THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Samsung and University of Edinburgh's System for the IWSLT 2019

Citation for published version:
Wetesko, J, Chochowski, M, Przybysz, P, Williams, P, Grundkiewicz, R, Sennrich, R, Haddow, B, Miceli Barone, A & Birch-Mayne, A 2019, 'Samsung and University of Edinburgh's System for the IWSLT 2019', Paper presented at 16th International Workshop on Spoken Language Translation 2019, Hong Kong, 2/11/19 - 3/11/19. https://doi.org/10.5281/zenodo.3525537

Digital Object Identifier (DOI):
10.5281/zenodo.3525537

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

OPEN ACCESS

Download date: 19. Apr. 2024

Samsung and University of Edinburgh's System for the IWSLT 2019

Joanna Wetesko[1], Marcin Chochowski[1], Pawel Przybysz[1], Philip Williams[2], Roman Grundkiewicz[2]
Rico Sennrich[2], Barry Haddow[2], Antonio Valerio Miceli Barone[2] Alexandra Birch[2]

[1]Samsung R&D Institute, Poland
[2]School of Informatics, University of Edinburgh
{j.wetesko,m.chochowski,p.przybysz}@samsung.com
{pwillia4,rgrundki,bhaddow,amiceli}@inf.ed.ac.uk, {rico.sennrich,a.birch}@ed.ac.uk

Abstract

This paper describes the joint submission to the IWSLT 2019 English to Czech task by Samsung R&D Institute, Poland, and the University of Edinburgh. Our submission was ultimately produced by combining four Transformer systems through a mixture of ensembling and reranking.

1. Introduction

This paper describes the joint submission to the IWSLT2019 Text Translation task by Samsung R&D Institute, Poland (SRPOL), and the University of Edinburgh (UEDIN). Our methods are based on our approach from last year [1] including exploiting non-parallel resources, large scale use of back-translation for model fine-tuning, as well as model ensembling. This year, however, we decided to train our NMT systems using Marian toolkit [2], focusing on experimenting with bigger Transformer [3] architectures and significantly simplifying the data pre-processing stage.

Our final submission involved the combination of four individual systems. Two of these were trained from scratch using a mixture of WMT and MUST-C data. We describe the training data in detail in Section 2 and the specifics of the system architecture and training in Section 3. Since we had already trained systems for the WMT19 News Translation Task, we experimented with fine-tuning on the MUST-C data and used the resulting systems for reranking. We describe these additional systems in Section 4. The final combination of systems used in our submission is detailed in Section 5.

2. Training Data

2.1. Provided training data

We constructed several systems using different combinations of supplied data for training. We used the in-domain TED MUST-C data, as well as all other provided parallel resources. The amount of in-domain data was relatively small compared to the size of out-of-domain corpora. Therefore, we decided to generate more pseudo in-domain synthetic sentence pairs through back-translation of monolingual Czech CommonCrawl and NewsCrawl 2018, resulting in 32M additional parallel sentence pairs. The details of the back-translation system and the whole filtering process are described in section 2.2.

As a result, we created three training sets used in our experiments. The main one was a set constructed using provided parallel data, including 10 times ovesampled MUST-C TED sentence pairs (around 64M lines in total). This will be referred later as the *base* training set. It was further extended with back-translated lines from CommonCrawl (80M), constituting the second set. The third set was composed of synthetic back-translated sentence pairs from NewsCrawl (16M) and used only for further model domain adaptation.

We noticed overlapping sentences in provided training and MUST-C development sets. After filtering them out from the development set, we got a set containing 975 lines. During all trainings we used that pruned MUST-C development set for progress validation.

We filtered out noisy lines in our training corpus, containing garbled encoding and unnecessary or rare characters. Furthermore, lines longer than 500 characters and empty lines were removed.

