NICT's Neural and Statistical Machine Translation Systems for the WMT18 News Translation Task

Benjamin Marie Rui Wang
Atsushi Fujita Masao Utiyama Eiichiro Sumita
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Souraku-gun, Kyoto, 619-0289, Japan
{bmarie, wangrui, atsushi.fujita, mutiyama, eiichiro.sumita}@nict.go.jp

Abstract

This paper presents the NICT's participation to the WMT18 shared news translation task. We participated in the eight translation directions of four language pairs: Estonian-English, Finnish-English, Turkish-English and Chinese-English. For each translation direction, we prepared state-of-the-art statistical (SMT) and neural (NMT) machine translation systems. Our NMT systems were trained with the transformer architecture using the provided parallel data enlarged with a large quantity of back-translated monolingual data that we generated with a new incremental training framework. Our primary submissions to the task are the result of a simple combination of our SMT and NMT systems. Our systems are ranked first for the Estonian-English and Finnish-English language pairs (constraint) according to BLEU-cased.

1 Introduction

This paper describes the neural (NMT) and statistical machine translation systems (SMT) built for the participation of the National Institute of Information and Communications Technology (NICT) to the WMT18 shared News Translation Task (Bojar et al., 2018). We participated in four language pairs (eight translation directions): Estonian-English (Et-En), Finnish-English (Fi-En), Turkish-English (Tr-En), and Chinese-English (Zh-En). We chose these language pairs since they appear to be among the most challenging: involving distant languages and with less training data, for Finnish, Estonian, and Turkish, provided by the organizers than for Russian, German, and Czech. All our systems are *constrained*, i.e., we used only the parallel and monolingual data provided by the organizers to train and tune them. For all the translation directions, we trained NMT and SMT systems, and combined them through n-best list reranking using different informative features as proposed by Marie and Fujita (2018). This simple combination method, associated to the exploitation of large back-translated monolingual data, performed among the best MT systems at WMT18. Especially for the competitive Et-En and Fi-En translation tasks, for which our submissions are ranked first according to the BLEU-cased metric (henceforth BLEU). Our systems for Et-En, Fi-En, and Tr-En were trained using the exactly same procedures, without any specific linguistic treatments. On the other hand, for Zh-En, we used a specific tokenizer and used slightly different training parameters due to the much larger quantity of training data.

The remainder of this paper is organized as follows. In Section 2, we introduce the data preprocessing. In Section 3, we describe the details of our NMT and SMT systems. The back-translation of monolingual data using our new incremental training framework for NMT is described in Section 4. Then, the combination of NMT and SMT is described in Section 5. Empirical results produced with our systems are showed and analyzed in Section 6, and Section 7 concludes this paper.

2 Data Preprocessing

2.1 Data

As parallel data to train our systems, we used all the available data for all our targeted translation directions, except the "Wiki Headlines"[1] corpus for Fi-En. As English monolingual data, we used all the available data except the "Common Crawl" and "News Discussions" corpora.[2] For all other languages, we used all the available monolingual corpora, except for Turkish for which we

[1] It contains only very short segments that are not sentences and that we therefore assume to be of no use in NMT.

[2] The "News Crawl" data are sufficiently large and that these corpora are not in-domain monolingual data.

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 449–455
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64046

Language pair	#sent. pairs	#tokens	
Et-En	1.9M	29.4M (Et)	36.0M (En)
Fi-En	3.1M	52.9M (Fi)	72.8M (En)
Tr-En	207.4k	4.4M (Tr)	5.1M (En)
Zh-En	24.8M	509.9M (Zh)	576.2M (En)

Table 1: Statistics of our preprocessed parallel data.

Language	#lines	#tokens
En	338.7M	7.5B
Et	146.1M	3.6B
Fi	177.1M	3.2B
Tr	105.0M	1.8B
Zh	130.5M	2.3B

Table 2: Statistics of our preprocessed monolingual data.

used only 100 millions sentence pairs randomly extracted from "Common Crawl."

To tune/validate and evaluate our systems, we used Newstest2016 and Newstest2017 for Fi-En and Tr-En, Newsdev2017 and Newstest2017 for Zh-En, and Newsdev2018 for Et-En.

