final evaluation we take the best performing model from the second training round.

	NMT	M-NMT	iELR	oELR
Basque-English	-	-	15.89	23.99

Table 3: *Official BLEU results of tst*2018 *evaluated using the in-domain* iELR *and the out-of-domain* oELR *best performing adapted multilingual models.*

Out-of-Domain Setting: oELR models are trained in a similar training strategy with iELR, except the availability of additional parallel and monolingual (both for EU and EN, see Table 1) data. The relatively higher amount of training data, contributed for the larger gain of the oELR model over the in-domain training condition. Compared to the baseline models (NMT and M-NMT), oELR showed the highest performance with 23.14 BLEU score at the third training round. Unlike the performance degradation observed in the iELR setting, the availability of monolingual data both for EU and EN benefits each training-inference stage. However, with only a 0.66 BLEU gain over the initial model after three rounds, we observed that the domain mismatch between parallel (EU-EN) and the monolingual data disadvantages the expected improvement using the training-inference approach.

In case of, the official evaluation campaign, this work focused on a primary submission using the oELR model and a contrastive submission using the in-domain iELR model. Table 3, shows the performance of the two models on *test-2018*.

An interesting aspect from the multilingual adaptation and the iterative training-inference stages is that improvements are observed within 6k-20k steps. Meaning, the continued training approach from the latest adapted model shows a faster convergence than training a model from scratch. Overall, our approach aimed at training a baseline multilingual model for a progressive adaptation to a target-task (i.e., EU-EN), and applying an iterative training-inference scheme using monolingual corpora showed to improve over the baseline model. Our results suggest that the progressive adaptation is critical when the target-task language pair has new additional data at each stage. The experimental findings have brought our attention for a further study on how to adapt a multilingual model and what type of monolingual data to utilize in the training-inference stages.

7. Conclusions

In this work, we showed how progressively adapting a multilingual model to an extremely low-resourced (EU-EN) language pair improves the translation performance, with an additional training-inference stage that utilizes monolingual data. To evaluate the approach, the experimental setting is carried out in an in-domain (iELR) and out-of-domain (oELR) scenarios. Results show a significant improvement over a single language pair model (NMT), as well as a 2.28 BLEU increase over the baseline M−NMT model in an in-domain setting. As future work, we will focus on improving the joint iterative training-inference and progressive adaptation stages.

8. Acknowledgements

This work has been partially supported by the EC-funded project ModernMT (H2020 grant agreement no. 645487). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

9. References

[1] S. M. Lakew, Q. F. Lotito, N. Matteo, T. Marco, and F. Marcello, "Improving zero-shot translation of low-resource languages," in *14th International Workshop on Spoken Language Translation*, 2017.

[2] P. Koehn and R. Knowles, "Six challenges for neural machine translation," *arXiv preprint arXiv:1706.03872*, 2017.

[3] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico, "Neural versus phrase-based mt quality: An in-depth analysis on english-german and english-french," *Computer Speech & Language*, vol. 49, pp. 52–70, 2018.

[4] 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.

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

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

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

[8] 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.

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

[10] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[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] 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.

[13] 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.

[14] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," *arXiv preprint arXiv:1604.02201*, 2016.

[15] T. Q. Nguyen and D. Chiang, "Transfer learning across low-resource, related languages for neural machine translation," *arXiv preprint arXiv:1708.09803*, 2017.

[16] J. Gu, H. Hassan, J. Devlin, and V. O. Li, "Universal neural machine translation for extremely low resource languages," *arXiv preprint arXiv:1802.05368*, 2018.

[17] B. Zoph and K. Knight, "Multi-source neural translation," *arXiv preprint arXiv:1601.00710*, 2016.

[18] S. M. Lakew, Q. F. Lotito, T. Marco, N. Matteo, and F. Marcello, "Fbks multilingual neural machine translation system for iwslt 2017," in *14th International Workshop on Spoken Language Translation (IWSLT 2017)*, 2017, pp. 35–41.

[19] N.-Q. Pham, M. Sperber, E. Salesky, T.-L. Ha, J. Niehues, and A. Waibel, "Kits multilingual neural machine translation systems for iwslt 2017." IWSLT, 2017.

[20] G. Neubig and J. Hu, "Rapid adaptation of neural machine translation to new languages," *arXiv preprint arXiv:1808.04189*, 2018.

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

[22] 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.

[23] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. J. Yepes, P. Koehn, V. Logacheva, C. Monz, *et al.*, "Findings of the 2016 conference on machine translation," in *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, vol. 2, 2016, pp. 131–198.

[24] I. San Vicente, I. Manterola, *et al.*, "Paco2: A fully automated tool for gathering parallel corpora from the web." in *LREC*, 2012, pp. 1–6.

[25] J. Tiedemann, "Parallel data, tools and interfaces in opus." in *Lrec*, vol. 2012, 2012, pp. 2214–2218.

[26] 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.

[27] M. Denkowski and G. Neubig, "Stronger baselines for trustable results in neural machine translation," in *Proceedings of the First Workshop on Neural Machine Translation*, 2017, pp. 18–27.

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

[29] 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.

[30] 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.

Learning to Segment Inputs for NMT Favors Character-Level Processing

Julia Kreutzer[*], *Artem Sokolov*[◇,*]

[*]Computational Linguistics, Heidelberg University, Germany
[◇]Amazon Research, Germany

kreutzer@cl.uni-heidelberg.de, artemsok@amazon.com

Abstract

Most modern neural machine translation (NMT) systems rely on presegmented inputs. Segmentation granularity importantly determines the input and output sequence lengths, hence the modeling depth, and source and target vocabularies, which in turn determine model size, computational costs of softmax normalization, and handling of out-of-vocabulary words. However, the current practice is to use static, heuristic-based segmentations that are fixed before NMT training. This begs the question whether the chosen segmentation is optimal for the translation task. To overcome suboptimal segmentation choices, we present an algorithm for dynamic segmentation, that is trainable end-to-end and driven by the NMT objective. In an evaluation on four translation tasks we found that, given the freedom to navigate between different segmentation levels, the model prefers to operate on (almost) character level, providing support for purely character-level NMT models from a novel angle.

1. Introduction

Segmentation of input sequences is an essential preprocessing step for neural machine translation (NMT) and has been found to have a high positive impact on translation quality in recent WMT shared task evaluations [1, 2]. This success can be explained statistically, since shorter segments are beneficial for reducing sparsity: They lower the type-to-token ratio, decrease the number of out-of-vocabulary (OOV) tokens and singletons, which improves the coverage of unseen inputs.

Two subword segmentation methods are presently the state-of-the-art in NMT: the *byte-pair encoding* (BPE), that starts with a dictionary of single characters and iteratively creates a new entry from the two currently most frequent entries [3, 4], and a similar *wordpiece* (WP) model [5].

While being empirically more successful than word-based NMT, both BPE and WP are preprocessing heuristics, they do not account for the translation task or the language pairs at hand (unless applied to both sides jointly), and require additional preprocessing for languages that lack explicit word separation in writing. Being used in a pipeline fashion, they make it impossible for an NMT system to resegment an unfavorably presplit input and require consistent application of the same segmentation model during testing, which adds an integration overhead and contributes to the 'pipeline jungles' in production environments [6].

On the other extreme from word-based NMT models lie purely character models. Their advantages are smaller vocabularies, thus smaller embedding and output layers, allowing for more learning iterations within a training time budget to improve generalization [7], and no preprocessing requirements. At the same time, longer input sequences aggravate known optimization problems with very large depths of time-unrolled RNNs [8] and may require additional memory for tracking gradients along the unrolling steps.

In this work[1], we pose the following question: **what would the input segmentations look like if the NMT model could decide on them dynamically?** Instead of heuristically committing to a fixed (sub)word- or character-segmentation level prior to NMT training, this would allow segmentation for each input to be driven by the training objective and avoid solving the trade-offs of different levels by trial and error. To answer this question, we endow an NMT model with the capacity of adaptive segmentation by replacing the conventional lookup embedding layer with a 'smart embedding' layer that sequentially reads input characters and dynamically decides to group a block of them into an output embedding vector, feeding it to the upstream NMT encoder before continuing with the next block (with an optional reverse process on the target side). To signal that a block of characters, encoded as an embedding vector, is ready to be fed upstream, we use accumulated values of a scalar halting unit [9], which learns when to output this block's embedding. It simultaneously affects weighting probabilities of intermediate output vectors that compose the output embedding. Thanks to this weighting, our model is fully differentiable and can be trained end-to-end. Similarly to BPE, it has a hyper-parameter that influences segmentation granularity, but in contrast to BPE this hyper-parameter does not affect the model size. While we evaluate our on-the-fly segmentation algorithm on RNN-based NMT systems, it is transferable to other NMT architectures (CNN [10] or Transformer [11]), since it only replaces the input embedding layer. Empirically, we find a strong preference of such NMT models to operate on segments that are

Work was done while the first author was interning at Amazon, Berlin.

[1]Extended report: https://arxiv.org/abs/1810.01480.

only one to a few characters long. This turns out to be a reasonable choice, as in our experiments character-level NMT systems of smaller or comparable size were able to outperform word- and subword-based systems, which corroborates results of [12, 13]. Given this finding and the unique advantages of character-level processing (no pipelining, no tokenization, no additional hyperparameters, tiny vocabulary and memory, and robustness to spelling errors [14]), we hope that character-level NMT, and in general character-level sequence-to-sequence learning, will receive more attention from researchers.

Note that, although our character-based models outperform (sub)word-based ones with similar architectures on some datasets, we are not seeking to establish a new state-of-the-art in NMT with our model. Our goal is to isolate the effects of segmentation on quality by introducing a flexibility-enhancing research tool. Therefore, in the comparisons between (sub)word- and character-based models we purposely avoided introducing changes to our baseline RNN NMT architecture beyond upgrading the embedding layer.

2. Related Work

To tackle the OOV problem in word-level models, [15] proposed a hybrid model that composes unknown words from characters both on encoder and decoder side. While their approach relies on given word boundaries, they report a purely character-based baseline performing as well as a word-based model with unknown word replacement, but taking 3 months to train, which seems to have cooled off the NMT community in investigating fully character-based models as an alternative to (sub)word-based ones. Unlike [15], we found that despite the training speed being slower than for (sub)word vocabularies, it is possible to train reasonable character-level models within a few weeks. To combine the best of both worlds, [16] proposed hierarchical en-/decoders that receive inputs on both word- and character-level. The encoder learns a weighted recurrent representation of each word's characters and the decoder receives the previous target word and predicts characters until a delimiter is produced. Similar to our work, they find improvements over BPE models. The idea to learn composite representations of blocks of characters is similar to ours, but their approach requires given word boundaries, which our model learns on-the-fly. [12] combined a standard subword-level encoder with a two-layer, hierarchical character-level decoder. The decoder has gating units that regulate the influence of the lower-level layer to the higher-level one, hence fulfilling a similar purpose as our halting unit. This model outperforms a subword-level NMT system, and achieves state-of-the-art on a subset of WMT evaluation tasks. While not requiring explicit segmentation on the target side, the model still relies on given source segmentations. Finally, [14] proposed a fully character-level NMT model. They mainly address training speed, which [15] identified as a problem, and introduce a low-level convolutional layer over character embeddings to extract information from variable-length character n-grams for higher-level processing with standard RNN layers. Thus, overlapping segments are modelled with a length depending on the filters.

Perhaps closest to our work is [13], where each layer of a hierarchical RNN encoder is updated at different rates, with the first layer modelling character-level structures, the following modelling sub(word)-level structures. They introduce a binary boundary detector, similar to our halting unit, that triggers feeding of a representation to the next level, so that latent hierarchical structures without explicit boundary information are learnt. Unlike our fully-differentiable model, such discrete decisions of the boundary detector prohibit end-to-end differentiability, forcing a recourse to the biased straight-through estimator [17]. On the other hand, while our model relies on a to-be-tuned computation time penalty, [13] do not impose constraints on the number of boundaries.

3. Jointly Learning to Segment and Translate

Instead of committing to a single segmentation before NMT model training, we propose to learn the segmentation-governing parameters along with the usual network parameters in a end-to-end differentiable manner. With this approach, we get rid of pipelining and pre-/postprocessing, and can adaptively segment arbitrary inputs we encounter during training or testing. Our segmentations are context-dependent, i.e. the same substring can be segmented into different parts in different contexts. Being able to smoothly interpolate between word-based and character-based models we allow the model to find a sweet spot in between.

We extend the *Adaptive Computation Time* (ACT) paradigm [9], where a general RNN model is augmented with a scalar halting unit that decides how many recurrent computations are spent on each input. For segmentation, we use the halting unit to decide how many inputs (characters) a segment consists of. The output of the ACT module can thus be thought of as an 'embedding' vector for a segment that replaces the classic lookup embedding for (sub)words in standard NMT models. While our model can in principle use larger units as elementary inputs, we will focus on character inputs to be able to model the composition of arbitrary segments. That means that we only add a small amount of parameters to a basic character-based model, but explicitly model higher-level merges of characters into subwords.

3.1. ACT for Dynamic Depth

Here we summarize the ACT model [9]. It is applicable to any recurrent architecture that transforms an input sequence $\mathbf{x} = (x_1, \ldots, x_T)$ into outputs $\mathbf{o} = (\bar{o}_1, \ldots, \bar{o}_T)$ via computing a sequence of states $\mathbf{s} = (s_1, \ldots, s_T)$ through a state transition function $\mathcal{S}$ on an embedded input Ex_t and a linear output projection defined by matrix W_o and bias b_o:

$$s_t = \mathcal{S}(s_{t-1}, Ex_t), \qquad o_t = W_o s_t + b_o \tag{1}$$

Instead of stacking multiple RNN layers in $\mathcal{S}$ to achieve increased complexity of an RNN network, the ACT model

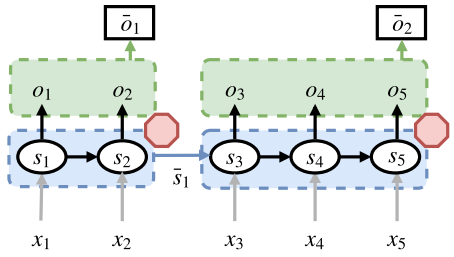


Figure 1: Diagram of the ACT-ENC encoder. Note the differences to the original ACT model: An input is here read on *every* internal recurrent iteration (gray arrows) and the halting unit (red stop sign) is repurposed to trigger feeding of an encoded embedding vector of a block of characters to the upstream NMT layers.

dynamically decides on the number of necessary recurrent steps (layers) for every input x_t. This saves computation on easy inputs, while still being able to use all of the processing power on hard inputs before emitting outputs. Concretely, an ACT cell performs an arbitrary number of internal recurrent applications of $\mathcal{S}$ for each input x_t:

$$s_t^n = \begin{cases} \mathcal{S}(\bar{s}_{t-1}, Ex_t), \text{ if } n = 1 \\ \mathcal{S}(s_t^{n-1}, Ex_t), \text{ otherwise} \end{cases} \quad (2)$$

The total number of internal steps is $N(t) = \min\{n' : \sum_{n=1}^{n'} h_t^n \geq 1 - \epsilon\}$, where $\epsilon \ll 1$ and h_t^n is the scalar output of sigmoid halting unit,

$$h_t^n = \sigma(W_h s_t^n + b_h). \quad (3)$$

Once halted, the final output $\bar{o}_t$ and state $\bar{s}_t$ (which is fed to the next ACT step in (2)) are computed as weighted means of intermediate outputs and states:

$$\bar{s}_t = \sum_{n=1}^{N(t)} p_t^n s_t^n, \qquad \bar{o}_t = \sum_{n=1}^{N(t)} p_t^n o_t^n \quad (4)$$

where probabilities p_t^n are defined as

$$p_t^n = \begin{cases} R(t), \text{ if } n = N(t) \\ h_t^n, \text{ otherwise} \end{cases} \quad (5)$$

and remainders $R(t) = 1 - \sum_{n'=1}^{N(t)-1} h_t^{n'}$. Finally, to prevent the network from pondering on an input for too long, the remainder $R(t)$ is added as a penalty to the RNN training loss (usually cross-entropy (XENT)) with a weight τ:

$$L_{\text{ACT}} = L_{\text{XENT}} + \tau R(t). \quad (6)$$

Thanks to (4), the model is deterministic and differentiable.