2.2. Additional synthetic data (back-translation)

To increase the rate of in-domain data in the training corpora we decided to apply back-translation for data augmentation [4]. To select a subset of best TED-like sentences from monolingual general-domain Czech CommonCrawl and NewsCrawl corpora we used the approach described in [5]. Two RNN language models were constructed using Marian toolkit: in-domain trained with MUST-C corpus and out-of-domain created using the same number of lines from CommonCrawl and NewsCrawl respectively. All these models were regularized with exponential smoothing of 0.0001, dropout of 0.2 along with source and target word token dropout of 0.1. For CommonCrawl and NewsCrawl sentence ranking, we used cross-entropy difference between scores of previously mentioned models as suggested in [5], normalized by the line length. Only sentences with score above arbitrarily chosen threshold were selected for further processing.

We observed that our back-translation system favors very

Table 1: BLEU *score results for single transformer systems on the cleaned en-cz IWSLT dev set 2.*

Model		
Base	Extension	BLEU score
Base transformer-big		24.9
Base transformer-big	+ synthetic data	25.0
Base transformer-huge	+ synthetic data	26.0
Transformer-big	+ synthetic data + fine-tuning (TED MUST-C)	25.3
Transformer-big	+ synthetic data + fine-tuning (NewsCrawl)	25.3
Transformer-huge	+ synthetic data + fine-tuning (TED MUST-C)	**26.3**
Transformer-huge	+ synthetic data + fine-tuning (NewsCrawl)	26.1
Transformer-huge	+ lexical shortcuts	25.5

short sentences, containing one or two words. Although these sentences matched TED domain perfectly well, they would not provide any additional valuable information to the model in training. Therefore, we decided to filter the input sentences, leaving only these consisting of more than 10 words. We have also excluded lines containing more than 500 characters.

All filtered sentences were back-translated into English with a bilingual Czech-English standard *transformer-big* model, trained using Marian toolkit in a similiar manner as described in section 3.2.1. Finally, we produced additional 16M new synthetic pseudo in-domain parallel sentence pairs.

3. SRPOL Systems

We constructed several transformer systems of various sizes, using different combinations of provided and augmented training data. The configuration of each system is more broadly specified in separate sections below. Evaluation was performed using SACREBLEU [6] on a reduced version of a provided development set, as indicated in Section 2.

3.1. Preprocessing

Following promising results achieved in [7], this time we also skipped common pre- and post-processing pipeline steps like tokenization and truecasing. We continued to use Unigram Language Model (ULM) from SentencePiece [8] as a segmentation algorithm on raw text. We learnt 32k subword units jointly on 10M sampled English and Czech sentences, with character coverage increased to 0.9999%.

3.2. Base systems

3.2.1. SRPOL - Transformer-big

Our initial attempt was to train an expanded version of transformer base model [3], which we will now refer to as *transformer-big*. This model consists of 6 encoder layers, 6 decoder layers, 16 heads, a model/embedding dimension of 1024 and a feed-forward network dimension of 4096.

The model is regularized with dropout between transformer layers of 0.2, label smoothing of 0.1 and exponential smoothing of 0.0001. We also used layer normalization [9] and tied the weights of the target-side embedding and the transpose of the output weight matrix as well as source- and target-side embeddings [10].

Adam [11] was used as an optimizer, providing a learning rate of 0.0003 and linear warm-up for first 32000 updates with subsequent inverted squared decay.

In preliminary experiments we decided to examine the influence of provided extra back-translated sentence pairs. We trained two *transformer-big* models using parameters mentioned above, one using only the *base* training set described in section 2, the other one with additional back-translated synthetic parallel data extracted from CommonCrawl.

We can see the results of the *transformer-big* experiments in Table 1. The second model reached slightly better BLEU score (24.9 BLEU versus 25.0 BLEU), therefore it was used as a base for further fine-tuning.