2.2 Tokenization, Truecasing and Cleaning

We used Moses tokenizer (Koehn et al., 2007) and truecaser for English, Estonian, Finnish, and Turkish. The truecaser was trained on one million tokenized lines extracted randomly from the monolingual data. Truecasing was then performed on all the tokenized data. For Chinese, we used Jieba[3] for tokenization but did not perform truecasing. For cleaning, we only applied the Moses script clean-n-corpus.perl to remove lines in the parallel data containing more than 80 tokens and replaced characters forbidden by Moses. Note that we did not perform any punctuation normalization. Tables 1 and 2 present the statistics of the parallel and monolingual data, respectively, after preprocessing.

3 MT Systems

3.1 NMT

To build competitive NMT systems, we chose to rely on the transformer architecture (Vaswani et al., 2017) since it has been shown to outperform, in quality and efficiency, the two other mainstream architectures for NMT known as deep recurrent neural network (deep RNN) and convolutional neural network (CNN). We chose Marian[4] (Junczys-Dowmunt et al., 2018) to train and evaluate our NMT systems since it supports state-of-the-art features and is one of the fastest NMT framework publicly available.[5] In order to limit the size of the vocabulary of the NMT models, we segmented tokens in the parallel data into subword units via byte pair encoding (BPE) (Sennrich et al., 2016b) using 50k operations. BPE segmentations were jointly learned on the training parallel data for source and target languages, except for Zh-En for which Chinese and English segmentations were trained separately. All our NMT systems for Et-En, Fi-En, and Tr-En were consistently trained on 4 GPUs,[6] with the following parameters for Marian: --type transformer --max-length 80 --mini-batch-fit --valid-freq 5000 --save-freq 5000 --workspace 8000 --disp-freq 500 --beam-size 12 --normalize 1 --valid-mini-batch 16 --overwrite --early-stopping 5 --cost-type ce-mean-words --valid-metrics ce-mean-words perplexity translation --keep-best --enc-depth 6 --dec-depth 6 --transformer-dropout 0.1 --learn-rate 0.0003 --dropout-src 0.1 --dropout-trg 0.1 --lr-warmup 16000 --lr-decay-inv-sqrt 16000 --lr-report --label-smoothing 0.1 --devices 0 1 2 3 --dim-vocabs 50000 50000 --optimizer-params 0.9 0.98 1e-09 --clip-norm 5 --sync-sgd --tied-embeddings --exponential-smoothing. For Zh-En, we did not use --dropout-src 0.1 --dropout-trg 0.1 since the training data is much larger. We performed NMT decoding with an ensemble of a total of six models according to the best BLEU (Papineni et al., 2002) and the best perplexity scores,[7] produced by three independent training runs.

[3] https://github.com/fxsjy/jieba

[4] https://marian-nmt.github.io/, version 1.4.0

[5] It is fully implemented in pure C++ and supports multi-GPU training.

[6] NVIDIA® Tesla® P100 16Gb.

[7] Note that the same model may give the best BLEU score and also the best perplexity score. Nonetheless, for consistency across language pairs, we systematically kept two models even if they were identical.

3.2 SMT

We also trained SMT systems using Moses. Word alignments and phrase tables were trained on the tokenized parallel data using mgiza. Source-to-target and target-to-source word alignments were symmetrized with the grow-diag-final-and heuristic. We trained hierarchical SMT models for Et-En and Fi-En since they provided better results than regular phrase-based models on our development data for these language pairs.[8] We also expected a similar observation for Tr-En and Zh-En. However, we were unable to exploit hierarchical models for the language pair Tr-En[9] while hierarchical models for the language pairs Zh-En were extremely large due to the size of our training data. Consequently, for Tr-En and Zh-En we simply trained regular phrase-based models using MSLR (monotone, swap, discontinuous-left, discontinuous-right) lexicalized reordering models and used the default distortion limit of 6. We trained two 4-gram language models: one on the entire monolingual data concatenated to the target side of the parallel data, and another one on the in-domain "News Crawl" corpora only, using LMPLZ (Heafield et al., 2013). For English, all singletons were pruned due to the large size of the monolingual data. To tune the SMT model weights, we used KB-MIRA (Cherry and Foster, 2012) and selected the weights giving the best BLEU score on the development data after 15 decoding runs.

4 Back-translation of Monolingual Data

4.1 Incremental Back-Translation with Et-En, Fi-En, and Tr-En

We introduced an incremental training framework for NMT aiming to iteratively increase the quality and quantity of the synthetic parallel data used for training. In this framework, we first simultaneously but independently train a source-to-target and a target-to-source NMT systems using the same original parallel data. Then, we back-translate source and target monolingual data respectively using the source-to-target and the target-to-source NMT systems, and obtain two sets of synthetic parallel data. And then, a new source-to-target and a new target-to-source NMT systems are trained, from scratch, on their respective new training data comprising the mixture of the original parallel data and the synthetic parallel data whose source side is back-translated from the target side. At this stage, we just do what is usually done by previous work (Sennrich et al., 2016a).

