Combination of Neural Machine Translation Systems at WMT20

Benjamin Marie **Raphael Rubino** **Atsushi Fujita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
{bmarie, raphael.rubino, atsushi.fujita}@nict.go.jp

Abstract

This paper presents neural machine translation systems and their combination built for the WMT20 English↔Polish and Japanese→English translation tasks. We show that using a Transformer Big architecture, additional training data synthesized from monolingual data, and combining many NMT systems through n-best list reranking improve translation quality. However, while we observed such improvements on the validation data, we did not observe similar improvements on the test data. Our analysis reveals that the presence of translationese texts in the validation data led us to take decisions in building NMT systems that were not optimal to obtain the best results on the test data.

1 Introduction

This paper describes the neural machine translation (NMT) systems and their combination built for a participation of the National Institute of Information and Communications Technology (NICT) in the WMT20 shared News Translation Task.[1] We participated in three translation directions: Japanese→English (Ja→En), English→Polish (En→Pl), and Polish→English (Pl→En). All our systems are *constrained*, i.e., we used only the parallel and monolingual data provided by the organizers to train and tune them, and validated/selected our best systems exclusively using the official validation data provided by the organizers. We trained NMT systems with several different frameworks and architectures, and combined them, for each translation direction, through n-best list reranking using informative features as proposed by Marie and Fujita (2018). This simple combination method, associated with the exploitation of large tagged back-translated monolingual data, improved BLEU scores on the official validation data provided by the organizers. However, we did not observe these improvements on the test data for which our baseline systems remained the best. While we have rigorously selected our systems according to their performance on the validation data, the analysis of our results reveal how easily we would have been able to achieve BLEU scores among the best submissions by choosing/selecting our best systems according to their performance on the test data, as encouraged by the WMT submission process (Section 2).

The remainder of this paper is organized as follows. In Section 2, we briefly describe the WMT20 translation task. In Section 3, we introduce the data pre-processing and cleaning. In Section 4, we describe the details of our NMT systems' architectures and frameworks. In Section 5, we describe two different strategies that we used to augment the training parallel data of our systems: parallel data extraction from monolingual data and backward/forward translations. Then, the combination of our NMT systems is described in Section 6. Empirical results produced with our systems on the validation and test data are presented in Section 7. We propose an analysis in Section 8 to better understand why our best systems on the validation data are significantly worse on the test data. Section 9 concludes this paper.

2 Description of the Task

The task is to translate texts in the news domain. For this purpose, news articles were sampled from online newspapers from September–November 2019. The sources of the test data are original texts whereas the targets are human-produced translations, i.e., participants are not asked to translate translationese texts unlike past WMT translation tasks. Although organizers also mentioned that the provided validation data were

[1] The team ID of our participation is "NICT_Kyoto".

Proceedings of the 5th Conference on Machine Translation (WMT), pages 230–238
Online, November 19–20, 2020. ©2020 Association for Computational Linguistics

created in the same way as the test data, they were actually made half of translationese texts and half of original texts. For the Inuktitut→English translation task, source texts to translate in the test data were only translationese texts.

Training parallel and monolingual data were provided for all language pairs. Participants were asked to mention whether they used additional external data. We chose to participate in the *constrained* settings using only the provided data to train our MT systems. Validation data were also provided for each language pair. We used the entire data to keep it sufficiently large for validation purposes, even though half of it was made up of translationese texts.

For collecting submissions, organizers relied on a new framework: `Ocelot`.[2] Each participant was allowed to submit up to 8 submissions per account but was not limited in the number of accounts. Upon submission, `Ocelot` shows the corresponding chrF and BLEU scores computed using reference translations that were not released during the competition. Participants could then rely on these scores to select and validate their best system on the test data.[3] We chose to ignore these scores obtained on the test data to remain in a much more realistic scenario where we do not have access to reference translations, i.e., we relied only on the validation data to select our primary submission.

Primary submissions selected by the participants were then evaluated by humans which is the official evaluation for WMT translation tasks.

3 Data Pre-processing and Cleaning

3.1 Data

As parallel data to train our systems, we used all the provided data for all our targeted translation directions, except the "Wiki Titles"[4] corpus. As English monolingual data, we used all the provided data, but sampled only 200M lines from the "Common Crawl" corpora, except the "News Discussions" and "Wiki Dumps" corpora. For all other languages, we used all the provided monolingual corpora but also sampled only 200M lines from the "Common Crawl" corpora.