3.2.2. SRPOL - Transformer-huge

As the size of training corpora was big enough, during another experiment we inspected the impact of using an even bigger transformer architecture, referred to from now on as *transformer-huge*. We expanded the size of feed-forward network to 8192 in both encoder and decoder blocks, hoping this can model additional dependencies in the output of attention layers. All the other parameter were kept as in *transformer-big* described above. In Table 1 we can see that *transformer-huge* gains 1 BLEU points over the *transformer-big* system.

3.3. Extensions

3.3.1. Fine-tuning

Our models were mainly trained on general data of various domains, therefore we expected that additional model adaptation using the in-domain data could bring even better results. We tried different approaches again using TED MUST-C and pseudo in-domain data extracted from NewsCrawl that was not provided to the model beforehand. Following [12], we also experimented with using ultra-large mini-batches during fine-tuning with TED MUST-C data, hoping to improve convergence. We increased the mini-batch size by delaying the gradient update, allowing the whole dataset to be read during one update, which unfortunately resulted in 0.1-0.2 BLEU loss, therefore we decided to drop this approach.

We switched the optimizer from Adam to SGD, turned off the learning rate warm up and inverted squared decay, leaving the learning rate flat and equal to 0.0001. We increased dropout probability to 0.3, 0.1, 0.1 between transformer layers, in attention and between feed-forward layers respectively. Consequently, two new improved models reached 0.2 higher BLEU score, one fine-tuned with TED MUST-C data, one with back-translated and filtered NewsCrawl data.

Following this approach we conducted several experiments to fine-tune the new upgraded *transformer-huge* model, again using both TED MUST-C corpus and pseudo in-domain data from NewsCrawl. The best results were once more achieved by using adaptation settings described above - 26.3 BLEU and 26.1 BLEU for mentioned adaptation data sets respectively, shown in Table 1.

3.3.2. Lexical shortcuts

One of the areas we investigated in order to improve our submission was implementing Lexical shortcuts [13] as a Marian extension and applying them to a new version of *transformer-huge*. Lexical shortcuts were proposed in order overcome the difficulty of propagating through a deep network of hidden layers. To alleviate this bottleneck, gated shortcut connections are introduced between the embedding layer and each subsequent layer within the encoder and decoder which enables the model to access relevant lexical content dynamically. Looking at the final row in Table 1 we can see that on top of *transformer-huge*, this did not improve results (-0.5 BLEU).

3.4. Model ensembles

Our top-performing systems were created using model ensembling. We tested two different methods: firstly a straightforward ensembling technique built into Marian framework and secondly by generating n-best list of translation hypotheses with one model, and reranking it with another model. The first method turned out to work much better, allowing us to achieve up to 1 BLEU score improvement. This procedure was further combined with optimal decoder parameters search, focusing on target sentence length normalization and choosing a beam size during beam search. All our most important model compositions together with best decoder parameters setting are presented in Table 2.

The best result was achieved through a combination of equally weighted *transformer-big* fine-tuned on back-translated NewsCrawl pseudo in-domain data (individually scoring 25.3 BLEU) and *transformer-huge* fine-tuned on in-domain TED MUST-C data set (with separate score of 26.3 BLEU). This final SRPOL model ultimately reached a score of 27.1 BLEU.

Table 2: BLEU *score results for ensemble transformer systems on the cleaned en-cz IWSLT dev set 2*

Translation		
Model	Decoder	BLEU score
Base transformer-big + fine-tuning (TED MUST-C) Base transformer-huge	Beam-size 4 Normalization: 1.3 Weights: 0.5 0.5	26.7
Base transformer-big + fine-tuning (NewsCrawl) Base transformer-huge	Beam-size 4 Normalization: 1.3 Weights: 0.5 0.5	26.7
Base transformer-big + fine-tuning (TED MUST-C) Base transformer-huge + fine-tuning (TED MUST-C)	Beam-size 20 Normalization: 1.3 Weights: 0.5 0.5	27.0
Base transformer-big + fine-tuning (TED MUST-C) Base transformer-huge + fine-tuning (NewsCrawl)	Beam-size 20 Normalization: 1.3 Weights: 0.5 0.5	**27.1**

4. WMT19 Systems

Earlier in the year, a team from the University of Edinburgh participated in the WMT19 Shared Task on News Translation[1]. Since the training data that is permitted for the current task is the same as it was for WMT (with the addition of MUST-C), we already had a fully-trained English-Czech system ready to use (albeit not adapted to the target domain).