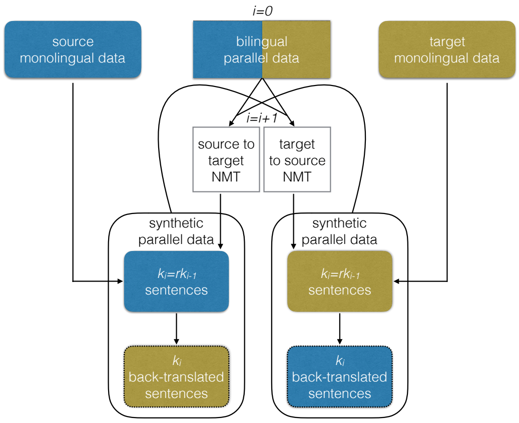

Figure 1: Our incremental training framework.

As illustrated in Figure 1, we continue this procedure iteratively. Using source-to-target and target-to-source NMT systems trained on the mixture of the synthetic and original parallel data, we back-translate a larger number of monolingual sentences, including the same sentences back-translated at the first iteration. Since we have better NMT systems than those at the first iteration, we can expect the back-translation to be of a better quality. We mix this new synthetic parallel data to the original one and train again from scratch a source-to-target and a target-to-source NMT systems to obtain further improved translation models. Note that this procedure is partially similar to the work proposed by Zhang et al. (2018) and Hoang et al. (2018), but differs in the sense that we increase incrementally our back-translated data.

Given the number of sentences used in the first iteration, k_1, and an expansion factor, r, we determine k_i, the number of monolingual sentences back-translated at iteration i, as follows:

$$k_i = rk_{i-1} \qquad (1)$$

The parameters used for the given language pairs are listed in Table 3. The monolingual sentences to be back-translated were randomly extracted from the NewsCrawl corpora. For Et-En and Fi-En, we stopped the incremental training after 2 iterations, back-translating up to 2M sentences. For Tr-En, we observed improvements for

[8] Between 0.5 and 1 BLEU points of improvement.

[9] Moses consistently crashed (*segmentation fault*) during the decoding of the development data.

Language pair	k_1	r	#iter. (total)
Et-En	1M	2	2
Fi-En	1M	2	2
Tr-En	200k	2	4

Table 3: Parameters used for our incremental training. For each language pair, the same parameters were used for both translation directions. In our preliminary experiments, we found that setting $r = 2$ and k_1 very close to, or smaller than, the size of the original parallel data consistently gives good results across language pairs. Fine-tuning r and k_1 would result in a better translation quality but at a greater cost.

both translation directions until the fourth iteration that back-translated 1.6M sentences (approximately 8 times the size of the original parallel data). In our preliminary experiments, we found that incremental training significantly improves the translation quality over an NMT system that was trained directly, on the same amount of back-translated sentences. For instance, we observed a 0.6 BLEU points improvements for Tr→En over a system trained on 1.6M sentences back-translated by a system trained on the original parallel data (as in (Sennrich et al., 2016a)).

4.2 Setting for Zh-En

For the Zh-En language pair, since much larger parallel data were provided to train the system, we did not perform the incremental back-translation described in Section 4.1. For En→Zh, we back-translated the entire XMU Chinese monolingual corpus containing 5.4M sentences as the source to produce synthetic English data. For Zh→En, we empirically compared the impact of back-translating different sizes of English monolingual data, using the first 10M, 20M, and 40M lines of the concatenation of News Crawl-2016 and News Crawl-2017 English corpora to produce synthetic Chinese data. As shown in Table 4, there is not a significant difference in exploiting back-translated data as large as 40M lines compared to only 10M lines. Therefore, we selected the first 10M lines of the News Crawl-2016 English corpus to produce synthetic Chinese data.

5 Combination of NMT and SMT

Although we can expect SMT to perform very poorly for all the language pairs we considered,[10] our primary submissions for WMT18 are the results of a simple combination of NMT and SMT. Indeed, as demonstrated by Marie and Fujita (2018), and despite the simplicity of the method used, combining NMT and SMT makes MT more robust and can significantly improve translation quality, even when SMT greatly underperforms NMT. Following Marie and Fujita (2018), our combination of NMT and SMT works as follows.