3.2. ACT for Dynamic Segmentation

We now describe how to use the ACT paradigm to enhance an encoder for dynamic segmentation on the source side (ACT-ENC). We reuse the idea of halting units, mean field updates and τ-penalized training objective, but instead of learning how much computation is needed for each atomic input, we learn how much computation to allow for an aggregation of atomic inputs, i.e. one segment.

The input to an ACT-ENC cell is a sequence of one-hot-encoded characters $\mathbf{x} = (x_1, \ldots, x_{T_\mathbf{x}})$. The ACT-ENC, depicted in Figure 1, receives one input x_t at a time and decides whether to halt or not. In the case of no halting, the cell proceeds reading more inputs; if it halts, it produces an output 'embedding' $\bar{o}$ of a block of characters read so far, and the cell resets for reading the next block. The sequence of the output embeddings $\mathbf{o} = (\bar{o}_1, \ldots, \bar{o}_{T_\mathbf{o}})$ is then fed to upstream standard (possibly bidirectional) NMT encoder layers, replacing the usual, one-hot encoded, (sub)word lookup embeddings. The length of $\mathbf{o}$ is variable: The more frequently ACT-ENC halts, the more embeddings are generated. In extreme cases, it can generate one embedding per input ($T_\mathbf{o} = T_\mathbf{x}$) or just one embedding for the full sequence of inputs ($T_\mathbf{o} = 1$).

Algorithm 1 ACT-ENC

Input: Weights W_o, b_o, W_h, b_h, transition function $\mathcal{S}$, embeddings E_{src}, inputs $\mathbf{x} = (x_1, \ldots, x_{T_\mathbf{x}})$, threshold ϵ.
Output: Outputs $\mathbf{o} = (\bar{o}_1, \ldots, \bar{o}_{T_\mathbf{o}})$, remainder R.

1: $\mathbf{o} = [\,]$ ▷ empty sequence
2: $R = 0, H = 0$ ▷ init remainder and halting sum
3: $\bar{s} = \mathbf{0}, \bar{o} = \mathbf{0}, s_0 = \mathbf{0}$ ▷ init mean state and output
4: **for** $t = 1 \ldots T_\mathbf{x}$ **do** ▷ loop over inputs
5: $s_t = \mathcal{S}(s_{t-1}, E_{src}\, x_t)$ ▷ new state
6: $o_t = W_o s_t + b_o$ ▷ new output
7: $h_t = \sigma(W_h s_t + b_h)$ ▷ halting score
8: $f = [\![H + h_t \geq 1 - \epsilon]\!]$ ▷ halting flag
9: $p_t = (1 - f)\, h_t + f\, (1 - H)$ ▷ halting probability
10: $H = H + h_t$ ▷ update halting sum
11: $\bar{s} = \bar{s} + p_t s_t, \bar{o} = \bar{o} + p_t o_t$ ▷ mean state and output
12: $R = R + (1 - f)\, h_t$ ▷ increment remainder
13: **if** f **then**
14: $\mathbf{o} = \mathbf{o}^\frown[\bar{o}]$ ▷ append output
15: $s_t = \bar{s}$ ▷ overwrite for next step
16: $\bar{s} = \mathbf{0}, \bar{o} = \mathbf{0}, H = 0$
17: $R = (1 - R)/t$ ▷ normalize remainder

In more detail, ACT-ENC implements the pseudocode given in Algorithm 1. Let $\mathcal{S}(s_{t-1}, i_t)$ be any recursive computation function (in this work we use GRUs) of an RNN that receives a hidden state s_{t-1} and an input vector i_t at time step t and computes the new hidden state s_t. In line 5 this function is computed on the regular previous state or, if there was a halt in the previous step (line 13), on the mean state vector $\bar{s}$ that summarizes the states of the previous segment (line 15, cf. (4), 1st eq.). Per-step outputs o_t are computed from the hidden states s_t with a feed-forward layer (line 6, cf. (1), 2nd eq.). A sigmoid halting unit computes a halting score in each step (line 7, cf. (3)). The halting probability for step t is either the halting score h_t or the current value of remainder $1 - H$ to ensure that all halting probabilities within one segment form a distribution (line 9, cf. (5)). ϵ is set to a small number to allow halting after a single step.

Whenever the model decides to halt, an output embedding $\bar{o}$ is computed as a weighted mean of the intermediate outputs of the current segment (line 14, cf. (4), 2nd eq.). The weighted mean on the one hand serves the purpose of circumventing stochastic sampling, on the other hand can be interpreted as a type of intra-attention summarizing the intermediate states and outputs of the segment. The halting scores from each step are accumulated (line 12) to penalize computation time as in (6). The hyperparameter τ here controls the segment length: The higher its value, the more preference will be given to smaller remainders, i.e. shorter segments. We introduce an additional normalization by input length (line 17), such that longer sequences will be allowed more segments than shorter sequences. This implementation exploits the fact that ACT-ENC outputs are weighted means over time steps and updates them incrementally. The algorithm allows efficient minibatch processing by maintaining a halting counter that indicates which embedding each current intermediate output in the batch contributes to. Incremental updates of embeddings and states are achieved with masks depending on the halting position.

4. Experiments

We reimplemented the Groundhog RNN encoder-decoder model with attention [18] in `MxNet Gluon` to allow for dynamic computation graphs. We report results on four language directions and domains, for word-, subword-, character-level and ACT-ENC segmentation: German-to-English TED talks, Chinese-to-English web pages, Japanese-to-English scientific abstracts and French-to-English news. Table 1 gives a data overview.

The IWSLT data is split and processed as in [19]; since it comes pretokenized and lowercased, models are evaluated with tokenized, lowercased BLEU (using `sacrebleu` [20]) and chrF scores on character bigrams [21]. For WMT, we used the 2014 dataset prepared for [18], additionally filtering the training data to include only sequences of a lengths 1 to 60, and models are evaluated with cased BLEU and chrF (`sacrebleu`, with the "13a" tokenizer).

The CASIA and ASPEC data are, respectively, from the 2015 China Workshop on MT (CWMT), used without preprocessing and with sampled dev/test sets, and from the WAT 2017 SmallNMT shared task, pretokenized with WP. Both datasets have BPE and WP vocabularies of around 16k for each side, and we report cased BLEU and chrF on them.

Hyperparameters. All models are trained with Adam [22] and a learning rate of 0.0003, halved whenever the validation score (tokenized BLEU) has not increased for 3 validations. Training stopped when the learning rate has been decreased 10 times in a row. All models use recurrent cells of size 1,000 for the decoder, with a bidirectional encoder of size 500 for each direction, input and output embedding of size 620, and the attention MLP of size 1,000, all following [18]. When multiple encoders layers are used, they are all bidirectional [23] with attention on the uppermost layer. The ACT layer for ACT-ENC models has size 50 for IWSLT, CASIA and ASPEC, and 25 for WMT (picked from $\{25, 50, 75, 100, 150\}$). The word-based models on IWSLT and WMT have a vocabulary of 30k for each side, the BPE models have separate 15k vocabularies for IWSLT and a joint 32k vocabulary for WMT. For IWSLT, CASIA and ASPEC all characters from the training data were included in the vocabularies, resulting in vocabulary sizes of 117 (de) and 97 (en), 7,284 (zh) and 166 (en), and 3,212 (ja) and 233 (en), respectively. For WMT the vocabularies included the 400 most frequent characters on each side. Word- and BPE-based models are trained with minibatches of size 80, character-based models with 40. The maximum sequence length during training is 60 for word- and BPE-based models, 200 for character-based models and 150 for ACT-ENC, to fit into available memory. $\tau = 1.0$ delivered the highest BLEU score for IWSLT and CASIA, $\tau = 0.8$ for WMT and $\tau = 0.7$ for ASPEC. Following [9], we fixed $\epsilon = 0.01$ in all the experiments. During inference, we use beam-search with a beam size of 5 and length-normalization.

Data	Domain	Lang	Train	Dev	Test
IWSLT	TED talks	de-en	153,352	6,970	6,750
CASIA	web	zh-en	1,045,000	2,500	2,500
ASPEC	sci. abstracts	ja-en	2,000,000	1,790	1,812
WMT	news	fr-en	12,075,604	6,003	3,003

Table 1: Data statistics (number of parallel sentences).

Data	Model	BLEU	chrF	Param	SegLen	TrainTime
IWSLT de-en	Word	22.11	0.44	80.5M	4.66	23h
	BPE	25.38	0.49	46.5M	4.09	20h
	Char	22.63	0.46	13.4M	1.00	1d22h
	ACT-ENC	22.67	0.46	13.5M	1.88	9d21h
CASIA zh-en	BPE	10.59	0.37	49.9M	1.72	18h
	Char	12.60	0.40	21.0M	1.00	10d6h
	ACT-ENC	9.87	0.36	21.3M	1.006	3d13h
ASPEC ja-en	WP	21.05	0.53	50.0M	2.07	4d4h
	Char	22.75	0.55	15.6M	1.00	24d15h
	ACT-ENC	15.82	0.46	15.6M	1.0007	15d4h
WMT fr-en	Word	20.32	0.49	80.5M	5.19	4d9h
	BPE	27.02	0.55	86.0M	4.05	3d23h
	Char	24.25	0.53	14.1M	1.00	9d
	ACT-ENC	13.74	0.42	14.2M	1.82	13d8h

Table 2: Results on test sets for 1-layer models, and number of parameters and average source segment lengths on dev sets. Time to reach stopping criterion is in (d)ays and (h)ours.

Evaluation Results. Table 2 lists the results for the most comparable, 1-layer, configuration. BPE/WP models expectedly outperform word-based models, however word-based models are also outperformed by character-based models. The picture is similar w.r.t. the chrF with even smaller relative differences. The ACT-ENC model with one unidirectional ACT layer manages to match the 1-layer bidirectional character-based model on IWSLT. But it does not reach the results of other models on CASIA and ASPEC, which can be explained by increased complexity of doing simultaneous segmentation during training on sentences longer than the average sentence length in IWSLT. However, the main

Data	Model	BLEU	chrF	Param	SegLen	TrainTime
IWSLT de-en	Word, 4L	24.54	0.45	97.0M	4.66	1d8h
	BPE, 1L	25.38	0.49	46.5M	4.09	20h
	Char, 5L	**28.19**	**0.51**	26.9M	1.00	3d10h
	ACT-ENC, 3L	25.10	0.49	25.6M	1.31	9d7h
CASIA zh-en	BPE, 3L	11.01	0.38	58.9M	1.72	24h
	Char, 3L	**13.43**	**0.42**	30.0M	1.00	5d6h
	ACT-ENC, 2L	10.35	0.37	21.3M	1.00	10d
ASPEC ja-en	WP, 3L	22.02	**0.55**	61.4M	2.07	4d2h
	Char, 1L	**22.75**	**0.55**	15.6M	1.00	24d15h
	ACT-ENC, 1L	15.82	0.46	15.6M	1.0007	15d4h
WMT fr-en	Word, 2L	21.04	0.48	94.0M	5.19	4d16h
	BPE, 3L	**27.93**	**0.56**	98.0M	4.05	5d3h
	Char, 6L	27.23	0.55	27.6M	1.00	18d13h
	ACT-ENC, 2L	14.01	0.43	21.7M	1.0001	9d10h

Table 3: Results on respective test sets after tuning the number of encoder (L)ayers (from 1 to 6) on the dev set.

finding here is that ACT-ENC recovers an almost character-level segmentation ("SegLen" column in Table 2). On the IWSLT dev set, the average segment length is only 1.88, with a maximum of 5 characters per segment. For CASIA and ASPEC domains, and with the larger datasets than IWSLT, the ACT-ENC segmentations becomes more fine-grained: The average segment length is, respectively, just 1.006 and 1.0007 on the dev set (max. 2 chars per segment). Given that the character model outperforms the BPE/WP models, it is not surprising that ACT-ENC converged to the character segmentation. We hypothesize that ACT-ENC could not improve over the 1-layer bidirectional character model because of complexity of identifying segments in Chinese and Japanese, unidirectionality of its initial layer, and increased hardness of optimization of character-based models with extra non-linearities [24], that causes earlier convergence to poorer minima in many runs. Similarly for WMT, failing to match the performance of the character model could be caused by harder optimization task on particularly long sentences in the WMT data, and unidirectionality of ACT-ENC. The ACT-ENC's segment length is 1.82 (max. 6 chars), again close on average to a purely character segmentation.

Inspired by the ACT-ENC's recovery of almost character segmentation and by the competitive performance of pure character-based models, we decided to verify if the advantage of character-level processing carries over to multiple layers. Since the character models are much smaller than their word-/BPE-based counterparts, one should allow multiple layers (consuming the same or less memory) to make up for the difference in number of parameters for fairer comparison. This also aimed to verify whether an increased number of non-linearities (one of ACT's benefits [25]) plays a role.

Table 3 shows the test results after tuning the number of bidirectional encoder layers, from 1 to 6, on dev sets. First, we observe the modest parameter number of character models even with multiple layers, that allows them to take advantage of deeper cascades of non-linearities while staying well below the memory budget of (sub)word-based 1-layer models. Second, comparing to Table 2, we again confirm the negative correlation of quality and segment length for ACT-ENC. Finally, we discover that BPE/WP models are outperformed by character-based models with multiple encoder layers, achieving gains of 2.8 BLEU points on IWSLT, 0.7 on ASPEC, and losing only 0.7 on WMT (with a minor decrease in chrF), despite having at least 3.5 times fewer parameters. Such ranking of character- and BPE-based models on WMT might be explained by much longer sentences in the corpus, compared other corpora, since the ability of character and ACT-based models to cover unseen input is limited by the maximum training sequence length limit (here 200 characters), which on WMT data crops 30.5% of sentences.

Analysis of Segmentation and Outputs. Randomly selected translation examples from the IWSLT dev set and their segmented sources are given in Table 4. In general, when encountering rare inputs, word-based models fail by producing the unknown word token (<unk>), and the BPE-model is able to translate only a more common part of German compounds (e.g. 'tiere' → 'animals'). The character-based models invent words ('altients', 'jes lag') that are similar to strings that they saw during training and the source. In a few cases they fallback to a language-modeling regime having attended to the first characters of a corresponding source word: e.g., instead of translating 'reisen' to 'journeys', the ACT-ENC model translates it to 'rows' (confusing 'reisen' to a similarly spelled German 'reihen'), or 'layering' instead of 'shift work' (confusing 'schichten' to the prefix-sharing 'schichtarbeit'). This is confirmed when inspecting attention scores: The model frequently attends to the correct source word, but mainly to the first characters only. Note that ACT-ENC segmentations are context-dependent, e.g. occurrences of 'tiere' are segmented differently.