The WMT19 system was an ensemble of two independently trained Transformer-Big models. They were trained using all provided parallel corpora, except for CommonCrawl. The training data was first cleaned, using simple heuristics, and then filtered, using a one-directional version of dual conditional cross-entropy filtering [14].

The data was cleaned to remove duplicate sentence pairs and pairs where i) the Czech sentence did not include at least one Czech diacritic character; ii) either sentence contained less than three or more than 200 tokens; iii) the ratio of alphabetic to non-alphabetic characters was less then 0.5.

After cleaning, the data was filtered by first training a Czech→English Transformer Base system and using that to score the training data. The sentence pairs with cross-entropy scores in the bottom 5% were removed.

System performance was improved with the addition of synthetic data, which was produced as follows: i) English

[1] `http://statmt.org/wmt19/translation-task.html`

monolingual text was translated using a English→Czech Transformer Base system; ii) Czech monolingual text was translated using a Czech→English system trained on the filtered parallel data and synthetic data from i.

The final WMT19 systems were then trained using a combination of the filtered parallel data (44.93M sentence pairs) and the back-translated Czech (80M sentence pairs).

As in our other systems, the WMT19 systems used a Unigram Language Model (ULM) with a vocabulary of 32k subword units learned jointly 10M sampled English and Czech sentences. They used the Transformer Big architecture and were trained using the Adam optimizer with a learning rate of 0.0002. The models were regularised with label smoothing (at a rate of 0.1) and dropout, at a rate of 0.2 between transformer layers and 0.1 in attention and feed-forward layers. Mini-batches were dynamically fitted into 48GB of memory on 4 GPUs, with updates delayed by one iteration, resulting in mini-batches of 1-1.2k sentences. For further details of the WMT19 systems, see Section 5 of [7].

4.1. Fine-Tuning

In order to adapt the models to the target domain, we continued training using a mixture of 3M sentence pairs of WMT19 data and all of MUST-C, which was oversampled by a factor of ten (giving 1.28M sentence pairs). We used early stopping with a patience of 5 based on word-level cross-entropy on the cleaned development set. The models were validated every 2k updates, and we selected the best model checkpoint according to uncased BLEU score. Table 3 shows performance of the two WMT19 systems with and without fine-tuning, and used both independently and as an ensemble. We also experimented with increasing dropout rates, but we obtained worse results.

4.2. Tagged Fine-Tuning

In addition to the standard fine-tuning approach, we also experimented with tagged fine-tuning for domain adaptation. Since the fine-tuning dataset contains a mix of WMT19 data (out-of-domain) and MUST-C data (in-domain), we explicitely provide the model with an extra bit of information for each sentence indicating which corpus, and therefore which domain, it came from. We provide this information by concatenating a domain indicator tag at the end of each source sentence, similar to [15, 16]. However, since we were fine-tuning pre-trained models with an already fixed vocabulary, we could not reserve two unique token types as domain indicator tags. Therefore we designated the two least frequent token types in the pre-trained BPE vocabulary as domain indicator tags. We fine-tuned and evaluated the models using the same hyperparameters of sec. 4.1. The results were within 0.1 BLEU points of untagged fine-tuning, therefore we did not use this approach in our final submission in order to avoid unnecessary extra complexity for system combination.