#lines back-translated	#BLEU
10M	21.4
20M	21.4
40M	21.5

Table 4: Results for different sizes of back-translated data for the Zh→En translation direction on News-dev2017.

5.1 Generation of n-best Lists

We first produced the 100-best translation hypotheses with our NMT and SMT systems, independently.[11] Unlike `Moses`, `Marian` must use a beam of size k to produce a k-best list during decoding. However, using a larger beam size during decoding for NMT may worsen translation quality (Koehn and Knowles, 2017).[12] Consequently, we also produced with `Marian` the 10-best lists, for Zh-En, and 12-best lists for the other language pairs, and merged them with `Marian`'s 100-best lists to obtain lists containing up to 110 or 112 hypotheses.[13] In this way, we make sure that we still have hypotheses of good quality in the lists despite using a larger beam size.[14] Then, we merged the lists produced by `Marian` and `Moses`. We rescored all the hypotheses in the resulting lists with a reranking framework using features to better model the fluency and the adequacy of each hy-

[10] Especially due to the rich morphology of the languages involved and the long distance reorderings to perform in order to produce a translation of good quality.

[11] We used the option `distinct` in `Moses` to avoid duplicated hypotheses, i.e., with the same content but obtained from different word alignments, and consequently to increase diversity in the generated n-best lists.

[12] For Zh-En, the decoding of the test data with k=100 resulted in a drop of 0.4 BLEU points compared to a decoding with k=10. However, for the other language pairs we did not observe such a quality drop but instead a consistent and slight improvement of BLEU scores.

[13] Note that we did not remove duplicated hypotheses that may appear, for instance, in both 10-best and 100-best lists.

[14] Note that we could have also generated many individual smaller n-best lists, for instance using all our NMT models independently, and merge them to increase the diversity of the hypotheses list to rerank and therefore obtained better results. However, we decided to leave the exploration of this possibility for feature work.

Feature	Description
L2R (6)	Scores given by each of the 6 left-to-right Marian models
R2L (2)	Scores given by each of the 2 (or 4 for Tr-En) right-to-left Marian models
LEX (4)	Sentence-level translation probabilities, for both translation directions
LM (2)	Scores given by the two language models used by the Moses baseline systems
WPP (2)	Averaged word posterior probability
LEN (2)	Difference between the length of the source sentence and the length of the translation hypothesis, and its absolute value
SYS (1)	System flag, 1 if the hypothesis comes from Moses n-best list or 0 otherwise
MBR (2)	For Tr-En only: MBR decoding using sBLEU and chrF++
PBFD (1)	For Tr-En only: The phrase-based forced decoding score
L2R-bwd (6)	Scores given by each of the 6 left-to-right Marian models for the backward translation direction
R2L-bwd (2)	Scores given by each of the 2 (or 4 for Tr-En) right-to-left Marian models for the backward translation direction

Table 5: Set of features used by our reranking systems. The column “Feature” refers to the same feature name used in Marie and Fujita (2018). Note that the two last feature sets, “L2R-bwd” and “R2L-bwd,” were not experimented in Marie and Fujita (2018). The numbers between parentheses indicate the number of scores in each feature set.

#	System	Et→En	En→Et	Fi→En	En→Fi	Tr→En	En→Tr	Zh→En	En→Zh
1.	Moses	18.2	15.1	15.8	10.7	12.1	8.4	16.9	28.0
2.	Moses NMT-reranked	20.2	17.6	17.5	12.2	14.2	10.1	19.0	29.9
3.	Marian single (w/o backtr)	22.9	18.5	17.6	13.2	20.2	12.2	23.7	33.0
4.	Marian single (w/ backtr)	28.6	24.0	23.1	16.8	25.2	18.0	24.7	37.2
5.	Marian ensemble (w/ backtr)	29.1	24.3	23.6	17.3	25.8	18.3	25.9	37.9
6	Moses + Marian	30.7	25.2	24.9	18.2	26.9	19.2	26.7	39.7

Table 6: Detokenized BLEU-cased scores for our MT systems on the *Newstest2018* test set. “NMT-reranked” denotes the reranking of the Moses’s 100-best hypotheses using all our NMT models (left-to-right and right-to-left, for both translation directions, trained with back-translated data) as features. “backtr” denotes the use or not of back-translated monolingual data. “Moses + Marian” denotes our combination of best NMT (#5) and SMT (#1) systems described in Section 5.

pothesis. This method can find a better hypothesis in these merged n-best lists than the one-best hypothesis originated by either Moses or Marian.