Table 5 lists the most frequent segments produced by 1-layer ACT-ENC. For IWSLT, we observe that many segments make sense statistically (frequent or rare patterns) and linguistically to some extent: Many of the frequent segments include whitespace (itself a frequent symbol); 2-gram segments amongst others include frequent word suffixes ('en', 'in', 'er'), but also frequent diphthongs ('ei' and 'ie'); 3-grams start with rare characters like 'x' and 'y' or single dashes; 4-grams combine single characters with whitespaces and double dashes; 5-grams cover numbers, in particular, years. Importantly, though, since the best test BLEU scores for IWSLT were obtained by a multi-layer character-based model, the ACT-ENC model has done a reasonable job in improving over the already well-performant strategy, one character per segment, despite having only a single NMT layer. For CASIA and ASPEC, ACT-ENC converged to a segmentation even closer to pure characters: for CASIA, the most frequent 2-grams are punctuation combined with frequent pronoun 他 or preposition 的, or with the hieroglyph 明 from a common phrase '[smth.] shows, [that]' (all 4-10k in train), and parts of rare English words; for ASPEC, it is mostly the Hiragana letter き that starts the segments. While this letter also occurs as singleton (183× in the dev set, vs. 52× as part of a learned segment), and is frequent in the training set (239k), it is not the most frequent letter. For WMT, charac-

Ref	in social groups of animals , the juveniles always look different than the adults .
Word	in groups of social animals , the children are always different from the other than the <unk>.
BPE	in gruppen sozialer tiere sehen die jung@@ tiere immer anders aus als die alt@@ tiere . in groups , in groups , the juveniles are seeing the same animals as well as the animals .
ACT-ENC	in| g|ru|pp|en| s|oz|ia|le|r |ti|er|e |se|he|n |d|ie| j|un|gt|ie|re| i|m|m|er| a|nd|er|s |au|s |al|s |d|ie| a|lt|ti|er|e |.| in groups , the juvenile seems to see the different approach than the algaes .
Char	in groups of social animals , the juveniles are still in the alite of the altients .
Ref	we 're living in a culture of jet lag , global travel , 24-hour business , shift work .
Word	we live in a civilization with <unk> , global travel , <unk> and <unk> .
BPE	wir leben in einer zivilisation mit jet@@ -@@ lag , weltweiten reisen , non@@ sto@@ p-@@ business und sch@@ icht@@ arbeit . we live in a civilization with a single , a variety of global travel , presidential labor and checking .
ACT-ENC	w|ir| l|eb|en| i|n |ei|ne|r |z|iv|il|is|at|io|n |m|it| j|et|-la|g |,| w|el|tw|ei|te|n |re|is|en| ,| n|on|st|op|-bu|si|ne|ss| u|nd| s|ch|ic|ht|ar|be|it| .| we live in a civilization with jes lag , worldwide rows , nonstop business and failing .
Char	we live in a civilization with jet walk , global journeys , nonstop-business and layering

Table 4: Greedy translations from the IWSLT dev set. Explicit segmentations are given for the ACT-ENC and BPE models.

Data	Len	Segments
IWSLT	2	en; n␣; er; ␣d; ie; e␣; ei; in; ␣s; ␣w …
	3	yst; -␣d; xtr; -␣u; 100; xpe; -␣w; xis; -␣e; -ge …
	4	–␣d; –␣w; –␣s; –␣i; –␣e; –␣u; –␣g; –␣m; –␣a; –␣k …
	5	1965␣; 969␣; 1987␣; 1938␣; 1621␣; 1994␣; 1985␣ …
CASIA	2	”。; ”，; er; ”他; --; ”的; le; 明，; li; ut; …
ASPEC	2	きる; きた; きな; きに; りん; きは; き，; きて …
WMT	2	e␣; s␣; ␣d; t␣; ␣l; es; on; ␣a; de; en …
	3	übe; Rüc; rüb; öve; ürs; Köp; üsl
	4	ümov; ölln; rüng; Jürg; ülle; Müsl Müni; üric; üdig; …

Table 5: Most frequent ACT-ENC segments.

ter 2-grams are all very frequent in the training data (8-11M occurrences) while longer segments are very rare (max. 1k occurrences). Longer segments all include umlauts (ü, ö), which are atypical for French and should be treated as one unit semantically since they are loan words or proper names.

Gating Behavior of Char-GRUs. To investigate the reasons for success of the deep character-based encoders and their better or on-par performance with the segmenting ACT-ENC model, we analyzed average activations of GRU gates. A GRU cell computes the next state as: $s_t = z \odot \tanh(x_t W_h + (h_{t-1} \odot r) W_g) + (1 - z) \odot s_{t-1}$, where z is the update gate and r the reset gate, both being outputs of sigmoid layers receiving x_t and h_{t-1} [26]. Taking a closer look at the average values of these gates, we find patterns of segmentation as depicted in Figure 2 for a 5-layer character model. Most of the time, a whitespace character triggers a visible change of gate behavior: Forward reset gates close (reset) one character after a whitespace and backward reset gates close at whitespaces and then both open at the subsequent character. The update gates show similar regularities, but here the average gate values are less extreme. For longer words all gate activations progressively decay with the length. In addition, the block-wise processing of the compound ‘schreibtisch’ (German: ‘writing table’) that was correctly split into ‘schreib’ and ‘tisch’, points to decompounding abilities that pure character-level models possesses beyond simple whitespace tokenization.

Overall, this illustrates that the recurrent gates equip pure

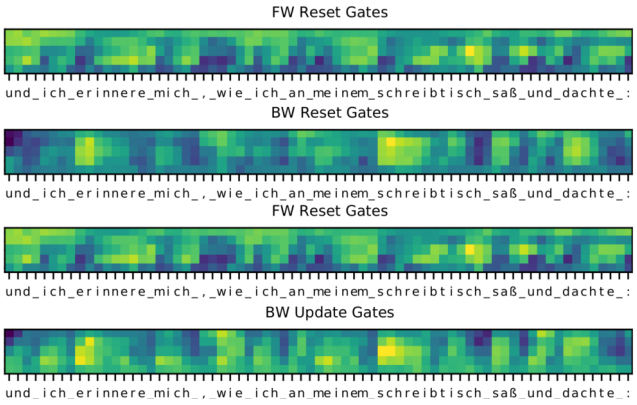


Figure 2: Mean activations for reset and update forward (FW) and backward (BW) GRU gates for an IWSLT sentence as produced by the 5-layer char model. Layers are stacked from bottom to top. Blue: values $\simeq 0$, yellow: values $\simeq 1$.

character models with the capacity to implicitly model input segmentations, which would explain why ACT-ENC could not find a radically different or advantageous segmentation.

5. Summary & Conclusion

We proposed an approach to learning (dynamic and adaptive) input segmentation for NMT based on the Adaptive Computation Time paradigm [9]. Experiments on four translations tasks showed that our model prefers to operate closely to the character level. This is echoed by the quantitative success of pure character-level models (without dynamic segmentation) and a qualitative analysis of gating mechanisms, suggesting that our adaptive model rediscovers the segmenting capacity already present in gated recurrent, pure character-based models. Given this and the absence of many development hurdles with character-based models, their lower memory consumption and higher robustness, the presented dynamic segmentation capacity, being primarily a diagnostic research tool, does not seem to be necessary to be modelled explicitly. We hope these insights can serve as justification for intensification of research in pure character-level NMT models.

6. References

[1] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, *et al.*, "Findings of the 2016 conference on machine translation," in *WMT*, 2016.

[2] ——, "Findings of the 2017 conference on machine translation," in *WMT*, 2017.

[3] P. Gage, "A new algorithm for data compression," *The C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.

[4] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *ACL*, 2016.

[5] M. Schuster and K. Nakajima, "Japanese and Korean voice search," in *ICASSP*, 2012.

[6] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, *et al.*, "Hidden technical debt in machine learning systems," in *NIPS*, 2015.

[7] E. Hoffer, I. Hubara, and D. Soudry, "Train longer, generalize better: closing the generalization gap in large batch training of neural networks," in *NIPS*, 2017.

[8] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A field guide to dynamical recurrent neural networks*. IEEE Press, 2001.

[9] A. Graves, "Adaptive computation time for recurrent neural networks," in *arXiv:1603.08983*, 2016.

[10] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *ICML*, 2017.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, *et al.*, "Attention is all you need," in *NIPS*, 2017.

[12] J. Chung, K. Cho, and Y. Bengio, "A character-level decoder without explicit segmentation for neural machine translation," in *ACL*, 2016.

[13] J. Chung, S. Ahn, and Y. Bengio, "Hierarchical multiscale recurrent neural networks," *ICLR*, 2017.

[14] J. Lee, K. Cho, and T. Hofmann, "Fully character-level neural machine translation without explicit segmentation," *TACL*, vol. 5, pp. 365–378, 2017.

[15] M.-T. Luong and C. D. Manning, "Achieving open vocabulary neural machine translation with hybrid word-character models," in *ACL*, 2016.

[16] S. Zhao and Z. Zhang, "Deep character-level neural machine translation by learning morphology," in *arXiv:1608.04738*, 2016.

[17] Y. Bengio, N. Léonard, and A. C. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," in *arXiv:1308.3432*, 2013.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[19] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, *et al.*, "An actor-critic algorithm for sequence prediction," in *ICLR*, 2017.

[20] M. Post, "A call for clarity in reporting BLEU scores," in *arXiv:1804.08771*, 2018.

[21] M. Popovic, "chrF: character n-gram F-score for automatic MT evaluation," in *WMT*, 2015.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[23] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, *et al.*, "The best of both worlds: Combining recent advances in neural machine translation," in *arXiv:1804.09849*, 2018.

[24] W. Ling, I. Trancoso, C. Dyer, and A. W. Black, "Character-based neural machine translation," in *arXiv:1511.04586*, 2015.

[25] D. Fojo, V. Campos, and X. Giró-i Nieto, "Comparing fixed and adaptive computation time for recurrent neural networks," in *Workshop Track of ICLR*, 2018.

[26] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, *et al.*, "Learning phrase representations using RNN encoder–decoder for Statistical Machine Translation," in *EMNLP*, 2014.

Data Selection with Feature Decay Algorithms Using an Approximated Target Side

Alberto Poncelas, Gideon Maillette de Buy Wenniger, Andy Way
ADAPT Centre, School of Computing,
Dublin City University, Dublin, Ireland
{firstname.lastname}@adaptcentre.ie

Abstract

Data selection techniques applied to neural machine translation (NMT) aim to increase the performance of a model by retrieving a subset of sentences for use as training data.

One of the possible data selection techniques are transductive learning methods, which select the data based on the test set, i.e. the document to be translated. A limitation of these methods to date is that using the source-side test set does not by itself guarantee that sentences are selected with correct translations, or translations that are suitable given the test-set domain. Some corpora, such as subtitle corpora, may contain parallel sentences with inaccurate translations caused by localization or length restrictions.

In order to try to fix this problem, in this paper we propose to use an approximated target-side in addition to the source-side when selecting suitable sentence-pairs for training a model. This approximated target-side is built by pre-translating the source-side.

In this work, we explore the performance of this general idea for one specific data selection approach called Feature Decay Algorithms (FDA).

We train German-English NMT models on data selected by using the test set (source), the approximated target side, and a mixture of both. Our findings reveal that models built using a combination of outputs of FDA (using the test set and an approximated target side) perform better than those solely using the test set. We obtain a statistically significant improvement of more than 1.5 BLEU points over a model trained with all data, and more than 0.5 BLEU points over a strong FDA baseline that uses source-side information only.

1. Introduction

Supervised machine learning aims to learn predictive models from a set of labeled examples (training data) so that it can accurately predict the labels of new, unlabeled, examples. Having more data may seem at first glance to be beneficial to building more accurate models, but upon closer inspection this is not necessarily always the case. Machine learning models by design have an inductive bias that forces them to generalize over the training examples rather than just memorizing them without generalization. This means, however, that if the size of the training set is increased, this may lead to optimizing the model for predicting the labels of more examples, but which on average are less relevant at test time than would be the case for a more focused, smaller training set. The intuition of the importance of using a highly relevant set of training examples is captured well by the K-nearest neighbour model, which essentially computes at test time on-the-fly a very localized density estimate for every test example, based on the K training examples closest to the test example. It then uses this density estimate for classification. For the K-nearest neighbour model, increasing K too much is at the expense of basing predictions on an increasing number of less relevant examples. Furthermore similar to the K-nearest neighbour model, other predictive models which typically discard the original training examples and keep only a learned generalization over these examples, can suffer if the training data becomes bigger but on average less relevant to the test set.

In Machine Translation (MT), the data used to build the models are parallel sentences (pairs of sentences in two languages, which are translations of each other) and we encounter the same problem when the amounts of data become excessively large. Too much training data may cause the model to be too generic, and not perform well if $test_{src}$ (the document to be translated, i.e. the test set), belongs to a specific domain.

Data selection techniques aim to solve that problem by selecting a subset of training data. Models that are trained on a small set of parallel sentences can perform better than those trained on all training data [1; 2].

Within the data selection field we can find several approaches to reduce the data: select sentences of good translation quality (*data quality*), select sentences relevant for a particular domain (*domain adaptation*), or select sentences that are relevant for $test_{src}$ (*transductive learning*). We focus on this last type, and so in this paper we propose new methods to build Neural Machine Translation (NMT) models that are tailored towards a $test_{src}$.

Transductive learning [3] aims to find the best training instances given an unlabeled example. In MT this means finding the best parallel sentences given a document $test_{src}$ to be translated. In our work, the transductive data-selection method that we explore is Feature Decay Algorithms (FDA) [4; 5; 6]. Standard FDA uses the *n*-grams of $test_{src}$ to retrieve training sentence pairs with source-side most similar to $test_{src}$. FDA has demonstrated good performance in Statistical Machine Translation (SMT) and NMT [2].

In most cases, FDA is used as a single step in the pipeline

of building a model, using $test_{src}$ to extract a subset of parallel sentences. In this paper, we propose a different configuration of use of FDA for building NMT models (see left side of Figure 1). In particular, we propose executing FDA not only using the $test_{src}$ (source-side language), as is common, but additionally on a pre-translated test set (approximated target-side). In order to avoid confusion, in this work we use $test_{src}$ to indicate the test set (in the source-side language) and $test_{trg}$ to indicate the pre-translation of the test set (in the target-side language). The outputs of these two executions can be combined into one training set to build a model that produces better translations than models built using FDA having only $test_{src}$ as input.

Considering both the source side and target side of the parallel sentences as selection criteria is especially useful when using a corpus that includes sentences from subtitles in different languages. There are particular problems concerning parallel sentences comprising subtitles. For example, both sentences in the source and target side are limited to be displayed in the same time window (assuming they are synchronized). As the length of the same sentences in different languages can be different, this may causes the longest one to be rephrased, split in two, or have words omitted so it meets the time requirement.

In our work we use an approximated, synthetic target-side using a technique we call pre-translation. One way to look at this is as a form of synthetic-data generation. As such it is somewhat reminiscent of synthetic source-data generation using a target-to-source translation model, a technique known as back-translation introduced by Sennrich et al.(2016) [7].