Table 3: BLEU *score results for WMT19 systems on the cleaned en-cz IWSLT dev set 2.*

	BLEU Score	
System	Original	Fine-tuned
WMT19.1	22.7	25.5
WMT19.2	22.3	25.3
Ensemble	22.8	25.8

Table 4: *Summary of models used in our final submission*

Model	Norm.	Weight
Base transformer-big		
+ fine-tuning (MUST-C)	1.3	1.0
Base transformer-huge		
+ fine-tuning (NewsCrawl)	1.3	1.0
WMT19.1		
+ fine-tuning (WMT19 + MUST-C)	2.2	0.5
WMT19.2		
+ fine-tuning (WMT19 + MUST-C)	2.2	0.5

5. System Combination

Our strongest systems in Sections 3 and 4 were both ensembles of two independently trained systems. In this Section, we'll refer to the final system in Table 2 as *ensemble-a* and the final (fine-tuned) system in Table 3 as *ensemble-b*. Unfortunately, we could not combine the four component systems into a single ensemble due to the different vocabularies used by the *ensemble-a* and *ensemble-b* systems. Instead we used the following reranking procedure:

1. Produce two 20-best lists using *ensemble-a* and *ensemble-b*.
2. Merge the 20-best lists to produce a single list of candidate translations.
3. Rescore the translations using the four individual systems.
4. Rerank the translations according to a (weighted) sum of the four length-normalized scores.

Table 4 summarizes the systems, weights, and length normalization values of the component systems used for reranking. The final score on the development set was 27.3.

6. Summary

We describe the work of the SRPOL and the University of Edinburgh collaboration on the English to Czech translation task for IWSLT 2019. Our final system is an ensemble of large transformer models trained with large amounts filtered parallel data and selected synthetic data.

7. References

[1] P. Williams, M. Chochowski, P. Przybysz, R. Sennrich, B. Haddow, and A. Birch, "Samsung and university of edinburgh's system for the IWSLT 2018 low resource MT task," in *Proceedings of the 15th International Workshop on Spoken Language Translation*, 2018, pp. 118–123.

[2] 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*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 116–121. [Online]. Available: https://www.aclweb.org/anthology/P18-4020

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 5998–6008.

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

[5] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, Jul. 2011, pp. 355–362. [Online]. Available: https://www.aclweb.org/anthology/D11-1033

[6] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 186–191. [Online]. Available: https://www.aclweb.org/anthology/W18-6319

[7] R. Bawden, N. Bogoychev, U. Germann, R. Grundkiewicz, F. Kirefu, A. V. Miceli Barone, and A. Birch, "The university of edinburghs submissions to the wmt19 news translation task," in *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, August 2019, pp. 103–115. [Online]. Available: http://www.aclweb.org/anthology/W19-5304

[8] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. [Online]. Available: https://www.aclweb.org/anthology/D18-2012

[9] L. J. Ba, R. Kiros, and G. E. Hinton, "Layer Normalization," *CoRR*, vol. abs/1607.06450, 2016.

[10] 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.

[11] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *The International Conference on Learning Representations*, San Diego, California, USA, 2015.

[12] S. L. Smith, P.-J. Kindermans, and Q. V. Le, "Don't decay the learning rate, increase the batch size," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1Yy1BxCZ

[13] D. Emelin, I. Titov, and R. Sennrich, "Widening the Representation Bottleneck in Neural Machine Translation with Lexical Shortcuts," in *Proceedings of the Fourth Conference on Machine Translation*. Florence, Italy: Association for Computational Linguistics, August 2019, pp. 102–115. [Online]. Available: http://www.aclweb.org/anthology/W19-5211

[14] M. Junczys-Dowmunt, "Dual conditional cross-entropy filtering of noisy parallel corpora," in *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 888–895. [Online]. Available: https://www.aclweb.org/anthology/W18-6478

[15] 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*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 35–40. [Online]. Available: https://www.aclweb.org/anthology/N16-1005

[16] C. Kobus, J. M. Crego, and J. Senellart, "Domain control for neural machine translation," *CoRR*, vol. abs/1612.06140, 2016. [Online]. Available: http://arxiv.org/abs/1612.06140