5.2 Reranking Framework and Features

We chose KB-MIRA as a rescoring framework and used a subset of the features proposed in Marie and Fujita (2018). As listed in Table 5, it includes the scores given by the 6 left-to-right NMT models used to perform ensemble decoding (see Section 3.1). We also used as features the scores given by right-to-left NMT models that we trained for each translation direction with the same parameters as left-to-right NMT models. The two right-to-left NMT models, each achieving the best BLEU and the best perplexity scores on the development data, were selected, giving us two other features for each translation direction. Since the Tr-En training parallel data are much smaller, we were able to perform one more right-to-left training run for Tr→En and En→Tr.[15] We also experimented with the use of the scores computed from the NMT models trained for the backward translation direction. In total, we have then 16 features, or 20 for Tr-En, computed from NMT models. All the following features we used are described in details by Marie and Fujita (2018). We computed sentence-level translation probabilities using the lexical translation probabilities learned by mgiza during the training of our SMT systems. The two language models trained for SMT for each translation direction were also used to score the n-best translation hypotheses. To account for hypotheses length, we added the difference, and its absolute value, between the number of tokens in the translation hypothesis and the source sentence. As a consensus-based feature, we used the word posterior probabilities.

For only the Tr-En language pair, we were also able to compute a phrase-based forced decoding

[15] In practice, adding one more right-to-left model for reranking did not significantly improve the BLEU score on the development data.

score (Zhang et al., 2017) thanks to the small size of the phrase table learned for this language pair. Also only for this language pair, we computed the scores for each hypothesis given by the so-called minimum Bayes risk (MBR) decoding for n-best list using two metrics: sBLEU and chrF++ (Popović, 2017).

The reranking framework was trained on n-best lists produced by the decoding of the same development data that we used to validate NMT system's training and to tune SMT's model weights.

6 Results

The results of our systems computed for the Newstest2018 test set are presented by Table 6.

As expected, SMT systems greatly underperformed our best NMT systems with differences in BLEU points ranging from 6.6 (En→Fi) to 13.7 (Tr→En). Reranking `Moses` 100-best hypotheses using NMT models (NMT-reranked) significantly improved the translation quality for all the translation directions. For Fi→En, `Moses` NMT-reranked performed only 0.1 BLEU points worse than `Marian` single (w/o backtr). This result demonstrates the ability of SMT in producing better translation hypotheses than its one-best hypothesis. Indeed, a better translation can be easily retrieved with the help of NMT models within the 100-best lists. Using back-translated data during training was very effective for Et-En, Fi-En, and Tr-En, with improvements ranging from 3.6 to 5.8 BLEU points. Improvements were less significant for Zh-En, especially for Zh→En with only 1.0 BLEU points of improvements. This may be explained by the much larger parallel data already used to train systems for Zh-En. Another interesting finding is the relative inefficiency of using an ensemble of 3 models for NMT decoding with the transformer architecture over using a single model, as opposed to what was reported by most participants at WMT17 (Bojar et al., 2017) using RNN. For instance, for En→Et and En→Tr ensemble decoding improved the translation quality by only 0.3 BLEU points.

Our combination of SMT and NMT significantly outperformed all our NMT systems for all translation directions. For instance, this combination brought 1.6 and 1.8 BLEU points of improvements for Et→En and En→Zh, respectively, over our best NMT systems.

7 Conclusion

We participated in eight translation directions and for all of them we did experiments to compare SMT and NMT performances. While SMT significantly underperforms NMT, we showed that a simple combination of both approaches delivers the best results.

Acknowledgments

We would like to thank all the reviewers for their valuable comments and suggestions. This work was conducted under the program "Promotion of Global Communications Plan: Research, Development, and Social Demonstration of Multilingual Speech Translation Technology" of the Ministry of Internal Affairs and Communications (MIC), Japan.

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. 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.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada. Association for Computational Linguistics.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria. Association for Computational Linguistics.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine*

Translation and Generation, pages 18–24. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Benjamin Marie and Atsushi Fujita. 2018. A smorgasbord of features to combine phrase-based and neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 111–124. Association for Machine Translation in the Americas.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. 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)*, pages 86–96. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2017. 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)*, pages 152–162, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 555–562. Association for the Advancement of Artificial Intelligence.