2. Related Work

Data selection techniques aim to select a subset of data such that the models trained on that subset perform better. There are multiple approaches to achieve those improvements, such as domain adaptation or noise reduction approaches [8].

Methods based on domain adaptation include the work of Moore and Lewis (2010) [9], who propose to use language models (LM) to select data. An LM is a distribution over sequences of words in a monolingual text, and is often used by SMT systems to model the fluency of the outputs. Given a string s and a language model LM_d, $H_d(s)$ is the entropy of the distribution of s according to LM_d.

Moore and Lewis build an in-domain language model LM_I and an out-of-domain language model LM_O, and determine how likely each sentence s is to be in-domain by computing the entropy difference $[H_I(s) - H_O(s)]$. Axelrod et al. (2011) [10] extend the method by using LMs in both the source-side and target-side languages, defining the bilingual cross-entropy difference.

Another method, proposed by van der Wees et al. (2017) [1], is to gradually remove out-of-domain sentences each η epochs when training the NMT model.

In our work, we select data that is similar to $test_{src}$ (and so, more relevant for use as training data). Previous research on selecting data considering the test set includes the work of Li et. al. (2018) [11] where they fine-tune a pre-built NMT model using training data selected based on $test_{src}$. They use similarity measures, such as Levenshtein distance [12] or the cosine similarity of the average of the word embeddings, [13].

The method that we use to select data is FDA [4; 5; 6], which has already proven to be useful in SMT [14; 15; 16] and NMT [2]. Selecting a small subset of sentences from a parallel corpus using FDA is enough to train SMT systems that perform better than systems trained using the whole parallel corpus.

FDA takes as input a set of parallel sentences U and a seed (generally the $test_{src}$). Given U and the seed, FDA retrieves an ordered sequence of sentences L from U. Sentences are ordered based on the amount of *n*-grams they share with the seed, with more shared *n*-grams meaning higher preference, while also considering the variability of the *n*-grams in the selected sentences.

The algorithm initializes L as a void sequence and iteratively selects one sentence $s \in U - L$ and appends it to L. The sentence s to select at each step is the one most relevant to $test_{src}$, based on the number of *n*-grams that s shares with the $test_{src}$. The score of the relevance is computed as in (1):

$$score(s) = \frac{\sum_{f \in F_s} 0.5^{C_L(f)}}{\text{\# words in s}} \quad (1)$$

where F_s is the set of *n*-grams present in s and $test_{src}$ (by default the order of the *n*-grams ranges from 1 to 3). $C_L(f)$ is the count of the *n*-gram f in the sequence L of selected sentences. Including $C_L(f)$ in the computation of the score causes the algorithm to penalize *n*-grams that have been selected several times, and hence favouring the selection of sentences that contain new *n*-grams.

3. Using an Approximated Test Target-side

FDA uses $test_{src}$ as seed to retrieve a subset from a set of parallel sentences. In order to retrieve the sentences it scores the *n*-grams of $test_{src}$ (source-side language). We show the pipeline of usage of FDA on the left side of Figure 1. Here, the files $test_{src}$ and *parallel text* are used as input, and FDA retrieves a subset of the sentences to be used for building a model that is adapted to $test_{src}$.

We propose to use both the test source-side $test_{src}$ and the approximated test target-side $test_{trg}$ as features in FDA, when selecting the set of sentences from the parallel text.

We show the pipeline of our approach on the right side of Figure 1. First, $test_{src}$ is translated (*translate* step). Then, using FDA, we select a subset of parallel sentences given: (a) $test_{src}$ as seed (FDA_{src}), and (b) $test_{trg}$ as seed (FDA_{trg}). These two sets can be combined into one set which serves as training data to build an MT model.

In the following subsections we explain in more detail two issues that are yet unanswered in the pipeline : (1) how

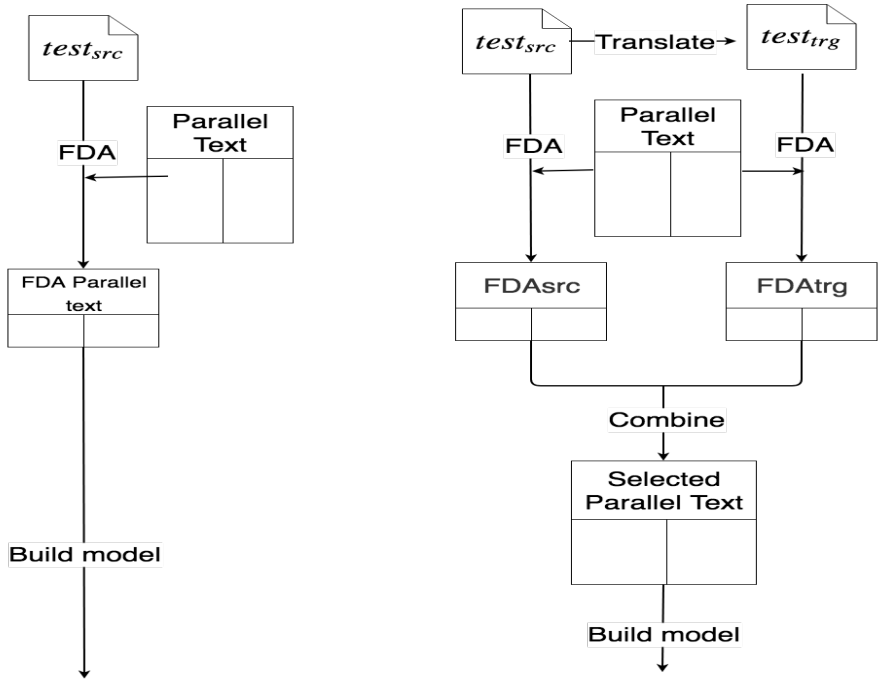


Figure 1: Pipeline of the traditional usage of FDA (left) and pipeline of our proposal, using the target-side (right).

to build $test_{trg}$ (addressed in Section 3.1), and (2) how to combine the outputs of FDA (addressed in Section 3.2).

3.1. Pre-Translation of $test_{src}$

The first step in our approach consists of building $test_{trg}$ (*translate* step on the right side of Figure 1) so it can be used as the seed to extract parallel sentences using the target side. In order to perform this pre-translation we need to build a model, which we refer to as the *initial model*.

There are several approaches to build the *initial model*, such as using SMT or NMT. These models can be trained using the full training data or subsets (such as randomly sampled, selected according to a particular domain, etc.). In this work we use an NMT model built with the full training data.

3.2. Combining FDA outputs

In order to combine the sentences of FDA_{src} and FDA_{trg} into one training set of N sentences, various strategies are possible such as retrieving the intersection or the union of sentences. In this work we explore the strategy of concatenating both outputs (allowing the repetition of sentences) and propose as future work alternative methods for merging both parallel datasets.

The outputs of FDA_{src} and FDA_{trg} can be seen as an ordered sequence of sentences as in equation (2) and equation (3):

$$FDA_{src} = (s_1^{(src)}, s_2^{(src)}, s_3^{(src)}, ...s_N^{(src)}) \quad (2)$$

$$FDA_{trg} = (s_1^{(trg)}, s_2^{(trg)}, s_3^{(trg)}, ...s_N^{(trg)}) \quad (3)$$

In order to obtain a training set that combines the outputs of FDA_{src} and FDA_{trg}, we concatenate the top sentences of each subset to obtain a new list of sentences of size N, as in equation (4)

$$FDA = (s_1^{(src)}, ...s_{N*\alpha}^{(src)}, s_1^{(trg)}, ...s_{N*(1-\alpha)}^{(trg)}) \quad (4)$$

where $0 \leq \alpha \leq 1$ indicates the proportion of sentences that are selected from FDA_{src} and FDA_{trg}.

Note that some of the sentences may be replicated; it may happen that $s_i^{(src)} = s_j^{(trg)}$, i.e. those that have been retrieved by both executions FDA. In this work we decided to keep the duplicates as it may be beneficial to oversample those sentences in which there is an agreement of both executions of FDA. However, we propose as future work to investigate the effect of removing those duplicate sentences.

The core of our approach is combining the outputs of the two executions of FDA (using the test and translated sets). Given the concatenation method presented in this section, the outputs can be classified as one of the three options:

- Source-side only: use only the output of FDA_{src} for building the model. It is the configuration where $\alpha = 1$ in Equation (4), which is equivalent to the traditional procedure of using FDA (left side of Figure 1, so we use this approach as the baseline.
- Target-side only: use the output of FDA_{trg} for building the model, which is the configuration where $\alpha = 0$ in Equation (4).

- Source-and-target-side: combine FDA_{src} and FDA_{trg}. This is the configuration where different values of α in equation (4) are set. In our work we explore the values $\alpha = 0.25$, $\alpha = 0.50$ and $\alpha = 0.75$.

4. Experiments

4.1. Experimental Settings

We experiment with models for German-to-English direction. The parallel data used for the experiments is the training data provided in the WMT 2015 [17] (4.5M sentence pairs, 225M words). The dev set of the NMT models (both the initial model and those trained using the selected datasets) are 5K randomly sampled sentences from development sets from previous years. All the models presented here are evaluated using the same test set which comprises documents provided in WMT 2015 translation task as $test_{src}$.

In order to build the NMT models we use OpenNMT-py, which is the Pytorch port of OpenNMT [18]. All the NMT models we build use the same settings (we only change the training data used to build them). The value parameters are the default ones of OpenNMT-py (i.e. 2-layer LSTM with 500 hidden units, vocabulary size of 50000 words for each language). All the models are executed for 13 Epochs.

In the experiments we build models with the data selected by using FDA_{src} and FDA_{trg}. We explore selecting different sizes of selected data: 500K, 1M and 2M sentence pairs.

5. Results

	baseline
BLEU	0.2474
TER	0.5525
METEOR	0.2798
CHRF3	48.9473

Table 1: Results of the model trained with all available training data; also the no-FDA baseline.

First, we show in Table 1 the quality of the pre-translated $test_{trg}$. This has been produced by the *initial model*, an NMT model trained with all training data. This result also serves as a no-FDA baseline to asses the benefit of using FDA in general with.

The evaluation metrics presented in Table 1 give an estimation of the similarity between the model output and a human-translated reference. The evaluation metrics we use are: BLEU [19], TER [20], METEOR [21] and CHRF [22].

The results of the models are shown in Table 2. The columns show the different configurations used to build the set of selected sentences (i.e. the value of α in equation (4) used). This means that the column $\alpha = 0.75$ shows the results of the model trained with the sentences from the top-750K sentences of FDA_{src} and the top-250K sentences of FDA_{trg}.

First, one may wonder whether FDA data selection is at all helpful? Comparing the scores in Table 2 to the baseline system trained on all data in Table 1, we see that all FDA systems outperform it, with the best one obtaining more than 1.5 BLEU points improvement (a relative improvement of 6%).

We have marked in bold the scores that outperform the second baseline: FDA applied using $test_{src}$ only (i.e the configuration using FDA_{src} and $\alpha = 1$), as proposed in [2], and computed the statistical significance (marked with an asterisk) with multeval [23] for BLEU, TER and METEOR when compared to the baseline at level p=0.01 using bootstrap resampling [24].

5.1. Ratio of data obtained using source and target side

Intuitively, models built using the data selected based on $test_{trg}$ might perform worse than using $test_{src}$ only. $test_{trg}$ may contain errors produced by the machine-generated text, so an algorithm that bases the decision on that text may not select the best sentences. Indeed, this can be seen in the column $\alpha = 0$ of Table 2, where most of the scores are worse than those in column $\alpha = 1$.

On the other hand, using only $test_{src}$ as a selection criterion also has its limitations. While it guarantees the selected source sentences to be similar to $test_{src}$, it does not provide any information about the target side of the selected sentences. Therefore, it may still select sentences with target-side translations that are wrong or not suitable given the domain of the test-set, thereby hurting the final translation accuracy.

Using training data containing parallel sentences that are not an accurate translation of each other is a problem that can be encountered when using parallel sentences obtained from subtitles. Often, translation of subtitles needs to be adapted to meet length requirements (due to the restriction of time it is displayed on screen). We present some examples of sentences that are not accurately translated in Table 3.

We find that selecting sentences based both on $test_{src}$ and on $test_{trg}$ works better than using one selection criterion alone. Thus, using an approximated target side, even if imperfect, can help. The best performance is obtained using configurations that combine outputs of FDA_{src} and FDA_{trg} ($\alpha = 0.75$, $\alpha = 0.50$ and $\alpha = 0.25$ columns).

The best results are obtained for $\alpha = 0.75$ using 1 million sentences for selection. This setting improves 1.53 BLEU points over the no-FDA baseline (model trained with all data) and 0.67 BLEU points over the baseline that uses only the source side for selection in FDA.

In Table 3 we show examples of sentences that are exclusive outputs of FDA_{src} or FDA_{trg}. These examples give an indication about how including the output of FDA_{trg} can benefit (or hurt) the quality of the selected data.

In the first row we see that the sentence "nun gibt es kein Zurück mehr ." has been selected by FDA_{src} as it matches

		$\alpha = 1$	$\alpha = 0.75$	$\alpha = 0.50$	$\alpha = 0.25$	$\alpha = 0$
500K lines	BLEU	0.2517	**0.2542**	**0.2543**	**0.2534**	0.2441
	TER	0.5601	**0.5521***	**0.5563**	**0.5544**	0.5628
	METEOR	0.2886	**0.2895**	0.2882	**0.2888**	0.2789
	CHRF3	49.8314	**50.0915**	**49.8898**	**49.9074**	48.7796
1M lines	BLEU	0.256	**0.2627***	**0.2595**	**0.2600***	0.2496
	TER	0.5497	**0.5455***	**0.5462**	**0.5493***	0.5534
	METEOR	0.2886	**0.2920***	**0.2921***	**0.2918***	0.2833
	CHRF3	50.0932	**50.6273**	**50.5226**	**50.5682**	49.5192
2M lines	BLEU	0.2585	**0.2610**	0.2580	**0.2614**	0.2547
	TER	0.5454	**0.5429**	0.5465	**0.5437**	0.5496
	METEOR	0.2894	**0.2923***	**0.2903**	**0.2927***	0.2852
	CHRF3	50.095	**50.5582**	**50.2431**	**50.5487**	49.7838

Table 2: Results of the models using different sizes of FDA_{src} and FDA_{trg}.

German	English	pos FDA_{src}	pos FDA_{trg}
nun gibt es kein Zurück mehr .	there is no going back now .	12	-
diese Zahl ist mehr als doppelt so viel , als vor 10 Jahren .	famous pieces from the 19th century include those by Delacroix , Gauguin , Monet , Renoir and Corot .	50	-
diese Aufzählung ließe sich beliebig fortführen .	and I could continue .	-	63
bitte beachten Sie , dass Sie sich registrieren lassen müssen , um einen Zugang zu den detaillierten Außenhandelsdaten zu erhalten .	all data can be downloaded free of charge .	-	92

Table 3: Examples of sentences retrieved by FDA_{src} and FDA_{trg}

"kein Zurück mehr" in the input. According to this sentence, this *n*-gram should be translated as "no going back". The translation found for "kein Zurück mehr" in $test_{trg}$ is "point where there is no return" (which, in addition, is closer to the reference "point of no return") and hence FDA_{trg} will use *n*-grams such as "point" or "no return" to retrieve sentences.

In the second row, we find an example of a sentence retrieved by FDA_{src} whose translation is not accurate (this is easily noticeable as the names "Delacroix, Gauguin, Monet, Renoir and Corot" are not present in the English-side sentence). Including this sentence in the training data causes the quality to decrease and the models to perform worse. This problem is not exclusive of FDA_{src}, as in rows 3 and 4 we see the same problem happening in the output of FDA_{trg}.

Combining the outputs of FDA_{src} and FDA_{trg} causes the training data to be reinforced with sentences with relevant translations. Note that mixing the outputs of the two executions of FDA cause some sentence pairs to be replicated, as there is an overlap of the outputs.

In Table 4 we indicate the amount of unique lines contained in the training data of the models (those presented in Table 2). In the table we observe that the number of unique lines is high in all training sets. The proportion of unique lines ranges from 82% to 94%, which shows how FDA_{src} and FDA_{trg} retrieve different sentences mostly.

	$\alpha = 0.75$	$\alpha = 0.50$	$\alpha = 0.25$
500K	471753 (94%)	460993 (92%)	471174 (94%)
1M	918506 (92%)	886685 (89%)	917087 (92%)
2M	1749015(87%)	1648727(82%)	1745142(87%)

Table 4: Number of unique sentences in the training data.

When performing a column-wise comparison in Table 4, we see how the number of unique lines is larger when the output of one of the FDA models dominates the training data ($\alpha = 0.25$ or $\alpha = 0.75$ columns) compared to those sets that contain the same amount of sentences extracted from FDA_{src} and FDA_{trg} (column $\alpha = 0.50$).

We also see that the larger the amount of selected data, the more overlap exists between the two outputs (the proportional amount of unique lines is smaller). For example, in column $\alpha = 0.50$, when 500K lines are selected, there are 92% non-repeated lines, and this decreases to 82% when selecting 2M lines. The same can be observed in the other columns. This indicates how the selected data using FDA_{src} and FDA_{trg} tend to be more similar the more sentences are retrieved.

6. Conclusion and Future Work

In this work, we explored a different pipeline in which FDA can be used. We discovered that using $test_{trg}$ (which is machine-generated) as the seed of FDA can improve the performance.

In our experiments, we built models using training sets containing replicated instances of sentence pairs (as the output of the two runs of FDA, on the source-side and target-side, may overlap). This opens the door to exploring data selection algorithms allowing the repetition of selected instances.

In the future, we want to consider other procedures for combining the outputs of FDA, as we believe that other merging strategies may achieve better results. For example, considering both *n*-grams on the source and target side in combination (rather than two separate executions of FDA) may achieve better performance.

In addition, we want to explore the performance when using a different *initial model*. Changing the initial model to produce the $test_{trg}$ causes FDA_{trg} to have a different performance. We believe that using another dataset to build the initial NMT model (or even using different paradigms such as SMT or rule-based MT) or choosing an initial model that is also closer to $test_{src}$ (e.g. using FDA to build the initial model) should boost the performance. Moreover, the use of several initial models allow us to perform concatenations of several outputs of FDA using different seeds.

Finally, we want to explore how data selection algorithms may improve when allowing the algorithm to select the same sentence pairs several times.

7. Acknowledgements

This research has been supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

This work has also received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 713567.

8. References

[1] M. van der Wees, A. Bisazza, and C. Monz, "Dynamic data selection for neural machine translation," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, 2017, pp. 1400–1410.

[2] A. Poncelas, G. M. de Buy Wenniger, and A. Way, "Feature decay algorithms for neural machine translation," in *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, Alacant, Spain, 2018, pp. 239–248.

[3] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[4] E. Biçici and D. Yuret, "Instance selection for machine translation using feature decay algorithms," in

Proceedings of the Sixth Workshop on Statistical Machine Translation, Edinburgh, Scotland, 2011, pp. 272–283.

[5] E. Biçici, Q. Liu, and A. Way, "ParFDA for fast deployment of accurate statistical machine translation systems, benchmarks, and statistics," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal, 2015, pp. 74–78.

[6] E. Biçici and D. Yuret, "Optimizing instance selection for statistical machine translation with feature decay algorithms," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 339–350, 2015.

[7] 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, 2016, pp. 86–96.

[8] S. Eetemadi, W. Lewis, K. Toutanova, and H. Radha, "Survey of data-selection methods in statistical machine translation," *Machine Translation*, vol. 29, no. 3-4, pp. 189–223, 2015.

[9] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *Proceedings of the ACL 2010 conference short papers*, Uppsala, Sweden, 2010, pp. 220–224.

[10] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, UK., 2011, pp. 355–362.

[11] X. Li, J. Zhang, and C. Zong, "One Sentence One Model for Neural Machine Translation," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018, pp. 910–917.

[12] V. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," in *Soviet Physics Doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[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] E. Biçici, "Feature decay algorithms for fast deployment of accurate statistical machine translation systems," in *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria, August 2013, pp. 78–84.

[15] A. Poncelas, A. Way, and A. Toral, "Extending feature decay algorithms using alignment entropy," in *International Workshop on Future and Emerging Trends in Language Technology*, Seville, Spain, 2016, pp. 170–182.

[16] A. Poncelas, G. M. de Buy Wenniger, and A. Way, "Applying n-gram alignment entropy to improve feature decay algorithms," *The Prague Bulletin of Mathematical Linguistics*, vol. 108, no. 1, pp. 245–256, 2017.

[17] 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*, Lisboa, Portugal, September 2015, pp. 1–46.

[18] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations*, Vancouver, Canada, 2017, pp. 67–72.

[19] 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, July 2002, pp. 311–318.

[20] 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.

[21] S. Banerjee and A. Lavie, "Meteor: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, Ann Arbor, Michigan, 2005, pp. 65–72.

[22] M. Popovic, "chrF: character n-gram F-score for automatic MT evaluation," in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, Lisbon, Portugal, 2015, pp. 392–395.

[23] J. H. Clark, C. Dyer, A. Lavie, and N. A. Smith, "Better hypothesis testing for statistical machine translation: Controlling for optimizer instability," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, Portland, Oregon, 2011, p. 176–181.

[24] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, 2004, pp. 388–395.

Multi-paraphrase Augmentation to Leverage Neural Caption Translation

Johanes Effendi[1], Sakriani Sakti[1,2], Katsuhito Sudoh[1,2], Satoshi Nakamura[1,2]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project AIP, Japan
{johanes.effendi.ix4,ssakti,sudoh,s-nakamura}@is.naist.jp

Abstract

Paraphrasing has been proven to improve translation quality in machine translation (MT) and has been widely studied alongside with the development of statistical MT (SMT). In this paper, we investigate and utilize neural paraphrasing to improve translation quality in neural MT (NMT), which has not yet been much explored. Our first contribution is to propose a new way of creating a multi-paraphrase corpus through visual description. After that, we also proposed to construct neural paraphrase models which initiate expert models and utilize them to leverage NMT. Here, we diffuse the image information by using image-based paraphrasing without using the image itself. Our proposed image-based multi-paraphrase augmentation strategies showed improvement against a vanilla NMT baseline.

1. Introduction

In general, sentence paraphrasing is a way to restate a concept with different vocabulary, style, and level of detail. As defined by De Beaugrande and Dressler, a paraphrase is an approximate conceptual equivalence among outwardly different material [1]. In many language generation tasks, paraphrasing plays a critical role for enrichment and adding flexibility. In the MT system, paraphrases are often used for multi-reference evaluation [1], pre-editing of source sentences [2, 3, 4] and automatic post-editing [5, 6, 7].

Moreover, since the development of SMT, there have been a lot of approaches for using paraphrasing to elaborate the source language data. Such method have been concluded as a convenient way to handle out-of-vocabulary (OOV) and rare words problem [8]. A study by Madnani and Dorr also showed that by using targeted paraphrases, unfair penalization of translation hypotheses could be avoided [9]. Paraphrasing could also be used to augment the dataset size, which correlates positively with translation result in SMT [4, 10].

However, despite a wide range of existing works of paraphrasing, MT studies usually use a strict definition of paraphrase which accepts only word substitution and reordering. The reason is that we cannot grasp a tangible concept about the idea of the sentence being translated. On the other hand, Hirst argues that paraphrases don't necessarily need to be fully synonymous. It is sufficient for them to be quasi-synonymous, as a mutually replaceable form of truth applicable in some contexts [11]. By taking further this idea, as long as the semantics of the mutual paraphrase sentence can be determined, we actually can widen the paraphrase definition to some extent.

In this research, we treat an image as a symbolic form of sentence idea, regarded as the basis of paraphrasing. We consider two sentences as paraphrase as long as both of them are talking about the same image. This means that the word or phrase insertion and deletion based on the same picture as a concept may now be accepted as one of the paraphrase variations. Slightly different from the usual use case, this definition can be called image-based paraphrasing. Furthermore, as paraphrasing to enable multi-source information in NMT is not much investigated yet, in this study we explore the use of image-based paraphrasing to leverage NMT quality.

Recently, the Second Conference on Machine Translation (WMT17) accelerated a "Multimodal Machine Translation" shared task that aimed to translate the image descriptions into the target language. Most approaches focus on utilizing image features in addition to the information from a single caption of the source language. However, the results from most submitted systems reveal that the additional image features could only slightly contribute to system performance. As pointed out by Calixto et al. [12] the image-text latent representation combination approach has not yielded significant improvement on WMT 2017 Multimodal shared task dataset testing. Here, we attempt to go in another direction in which we diffuse the image information by using image-based paraphrasing without using the image itself. The resulting paraphrase captions are then utilized within a multi-source and multi-expert NMT model.

In summary, the contributions of this work include:

1. Introduce a new way of creating a multi-paraphrase corpus through image captions so-called image-based paraphrasing.
2. Generate multi-paraphrase sentences of the WMT17 Multimodal Translation Task dataset through crowdsourcing, which can be used by the community[1]
3. Develop automatic paraphrase generation in a semi-supervised manner;

[1]The data will be soon available publicly.

4. Utilize multi-expert translation in neural machine translation using our proposed paraphrase; and
5. Improve the baseline used at WMT17 with a 13.2 BLEU score margin, which is close to the top score that used a multimodal model.

2. Multi-paraphrase Generation

2.1. Defining Paraphrase Elementary Operation

To train an NMT model with our image-based multi-paraphrases, firstly we need to build a set of paraphrased source sentences with images as the basis of paraphrasing. However, the process of manually collecting paraphrases is expensive and time-consuming. On the other hand, Resnik et al. (2013) proposed that corpus creation with a crowdsourcing platform provides such advantages as low cost, effectiveness, and reasonable quality [13].

Figure 1: Reference image for captioning and paraphrasing shown in Table 1.

Furthermore, the requirement to have an image and several captions, are similar with an image captioning dataset such as Microsoft Common Object in Context (MSCOCO) dataset [14]. The caption of this dataset can be regarded as paraphrase, such as done by Prakash et al. for their neural paraphrase generation study [15]. They stated that the annotators described the most obvious things in an image and concluded that several captions of an image can be counted as paraphrases. While this may be true, we cannot define what kind of operation has been done from the original sentence to the paraphrase. Consequently, the arbitrary nature of the corpus distribution might cause the paraphrases to become noise to each other.

To prevent this, a set of paraphrase operation which covers all possible paraphrase variations needs to be defined. Bhagat and Hovy categorized the variations of how humans paraphrase [16] and argued that "although the logical definition of paraphrases requires strict semantic equivalence, linguistics accept a broader, approximate, equivalence." Based on this idea, they analyzed paraphrase characteristics in various studies and in corpora and established 25 quasi-paraphrase classes, such as change in tenses, metaphor substitution, and, function-word variations.

Given some quasi-paraphrases have very small frequency in the MTC and MSRP corpora as reported by them, we grouped these into 4 elementary paraphrase operations: deletion, insertion, reordering, and substitution. Then, we constructed a paraphrase corpus based on these four operations. The paraphrase collection was done through a crowdsourcing platform on the partial WMT17 Multimodal Translation Task dataset [17]. After that, we constructed our automatic neural paraphrase model based on partial data to generate the paraphrase sentences of the full WMT17 dataset. The details are described below.

2.2. Crowdsourcing Paraphrases on Partial WMT17 Dataset

The WMT17 Multimodal Translation Task dataset [17] contains a set of images with triplets of captions in English, German, and French. The dataset was created from the Flickr30K Entities dataset of image captions in English [18] that was extended to also contain manually translated German and French captions. The data consists of 29000, 1014, and 1000 triplets respectively for the training, development and testing. An out-of-domain dataset consisting 461 images taken from the MSCOCO dataset [14] was also introduced, which contains ambiguous verbs [19].

We focused on paraphrasing the English sentences which are considered as source language. Table 1 shows an example of a paraphrased image caption based on four elementary operations (deletion, insertion, reordering, and substitution) and Figure 1 shows the reference image. As paraphrasing the whole 29k triplet training dataset (29k training dataset) using crowdsourcing would not be efficient in terms of cost and time, we crowdsourced only 10k triplets of this dataset (10k training dataset), along with the whole development and testing datasets.

We used Crowdflower[2] (now Figure Eight) as the crowdsourcing platform. Each crowdworker was instructed to paraphrase at least two image captions for one session. We limited the task to English speakers, or at least those who spoke English as their second language, to maintain quality. We discarded sentences that were not valid such as randomly inputted character, empty string, or captions that aren't English. The crowdsourcing process took about 3 months and 201 workers participated from 16 countries such as the United States, Philippines, and Malaysia. Each workers created 50.1 quintuplets of paraphrases on average.

2.3. Semi-supervised Paraphrase Generation on Full WMT17 Dataset

Furthermore, to complete the paraphrasing on the full WMT17 dataset, we then used 10k quintuplets of crowdsourced paraphrases and constructed neural paraphrase model using four encoder-decoder long short-term memory (LSTM) models with attention [20] for each paraphrase oper-

[2]http://www.figure-eight.com

Table 1: Image caption and example paraphrases

Operation		Sentence
Image Caption		A little gray dog jumps over a small hurdle.
Paraphrase	**Deletion**	A little gray dog jumps over a hurdle.
	Insertion	A little gray dog jumps over a small hurdle successfully.
	Substitution	A little gray dog pass over a small hurdle.
	Reordering	Over a small hurdle, a little gray dog jumps.

ation. We tuned and tested our automatic neural paraphrase model using these crowdsourced paraphrases of the development and testing datasets, respectively. With these four paraphrasing models, we generated multi-paraphrases on the remaining 19k image captions.

The generated 19k dataset was combined with the original crowdsourced 10k training dataset. Finally, these 29k paraphrased dataset are combined with original dataset resulting 58k-triplet training dataset for each operation. In conclusion, the 29k paraphrased training dataset is working as the regularizer for the original dataset. These are the final data that will be used to train a mixture-of-experts translation model, which is described in the next section. The data will be publicly available to augment the WMT17 dataset.

Based on our empirical observation, using paraphrased data on development and testing dataset will reduce the performance of the overall system. When using paraphrased data on development, the training objective becomes unclear, and the loss returned will not represent the real loss. Given that, we emphasize that the use of paraphrased dataset in translation step was done on training step, in combination with original dataset. In this stage, the paraphrases were acting as regularizer and the means of ensembling, improving robustness of the ensembled model as a whole.

3. Neural Caption Translation

This section describes several approaches on using our proposed multi-paraphrase operations to improve NMT. The score of these approaches will then be compared with WMT baseline and our encoder-decoder LSTM NMT baseline.

3.1. Combining All Data in a Single Model

This method was done by just using the paraphrase as a means for data augmentation in source side, such as reported by Nichols et al. (2010) to leverage SMT system [10]. All paraphrases and its original sentence were combined, and the target sentence was duplicated to the number of multiple paraphrases. This approach was done to measure the baseline performance with augmented data.

3.2. Multi-source Model

We implemented Zoph and Knight (2016) multi-source NMT to incorporate various paraphrase inputs with one output [21]. For this model, the encoded representation and attention were combined by concatenation. They reported that this model has the advantage of information triangulation to reduce ambiguity. In their paper, they used several translation pairs such as {French, German} to English in which this triplet of language has similar language structure. However, given this advantages, the use of this model to monolingual input has never been investigated.

3.3. Uniform-weighted Ensemble Model

For this uniform weighted ensemble model, we trained NMT models which source sentence has been paraphrased based on each elementary operations and another one that uses original source sentence, resulting five expert NMT models. After that, these five models are ensembled by averaging each output layer probability distribution, so that every model was weighted uniformly. This model is used to compare the performance with mixture-of-experts model listed in the next subsection, where each expert model have different weight.

The training of this translation model consists of two steps. The first step is to train five translation models based on each paraphrase as the source sentence using the 56k dataset (the combination of original and paraphrased source sentences). Five of those models are trained against the same target sentence. Each model is then regarded as an expert model. Each of the expert models operates on subword level, tokenized by Sentence Piece with 3000 vocabulary unit[3].

3.4. Mixture-of-experts Model

Next, we adopted the mixture-of-experts model proposed by Garmash and Monz (2016). Here, instead of linear layer proposed in their study [22], the expert model is implemented into a single LSTM layer hid that receives the concatenated decoder hidden state output h_n.

$$c_t = tanh(LSTM_{hid}([h_0, h_1, ..., h_n]))$$
$$g_{0:i} = softmax(W_{gate}D(c_t) + b_{gate}).$$

A $softmax$ function is then applied to obtain the weights of each expert model's output layer o_n. Assuming W_n is the weight of the output layer from expert n. Then, the aggregated weight W_{agg} is a linear combination function of each of those weights:

$$W_{agg} = g_0W_0 + g_1W_1 + ... + g_nW_n.$$

[3]https://github.com/google/sentencepiece

Table 2: Paraphrasing model result in BLEU and METEOR

Operation	BLEU	METEOR
Deletion	53.0	42.2
Insertion	56.1	40.5
Reordering	47.2	42.0
Substitution	59.6	44.8

For this model, a 50% dropout D will be applied on the hidden representation after $tanh$ nonlinearity was applied. The regularized representation was further transformed by the gate layer which has the same output size with the number of expert.

A diagram of mixture-of-experts neural caption translation model using our proposed approach is shown in Fig. 2. First, the source sentence is paraphrased into four different paraphrases used to train each of the expert model. Then, each expert will pass their abstract decoding state into mixture model which will produce weights as many as the number of expert. The resulting weight distribution is the linear combination function between each expert's output probability distribution and gating weight produced by mixture model.

4. Experiments

The purpose of this experiment is to choose the best type of model suitable for our multi-paraphrase, by comparing score between Bahdanau et al. NMT baseline and several popular multi-source NMT.

4.1. Setup

We followed the training, development, and test set-up of WMT17 shared task. All result were scored using *multeval* [23] with lowercased and tokenized sentences. We used BLEU [24] and METEOR [25] as evaluation metrics.

The multi-source NMT has five single-depth encoders with 512 hidden size trained with Adam [26]. The mixture-of-experts model was trained using RMSprop optimizer with 0.0001 learning rate [27]. In every increase of development loss, the learning rate is decayed by half into maximum 5 decays. The results are decoded with beam size of 5.

4.2. Evaluation of Neural Paraphrase Model

We constructed four encoder-decoder LSTM models with attention [20] for each elementary paraphrase operation. Each model has a bidirectional encoder and attentional decoder with one layer, 50% dropout ratio, and 512 hidden layer size. Implementation was done using Chainer framework version 3.0 [28] and ran on GTX Titan X GPU. We used Adam [26] as the optimizer with decaying alpha into half in every development loss increase with maximum of 7 decays for training early stopping. After stopping the training, model with the lowest development was selected and used for decoding.

Table 2 lists the scores of the paraphrases produced with our automatic paraphrasing model. The substitution operation produced the highest BLEU score while the reordering operation producing the lowest BLEU score. This was expected because the reordering operation sometimes includes the changing of the active/passive properties of a sentence. Overall, we believe this score is high enough to paraphrase the remaining 19k WMT dataset.

4.3. Translation Model Results

Table 3 shows the performance of our proposed neural caption translation. All results using our multi-paraphrase outperformed the NMT baseline. There are no improvements gained from combining all data, which is the simplest form of data augmentation. This simple combination of data breaks the relation existed between each paraphrases that mention the same image. Furthermore, we cannot be sure that each source sentence has the same amount of paraphrase. By considering these factors, we utilized multi-source NMT and multi-expert NMT, which yield better BLEU and METEOR score.

This performance increase indicates that each expert model is slightly different between each other, and worked well in uniform-weighted ensemble and mixture-of-experts scenario. This model also performed better than uniform-weighted NMT in three cases. Moreover, the mixture-of-experts model performed better in out-of-domain ambiguous MSCOCO test dataset, implying that overfitting did not occur. This also proves the argument that adding additional knowledge will improve model performance on disambiguating inputs. From applying to these several models, we can conclude that our elementary operation paraphrase is suitable to be used as a means for ensembling.

Table 4 shows the current submission systems in the official WMT17 shared task which submissions consist of one textual model [29] and several multimodal models. Our proposed approach outperformed the baseline in WMT17 with a 13.2 BLEU score margin. Our proposed model, although it is textual, could produce competitive result with other multimodal models. The mixture-of-experts model outperformed several multimodal models such as other WMT submission [30, 31, 32, 33]. Even in the out-of-domain dataset of COCO 2017, the mixture-of-experts model also performed reasonably high with a 28.0 BLEU score. Nevertheless, our score was close to that best score. This proved that the paraphrasing of the source side also helped our model to work with unseen data and prevent overfitting.

4.4. Discussion

To further analyze the contribution between the experts trained on the original data and that trained on paraphrased data, we compared the translation process step-by-step in our proposed approach. This source sentence shown in Table 5 was translated using each baseline model (an expert), resulting five different translation hypotheses. Each expert has been trained with slightly different paraphrased source

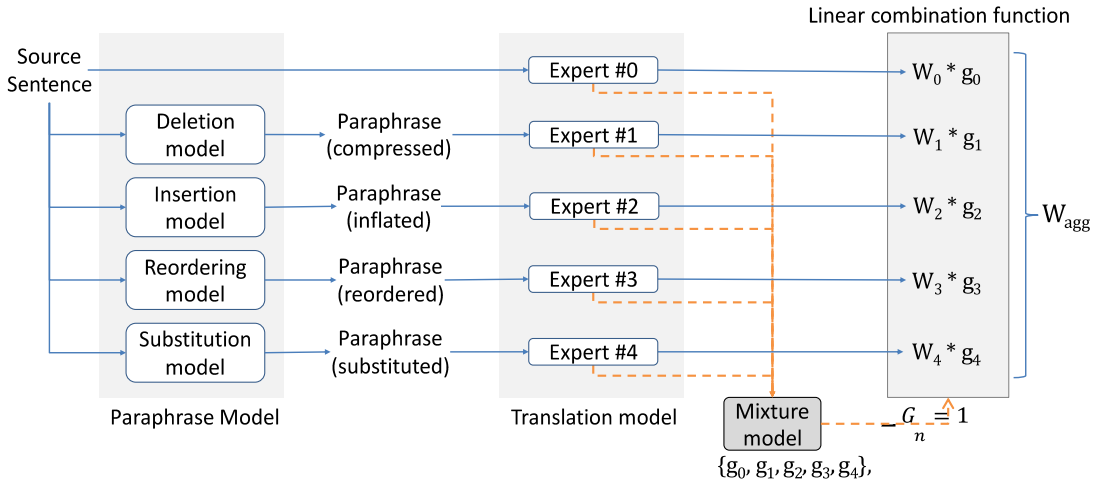


Figure 2: Diagram of proposed mixture-of-experts neural caption translation model

Table 3: The performance of proposed neural caption translation in comparison with the baseline.

Textual Model	Test 2016		Test 2017		Test COCO 2017	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
Our NMT Baseline	37.7	55.6	30.1	49.7	25.0	44.6
Combine all data	36.7	53.9	29.6	47.7	25.1	43.7
Multi-source NMT	37.6	55.4	30.1	49.4	24.4	44.3
Uniform weighted ensemble	39.6	56.9	31.4	50.7	26.7	46.0
mixture-of-experts ensemble	**40.5**	**57.6**	**32.5**	**51.3**	**28.0**	**46.8**

Table 4: Existing submission systems in official WMT17 shared task.

Textual Model	Test 2016		Test 2017		Test COCO 2017	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
Official WMT Baseline	32.5	52.5	19.3	41.9	18.7	37.6
Zhang et al. (2017)	-	-	31.9	53.9	28.1	48.5
Multimodal Model	**Test 2016**		**Test 2017**		**Test COCO 2017**	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
Madhyastha et al. (2017)	-	-	25.0	44.5	21.4	40.7
Calixto et al. (2017)	41.3	59.2	29.8	50.5	26.4	45.8
Ma et al. (2017)	-	-	31.0	50.6	27.4	46.5
Helcl and Libovicky (2017)	36.8	53.1	31.1	51.0	26.6	46.0
Caglayan et al. (2017)	41.0	60.4	33.4	54.0	28.5	48.8

Table 5: Examples of resulting sentences in mixture-of-experts model.

Translation Model	Type	Sentences	BLEU-1
(Data)	**Original**	**two motorcycles drive on a road along the river .**	
Baseline	Original	zwei motorradfahrer fahren auf einer straße entlang .	0.75
Single Paraphrase Model	Deletion	zwei motorräder fahren auf einer straße am fluss .	0.87
	Insertion	zwei motorradfahrer fahren auf einer straße am fluss .	0.84
	Reordering	zwei motorradfahrer fahren auf einer straße am fluss entlang .	0.95
	Substitution	zwei motorradfahrer fahren auf einer straße am flussufer .	0.82
Uniform Weight	Ensemble	zwei motorradfahrer fahren auf einer straße am fluss .	0.84
mixture-of-experts	Ensemble	zwei motorräder fahren auf einer straße am fluss entlang .	**0.97**
(Data)	**Target**	**zwei motorräder fahren auf einer straße dem fluss entlang .**	

sentence. We calculated BLEU-1 scores for each hypothesis against the target, resulting the source-reordered expert model yielded the best result between all experts.

The aim of proposed mixture-of-experts model task is to

make sure the best part of each model is kept, and leaving out any noise or error that might occur in each model result. As can be seen from the German result from the mixture-of-experts model compared with the target sentence, the only difference is the word "*am*" in which the correct one should be "*dem*".

In this example, in deletion translation result, the word "*motorräder*" is decoded instead of "*motorradfahrer*". Another example is the phrase "*fluss entlang*" which can only be found in reordering translation result. This goodness on each expert model however, should be kept by the mixture model by distributing right word in every word being decoded. In conclusion, the final result of the ensemble of expert model combines every goodness in each expert model.

Quantitatively, the mixture-of-experts model successfully kept the good feature of best performing 0.87 and 0.95 BLEU-1 score yielded in source-deleted and source-reordered model results respectively, resulting 0.97 BLEU-1 score. This is a significant improvement compared with the BLEU-1 score of the uniform weighted model that was only increased into 0.84.

5. Conclusions and Future Works

A single caption cannot represent all the information of the image to which it refers to. In this study, we elaborated an image by various paraphrase operations. This enables us to incorporate additional knowledge from image to the translation process, without using the image itself, but diffused in a form of paraphrase.

We successfully generated multi-paraphrase sentences of the WMT17 Multimodal Translation Task dataset through crowdsourcing which will be publicly available. We constructed an automatic paraphrase generation model, and used it with the multi-expert approach within NMT.

The results indicate that our proposed paraphrase elementary operations are best to be used for ensembling, especially on multi-expert ensembling settings. The hypothesis of regularizing models by paraphrasing on the source sentence was proven to be effective. In the future, we will further investigate various methods of incorporating visual information into NMT models.

6. Acknowledgement

Part of this work was supported by JSPS KAKENHI Grant Numbers JP17H06101 and JP17K00237.

7. References

[1] R. De Beaugrande and W. Dressler, *Introduction to text linguistics*, ser. Longman linguistics library. Longman, 1981. [Online]. Available: https://books.google.co.jp/books?id=mvJsAAAAIAAJ

[2] A. Barreiro, *SPIDER: A System for Paraphrasing in Document Editing and Revision — Applicability in Machine Translation Pre-editing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 365–376. [Online]. Available: https://doi.org/10.1007/978-3-642-19437-5_30

[3] C. Callison-Burch, P. Koehn, and M. Osborne, "Improved statistical machine translation using paraphrases," in *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL '06)*, Stroudsburg, PA, USA, 2006, pp. 17–24. [Online]. Available: http://dx.doi.org/10.3115/1220835.1220838

[4] W. He, S. Zhao, H. Wang, and T. Liu, "Enriching smt training data via paraphrasing," in *Proceedings of IJCNLP*, 2011.

[5] M. Simard, N. Ueffing, P. Isabelle, and R. Kuhn, "Rule-based translation with statistical phrase-based post-editing," in *Proceedings of the Second Workshop on Statistical Machine Translation (StatMT '07)*, Stroudsburg, PA, USA, 2007, pp. 203–206. [Online]. Available: http://dl.acm.org/citation.cfm?id=1626355.1626383

[6] M. Simard, C. Goutte, and P. Isabelle, "Statistical phrase-based post-editing," in *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL '07)*, Rochester, NY, USA, 2007, pp. 508–515.

[7] A.-L. Lagarda, V. Alabau, F. Casacuberta, R. Silva, and E. Díaz-de Liaño, "Statistical post-editing of a rule-based machine translation system," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, ser. NAACL-Short '09, Stroudsburg, PA, USA, 2009, pp. 217–220. [Online]. Available: http://dl.acm.org/citation.cfm?id=1620853.1620913

[8] S. Pal, P. Lohar, and S. K. Naskar, "Role of paraphrases in pb-smt," in *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 8404*, ser. CICLing 2014. Berlin, Heidelberg: Springer-Verlag, 2014, pp. 242–253. [Online]. Available: https://doi.org/10.1007/978-3-642-54903-8_21

[9] N. Madnani and B. J. Dorr, "Generating targeted paraphrases for improved translation," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 3, pp. 40:1–40:25, July 2013. [Online]. Available: http://doi.acm.org/10.1145/2483669.2483673

[10] E. Nichols, F. Bond, D. S. Appling, and Y. Matsumoto, "Paraphrasing training data for statistical machine translation," *Journal of Natural Language Processing*, vol. 17, no. 3, pp. 3_101–3_122, 2010.

[11] G. Hirst, "Paraphrasing paraphrased," *Invited talk at the ACL International Workshop on Paraphrasing*, 2003.

[12] I. Calixto, Q. Liu, and N. Campbell, "Doubly-attentive decoder for multi-modal neural machine translation," in *ACL*, 2017.

[13] P. Resnik, O. Buzek, Y. Kronrod, C. Hu, A. J. Quinn, and B. B. Bederson, "Using targeted paraphrasing and monolingual crowdsourcing to improve translation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 3, p. 38, 2013.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[15] A. Prakash, S. A. Hasan, K. Lee, V. Datla, A. Qadir, J. Liu, and O. Farri, "Neural paraphrase generation with stacked residual lstm networks," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, December 2016, pp. 2923–2934. [Online]. Available: http://aclweb.org/anthology/C16-1275

[16] R. Bhagat and E. Hovy, "What is a paraphrase?" *Computational Linguistics*, vol. 39, no. 3, pp. 463–472, 2013.

[17] D. Elliott, S. Frank, K. Sima'an, and L. Specia, "Multi30k: Multilingual english-german image descriptions," in *Proceedings of the 5th Workshop on Vision and Language*, 2016, pp. 70–74.

[18] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," *CoRR*, vol. abs/1505.04870, 2015. [Online]. Available: http://arxiv.org/abs/1505.04870

[19] D. Elliott, S. Frank, L. Barrault, F. Bougares, and L. Specia, "Findings of the Second Shared Task on Multimodal Machine Translation and Multilingual Image Description," in *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September 2017.

[20] 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

[21] B. Zoph and K. Knight, "Multi-source neural translation," *CoRR*, vol. abs/1601.00710, 2016. [Online]. Available: http://arxiv.org/abs/1601.00710

[22] E. Garmash and C. Monz, "Ensemble learning for multi-source neural machine translation," in *COLING*, 2016.

[23] A. L. Jonathan Clark, Chris Dyer and N. Smith, "Better hypothesis testing for statistical machine translation: Controlling for optimizer instability," in *Proceedings of the Association for Computational Lingustics*, 2011.

[24] K. Papineni, S. Roukos, T. Ward, and W. jing Zhu, "Bleu: a method for automatic evaluation of machine translation," 2002, pp. 311–318.

[25] A. Lavie and A. Agarwal, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," 2005, pp. 65–72.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[27] A. Graves, "Generating sequences with recurrent neural networks," *CoRR*, vol. abs/1308.0850, 2013. [Online]. Available: http://arxiv.org/abs/1308.0850

[28] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015. [Online]. Available: http://learningsys.org/papers/LearningSys_2015_paper_33.pdf

[29] J. Zhang, M. Utiyama, E. Sumita, G. Neubig, and S. Nakamura, "Nict-naist system for wmt17 multimodal translation task," in *WMT*, 2017.

[30] P. S. Madhyastha, J. Wang, and L. Specia, "Sheffield multimt: Using object posterior predictions for multimodal machine translation," in *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 470–476. [Online]. Available: http://www.aclweb.org/anthology/W17-4752

[31] I. Calixto, K. D. Chowdhury, and Q. Liu, "Dcu system report on the wmt 2017 multi-modal machine translation task," in *Proceedings of the Conference of Machine Translation (WMT)*, vol. 2, 2017.

[32] M. Ma, D. Li, K. Zhao, and L. Huang, "Osu multimodal machine translation system report," in *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 465–469. [Online]. Available: http://www.aclweb.org/anthology/W17-4751

[33] J. Helcl and J. Libovický, "CUNI system for the WMT17 multimodal translation task," *CoRR*, vol. abs/1707.04550, 2017. [Online]. Available: http://arxiv.org/abs/1707.04550

Using Spoken Word Posterior Features in Neural Machine Translation

Kaho Osamura[1], *Takatomo Kano*[1], *Sakriani Sakti*[1,2], *Katsuhito Sudoh*[1,2], *Satoshi Nakamura*[1,2]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project AIP, Japan
{osamura.kaho.oe5, kano.takatomo.km0, ssakti, sudoh, s-nakamura}@is.naist.jp

Abstract

A spoken language translation (ST) system consists of at least two modules: an automatic speech recognition (ASR) system and a machine translation (MT) system. In most cases, an MT is only trained and optimized using error-free text data. If the ASR makes errors, the translation accuracy will be greatly reduced. Existing studies have shown that training MT systems with ASR parameters or word lattices can improve the translation quality. However, such an extension requires a large change in standard MT systems, resulting in a complicated model that is hard to train. In this paper, a neural sequence-to-sequence ASR is used as feature processing that is trained to produce word posterior features given spoken utterances. The resulting probabilistic features are used to train a neural MT (NMT) with only a slight modification. Experimental results reveal that the proposed method improved up to 5.8 BLEU scores with synthesized speech or 4.3 BLEU scores with the natural speech in comparison with a conventional cascaded-based ST system that translates from the 1-*best* ASR candidates.

1. Introduction

Spoken language translation is one innovative technology that allows people to communicate by speaking in their native languages. However, translating a spoken language, in other words, recognizing speech and then translating words into another language, is incredibly complex. A standard approach in speech-to-text translation systems requires effort to construct automatic speech recognition (ASR) and machine translation (MT), both of which are trained and tuned independently.

ASR systems, which aim for the perfect transcription of utterances, are trained and tuned by minimizing the word error rate (WER) [1]. MT outputs are optimized and automatically measured based on a wide variety of metrics. One of the standard methods is the BLEU metric. However, all the errors from the words in ASR outputs are treated uniformly without considering their syntactic roles, which are often critical for MT. Many studies have investigated the effectiveness of the WER metric of ASR on the whole speech translation pipeline [2, 3, 4] and verified that ASR errors that compose the WER metric do not contribute equally to the BLEU score of translation quality.

Furthermore, most MT systems are only trained and optimized using error-free text data. Despite the fact that ASR technologies and their recognition rates have continued to improve, the occurrence of speech recognition errors remains inevitable. This is because there are many ambiguities due to a wide variety of acoustic and linguistic patterns produced by different speakers with various speaking styles and background noises. If the ASR engine makes mistakes, the translation accuracy will be significantly reduced. Thus, ignoring the existence of ASR errors while constructing a speech translation system is practically impossible.

Previous research on traditional phrase-based MTs has attempted to train the ASR and MT parameters of the log-linear model to directly optimize the BLEU score of the translation metric of full speech translation systems [3]. It allows the model to directly select recognition candidates that are easy to translate and improve the translation accuracy given an imperfect speech recognition. Ohgushi et al. [5] further elaborated various techniques in the context of the joint optimization of ASR and MT, including minimum error rate training (MERT) [6], pair-wise ranking optimization (PRO) [7], and the batch margin infused relaxed algorithm (MIRA) [8]. Other studies directly performed translation on the lattice representations of the ASR output [9, 10, 11]. The results showed that a better translation can be achieved by translating the lattices rather than with the standard cascade system that translated the single best ASR output.

Recently, deep learning has shown great promise in many tasks. A sequence-to-sequence attention-based neural network is one type of architecture that offers a powerful model for machine translation and speech recognition [12, 13]. Several studies revisited similar problems and proposed handling lattice inputs by replacing the encoder part with a lattice encoder to obtain a lattice-to-sequence model [14, 15]. With these methods, robust translation to speech recognition errors became possible. However, this approach requires a large modification to standard NMT systems, resulting in a complicated model that is hard to train. Also, as the NMT takes word lattices as input, it might be difficult to backpropagate a translation error to the ASR part.

An extreme case is to train the encoder-decoder architecture for end-to-end speech translation (ST) tasks, which directly translates speech in one language into text in another. Duong et al. [16] directly trained attentional models on par-

allel speech data. But their work focused only on alignment performance. The works by Berard et al. [17] might be the first attempts that successfully build a full-fledged end-to-end attentional-based speech-to-text translation system. But they only performed with a small parallel French-English BTEC corpus, and their best results were behind the cascade baseline model. Later on, Weiss et al. [18] proposed a similar approach and conducted experiments on the Spanish Fisher and Callhome corpora of telephone conversations augmented with English translations. However, most of these works were only done for language pairs with similar syntax and word order (SVO-SVO), such as Spanish-English or French-English. For such languages, only local movements are sufficient for translation. Kano et al. [19] showed that direct attentional ST approach failed to handle English-Japanese language pairs with SVO versus SOV word order.

In this research, we also focus on English-Japanese and we aim for a neural speech translation that is robust against speech recognition errors without requiring significant changes in the NMT structure. This can be considered as a simplified version of the one that directly performed translation on the lattice representations. But, instead of providing full lattice outputs, we perform a neural sequence-to-sequence ASR as feature processing that is trained to produce word posterior features given spoken utterances. This might resemble the word confusion networks (WCNs) [20] that can directly express the ambiguity of the word hypotheses at each time point. The resulting probabilistic features are used to train NMT with just a slight modification. Such vectors are expected to express the ambiguity of speech recognition output candidates better than the standard way using the 1-$best$ ASR outputs while also providing a simpler structure than the lattice outputs. During training, the approach also allows backpropagating the errors from NMT to ASR and performs join training. Here, we evaluate our proposed English-Japanese speech translation model using both synthesized and natural speech with various degrees of ASR errors.

2. Overview of Attention-based Speech Translation

Our English-Japanese end-to-end speech translation system consists of ASR and MT modules that were constructed on standard attention-based, encoder-decoder neural network architecture [21, 22].

2.1. Basic Attentional Encoder-Decoder model

An attentional encoder-decoder model consists of an encoder, a decoder, and attention modules. Given input sequence $\boldsymbol{x} = [x_1, x_2, ..., x_N]$ with length N, the encoder produces a sequence of vector representation $h^{enc} = (h_1^{enc}, h_2^{enc}, ..., h_N^{enc})$. Here, we used a bidirectional recurrent neural network with long short-term memory (bi-LSTM) units [23], which consist of forward and backward LSTMs. The forward LSTM reads the input sequence from x_1 to x_N and estimates forward $\overrightarrow{h^{enc}}$, and the backward LSTM reads the input sequence in reverse order from x_N to x_1 and estimates backward $\overleftarrow{h^{enc}}$. Thus, for each input x_n, we obtain h_n^{enc} by summation forward $\overrightarrow{h^{enc}}$ and backward $\overleftarrow{h^{enc}}$:

$$h_n^{enc} = \overrightarrow{h_n^{enc}} + \overleftarrow{h_n^{enc}}. \quad (1)$$

The decoder, on the other hand, predicts target sequence $\boldsymbol{y} = [y_0, y_1, y_2, ..., y_M]$ with length M by estimating conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$. Here, we use uni-directional LSTM (forward only). Conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$ is estimated based on the whole sequence of the previous output:

$$p(y_m|\boldsymbol{y}_{<m}, \boldsymbol{x}) = \text{softmax}(W_y \tilde{h}_m^{dec}). \quad (2)$$

Decoder hidden activation vector $\tilde{h}_m^{dec}$ is computed by applying linear layer W_c over context information c_m and current hidden state h_m^{dec}:

$$\tilde{h}_m^{dec} = \tanh(W_c[c_m; h_m^{dec}]). \quad (3)$$

Here, c_m is in the context information of the input sequence when generating current output at time m. It is estimated by the attention module over encoder hidden states h_n^{enc}:

$$c_m = \sum_{n=1}^{N} a_m(n) * h_n^{enc}, \quad (4)$$

where variable-length alignment vector a_m is computed whose size equals length of input sequence x:

$$\begin{aligned} a_m &= \text{align}(h_n^{enc}, h_m^{dec}) \\ &= \text{softmax}(\text{dot}(h_n^{enc}, h_m^{dec})). \end{aligned} \quad (5)$$

This step assists the decoder to find relevant information on the encoder side based on the current decoder hidden states. Several variations calculate align(h_n^{enc}, h_m^{dec}). Here, we simply use the dot product between the encoder and decoder hidden states.

2.2. Automatic Speech Recognition

Speech recognition tasks estimate a word sequence given a sequence of speech features. Input sequence $\boldsymbol{x} = [x_1, ..., x_N]$ is the input speech filter bank feature sequence of the source language, and target sequence $\boldsymbol{y} = [y_1, ..., y_M]$ is the predicted corresponding word sequence in the source language.

2.3. Machine Translation

Machine translation tasks estimate a word sequence of a target language given a word sequence of a source language. Input sequence $\boldsymbol{x} = [x_1, ..., x_N]$ is the word sequence of the source language, and target sequence $\boldsymbol{y} = [y_1, ..., y_M]$ is the predicted corresponding word sequence in the target

language. Here, x_n is a one-hot vector in the baseline or posterior vector in the proposed method, y_m is the index representation of the words, and y_0 is an index representation of the target sequence's start.

2.4. Speech-to-text Translation

Speech-to-text translation tasks estimate a word sequence of a target language given a sequence of speech features. Here, we use both the sequence-to-sequence ASR and MT systems. Output sequence $\boldsymbol{y}$ from ASR becomes input sequence $\boldsymbol{x}$ in an MT system.

3. Proposed method: NMT using Spoken Word Posterior Features

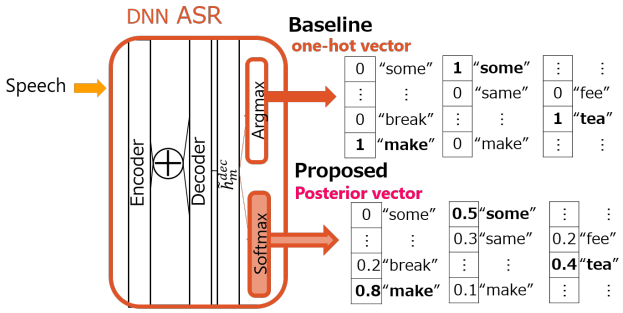


Figure 1: Construction of spoken word posterior features

Fig. 1 illustrates the construction of spoken word posterior features. Here, we train an end-to-end ASR using the standard attention-based encoder-decoder neural network architecture described in the previous section. But instead of providing 1-*best* outputs of the most probable word sequence to the translation system,

$$\hat{y}_m = \underset{y_m}{\operatorname{argmax}}\, p(y_m|\boldsymbol{y}_{<m}, \boldsymbol{x}), \tag{6}$$

we utilize the posterior probability vectors before the argmax function:

$$p(y_m|\boldsymbol{y}_{<m}, \boldsymbol{x}). \tag{7}$$

This way the vectors can still express the ambiguity of the speech recognition output candidates with probabilities.

The resulting probabilistic features are then used to train the NMT with only a slight modification. We train the end-to-end NMT using the standard attention-based encoder-decoder neural network architecture described in the previous section. The only difference is in the input features. Instead of training the model with the one-hot vector of the most probable words, we utilize the posterior vectors obtained from the ASR. However, the dimension of input vector representation used in a standard one-hot vector and the proposed posterior vectors is the same. The overall architecture is illustrated in Fig. 2.

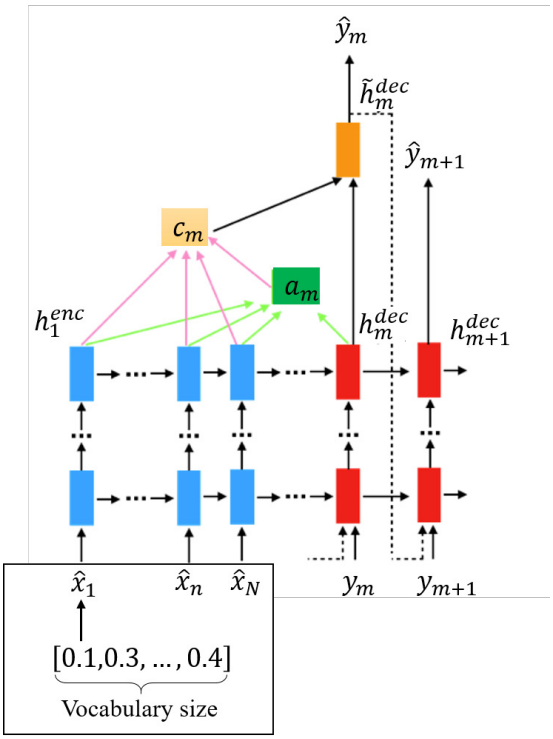


Figure 2: Proposed NMT architecture

4. Experiments

We evaluated the performance of the proposed method on an English-Japanese translation task. To simulate the effect of various ASR errors, we first assessed it on synthesized speech and later applied it to natural speech.

4.1. Data set

The experiments were conducted using a basic travel expression corpus (BTEC) [24]:

- **Text corpus**
  We used a BTEC English-Japanese parallel text corpus that consists of about 460k (BTEC1-4) training sentences and 500 sentences in the test set.

- **Synthesized speech corpus**
  Since corresponding speech utterances for the BTEC parallel text corpus are not available, we used Google text-to-speech synthesis [25] to generate a speech corpus of the BTEC1 source language (about 160k utterances). We used about 500 speech utterances in the test set.

- **Natural speech corpus**
  We also evaluated with natural speech. In this case, we used the ATR English speech corpus [26] in our experiments. The text material was based on the basic travel expression domain. The speech corpus we used consisted of American, British, and Australian (AUS)

English accents with about 120k utterances spoken by 100 speakers (50 males, 50 females) for each accent.

The speech utterances were segmented into multiple frames with a 25-ms window size and a 10-ms step size. Then we extracted 23-dimension filter bank features using Kaldi's feature extractor [27] and normalized them to have zero mean and unit variance.

4.2. Models

We further used the data to build a speech translation system with attention-based ASR and MT systems. The ASR and NMT share the same vocabulary (16,745 words). The dimensions of the distributed vector representation are smaller than vocabulary size (the size depends on the model settings). The hidden encoder and decoder layer consists of 500 nodes. A batch size of 32 and a dropout of 0.1 were also applied. For all systems, we used a learning rate of 0.0001 for the encoder and 0.0005 for the decoder and adopted Adam [28] to all the models.

As we aim to have a neural speech translation that is robust against speech recognition errors without requiring significant changes in the NMT structure. We constructed three types of models that fit those requirements:

- **Text-based machine translation system (upper-bound)**
  This is a text-to-text translation model from the source language to the target language. Here the BTEC English-Japanese parallel text corpus is used to train the model.

- **Baseline speech translation**
  This speech-to-text translation model was created by cascading the ASR (speech-to-text) in the source language with a text-to-text MT module using 1-*best* ASR outputs. First, we pre-trained the NMT with the BTEC English-Japanese parallel text corpus and then fine-tuned the NMT model with a one-hot vector provided from the ASR.

- **Proposed speech translation**
  This speech-to-text translation model was created by cascading ASR (speech-to-text) in the source language with the text-to-text MT module using the ASR posterior vectors. First, we pre-trained the ASR with the speech of the source language and the NMT with the BTEC English-Japanese parallel text corpus. After that, we fine-tuned the parameter of both models by jointly training, where the posterior vector of ASR output is used as the NMT input.

Note that the ASR systems used for the baseline and the proposed systems are the same. Also, all translation systems were tuned adequately, and the best model from training epochs was selected for each system.

5. Result

5.1. Speech Recognition System

To simulate different degrees of ASR errors, we constructed an ASR model using synthesized speech with different numbers of training epochs, resulting in four different models with the following WERs: (1) System 1 (WER=15.17%), System 2 (WER=12.34%), System 3 (WER=11.05%), and System 4 (WER=8.82%). As a model that is trained with natural speech, our performance achieved a 24.98% WER.

5.2. Translation System

As mentioned earlier, we compare three translation system: one for standard text-based machine translation, one for baseline speech translation with the cascade model, and one for our proposed speech translation.

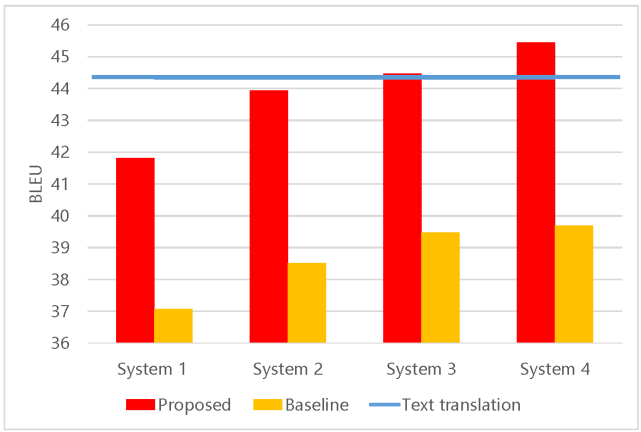


Figure 3: Translation quality given synthesized speech input

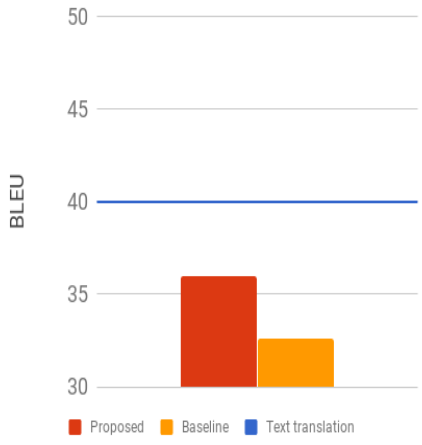


Figure 4: Translation quality given natural speech input

The quality of those translation systems with the input of synthesized speech was evaluated using BLEU [29] and

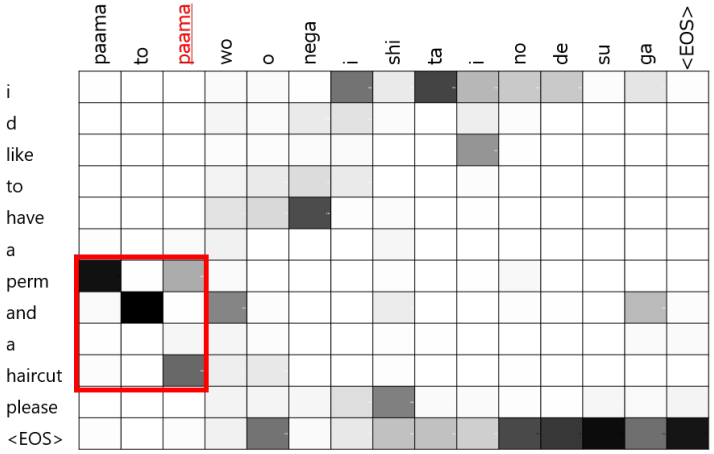


Figure 5: Attention matrix of text translation

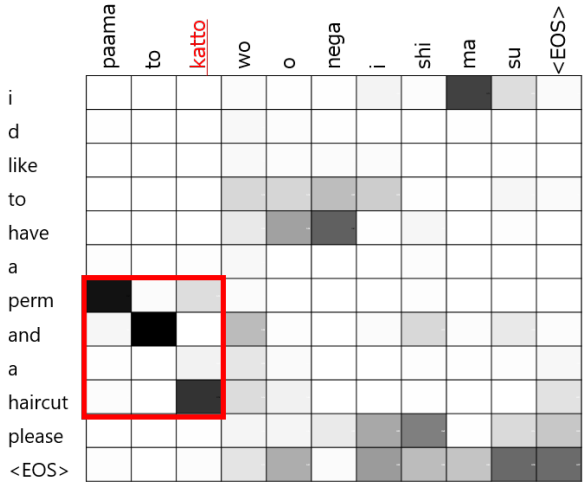


Figure 6: Attention matrix of proposed method

shown in Fig. 3. Here System 1-4 represent of using different ASR systems (1-4), respectively. The results show that the better the ASR performance, the stronger the baseline cascade model. Nevertheless, our proposed approach stable outperformed the cascade model in all cases. The BLEU score improved from 4.8 to 5.8 compared to the baseline model.

The proposed methods (System 3 and 4) exceed the text translation because the recognition candidates included in the posterior vector made it possible to correctly distinguish confusing words in the word embedding of the text translation. We will scrutinize this result in the next section.

Next, the quality of the speech translation systems using natural speech was also evaluated using BLEU and shown in Fig. 4. For the text translation, we provided the transcription of the natural speech, which is different than the text used in Fig. 3. This system used the ASR model where WER is 24.98%. Importantly, unlike several published ASR systems using BTEC dataset, our ASR system only used the text transcription of the training set for the language model. Therefore, the ASR results reported in the paper could not reach state-of-the-art ASR performance. Nevertheless, the translation results are still convincing as evidence of the proposed framework's effectiveness. The proposed method improved the 4.3 BLEU score of the baseline model, confirming that the proposed method is also effective for natural speech.

6. Discussion

Table 1 shows the sentence output examples in English-Japanese translation: (1) with ASR error, and (2) without ASR error. In the first example, to analyze the effect of ASR error, we compare the sentence output of the proposed model and the baseline (the cascase model). Here, ASR misrecognized "shoe" as "station". This error impacted the baseline (cascade system), where it translated "station" as "`eki`" (the correct translation for "shoe store" is "`kutsuya`"). However, in the proposed method, it was still able to translate it to "`kutsuya`". This might be because the ASR provided a posterior vector in which the recognition candidate and each a posteriori probability are weighted (Table 2). Here, "shoe" information was still contained in the posterior vector with only slightly lower probability than "station," and based on the context information, the machine translation translated the word as "`kutsuya`."

In the second example, ASR provided a correct sentence. Here, we compare the sentence output of the proposed model and the text translation. Since the contexts of "perm" and "haircut" are close, the text translation mistakenly translated

Table 1: Examples of sentences output: (1) with ASR error, and (2) without ASR error.

Example 1: With ASR error	
ASR reference	Excuse me where is the closest shoe store?
ASR result	Excuse me where is the closest station store?
Baseline	`Sumimasen ichiban chikai eki wa doko desuka?`
Proposed	`Sumimasen ichiban chikai kutsuya wa doko desuka?`
MT reference	`Sumimasen ichiban chikai kutsuya wa doko desuka?`
Example 2: Without ASR error	
ASR reference	i d like to have a perm and a haircut please
ASR result	i d like to have a perm and a haircut please
Text translation	`Paama to paama o onegai shitai nodesuga`
Proposed	`Paama to katto o onegaishimasu`
MT reference	`Paama to katto o onegaishimasu`

Table 2: Posterior vector

Recognized	Posterior
station	0.439
shoe	0.321
change	0.086
cashier	0.036
always	0.016

both "perm" and "haircut" into "paama" (Fig. 5 illustrates the text translation's alignment matrix). On the other hand, having a posterior vector as the input in the proposed model (see the attention matrix in Fig. 6) allowed NMT to correctly distinguish confusing words by the word embedding of the text translation.

7. Conclusions

In this research, a speech translation system that is robust against speech recognition errors is obtained by using a posterior vector, which is a normalized vector that expresses the ambiguity of the speech recognition candidates, as the input of an NMT engine. The lower the WER of the ASR model is, the weaker the tendency of translation error becomes. Nevertheless, the whole test's accuracy surpassed the baseline. As a result, the posterior vector improved the BLEU score by 4.8 to 5.8 points over the baseline in the simulation experiment and improved it by 4.3 BLEU points over the baseline in the experiment using natural voice. By providing the probability of the speech recognition output candidates in speech translation, an optimal input selection for NMT was made. In the future, we will directly perform join training from ASR to NMT.

8. Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Numbers JP17H06101 and JP17K00237. We also thank Phillip Arthur, Do Quoc Truong and Andros Tjandra for their feedback and insightful discussions.

9. References

[1] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 14–36, 2008.

[2] P. Dixon, A. Finch, C. Hori, and H. Kashioka, "Investigation on the effects of ASR tuning on speech translation performance," in *IWSLT*, San Francisco, USA, 2011.

[3] X. He, L. Deng, and A. Acero, "Why word error rate is not a good metric for speech recognizer training for the speech translation task?" in *ICASSP*, Prague, Czech Republic, 2011.

[4] N. Ruiz and M. Federico, "Assessing the impact of speech recognition errors on machine translation quality," in *Association for Machine Translation in the Americas (AMTA)*, 2014, pp. 261–274.

[5] M. Ohgushi, G. Neubig, S. Sakti, T. Toda, and S. Nakamura, "An empirical comparison of joint optimization techniques for speech translation," in *INTERSPEECH*, 2013, pp. 2619–2623.

[6] F. J. Och, "Minimum error rate training in statistical machine translation," in *ACL*, Sapporo, Japan, 2003.

[7] M. Hopkins and J. May, "Tuning as ranking," in *EMNLP*, Edinburgh, UK, 2011, pp. 1352–1362.

[8] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *NAACL*, Montreal, Canada, 2012, pp. 34–35.

[9] S. Saleem, S.-C. Jou, S. Vogel, and T. Schultz, "Using word lattice information for a tighter coupling in speech translation systems," in *ICSLP*, Jeju Island, Korea, 2004, pp. 41–44.

[10] R. Zhang, G. Kikui, H. Yamamoto, and W.-K. Lo, "A decoding algorithm for word lattice translation in speech translation," in *IWSLT*, Pittsburgh, USA, 2005, pp. 23–29.

[11] E. Matusov, B. Hoffmeister, and H. Ney, "ASR word lattice translation with exhaustive reordering is possible," in *Interspeech*, Brisbane, Australia, 2008, pp. 2342–2345.

[12] j. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

[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. Su, Z. Tan, D. Xiong, R. Ji, X. Shi, and Y. Liu, "Lattice-based recurrent neural network encoders for neural machine translation," in *AAAI*, 2017, pp. 3302–3308.

[15] M. Sperber, G. Neubig, J. Niehues, and A. Waibel, "Neural lattice-to-sequence models for uncertain inputs," in *EMNLP*, Copenhagen, Denmark, 2017, p. 13801389.

[16] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn, "An attentional model for speech translation without transcription," in *HLT-NAACL*, 2016.

[17] A. Berard and O. Pietquin, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," in *30th Conference on Neural Information Processing Systems*, 2016.

[18] R. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

[19] T. Kano, S. Sakti, and S. Nakamura, "Structured-based curriculum learning for end-to-end English-Japanese speech translation," in *INTERSPEECH*, 2017.

[20] 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.

[21] 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.

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

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world," in *LREC*, 2002, pp. 147–152.

[25] "Google Text to Speech API," https://github.com/pndurette/gTTS.

[26] S. Sakti, M. Paul, A. Finch, X. Hu, J. Ni, N. Kimura, S. Matsuda, C. Hori, Y. Ashikari, H. Kawai, H. Kashioka, E. Sumita, and S. Nakamura, "Distributed speech translation technologies for multiparty multilingual communication," *TSLP*, vol. 9, pp. 4:1–4:27, 2012.

[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

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

[29] 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.

Index

www.iwslt.org
2018