Language pair	#sent. pairs	#tokens	
En–Pl	8.7M	239.5M (En)	310.0M (Pl)
En–Ja	15.2M	394.5M (En)	380.6M (Ja)

Table 1: Statistics of our pre-processed parallel data.

To tune/validate and evaluate our systems, we used the official validation and test data provided by the organizers.

3.2 Pre-processing and Cleaning

Since some corpora were crawled from the Web and therefore potentially very noisy, we first performed language identification on all the data to keep only lines that have a high probability of being in the right language. We used `fastText` (Bojanowski et al., 2017) and its large model for language identification.[5] We only retained sentences that have a probability higher than 0.75 to be in the right language. For the parallel data, if at least one side of each sentence pair did not match this criteria, we removed the pair from the corpus.

We used Moses (Koehn et al., 2007) punctuation normalizer, tokenizer, and truecaser for English and Polish. The truecaser was trained on the News Crawl 2019 corpora. Truecasing was then performed on all the tokenized data. Then, for the Pl–En language pair, we jointly learned 32k BPE operations (Sennrich et al., 2016b) on the concatenation of English and Polish News Crawl 2019 corpora. We performed sub-word segmentation using this vocabulary on the Polish and English parallel and monolingual data. For the Ja–En language pair, we independently learned 32k BPE operations on the English News Crawl 2019 corpus for English, 32k sentence piece (Kudo and Richardson, 2018) operations on the Japanese News Crawl 2019 corpus for Japanese, and then applied the operations to perform sub-word segmentation on the data in their respective language.

For further cleaning of the data, we applied the script "clean-corpus-n.perl" from Moses to remove empty lines and sentences longer than 120 sub-word tokens. Tables 1 and 2 present the statistics of the parallel and monolingual data, respectively, after pre-processing.

[2] `https://ocelot.mteval.org/`

[3] We can read in the "competition updates" that this behavior was encouraged by the organizers: "Also added chrF computation to give you more data points for your primary submission selection. Submissions remain ordered by decreasing SacreBLEU score."

[4] It contains only very short segments that are not sentences. We therefore assume to be of limited use in NMT.

[5] `https://fasttext.cc/docs/en/language-identification.html`

Language	#lines	#tokens
En (En–Pl)	328M	7.9B
En (En–Ja)	328M	7.7B
Ja	184M	4.8B
Pl	137M	3.2B

Table 2: Statistics of our pre-processed monolingual data.

4 NMT systems

4.1 Architectures

Transformer Base and Big For our NMT systems, we chose the Transformer architecture (Vaswani et al., 2017). In this paper, we refer to Transformer Base and Big as the "*base*" and "*big*" configurations from Vaswani et al. (2017)'s paper. The architecture differences are as follows:

- Base: 512 embedding dimensions, 2,048 dimensions for the feed-forward, and 8 heads
- Big: 1024 embedding dimensions, 4,096 dimensions for the feed-forward, and 16 heads

Highway Transformer Residual connections (RCs) (Srivastava et al., 2015a; He et al., 2016) have been shown to increase forward and backward information flow in deep neural networks (Hardt and Ma, 2017) and thus are a crucial component of the Transformer architecture. Removing them has a negative impact on training and on the overall performances of the resulting model (Bapna et al., 2018). However, incorporating RCs through the addition operation as it is commonly done in the Transformer network does not allow for a distribution of weights between carrying or transforming the input. An alternative, inspired by the *Highway Network* (Srivastava et al., 2015b) and implemented within the Transformer by Chai et al. (2020), includes a trainable gating mechanism that regulates the information flow. We applied a few modifications to the implementation proposed in Chai et al. (2020): removing all layer normalization operations, adding depth-aware parameter initialization (Junczys-Dowmunt, 2019; Zhang et al., 2019), and initializing biases so that the residual blocks are initially forced to carry information rather than transforming it (Srivastava et al., 2015b).

4.2 Frameworks and Settings

`Marian` Our Models trained with the `Marian` toolkit (Junczys-Dowmunt et al., 2018) were only based on Transformer *Base*. We set the dropout at 0.1 and used the mini-batch-fit option of `Marian` to have batches as large as allowed by the size of the GPU memory. We used ReLU activation functions and optimized the models using the Adam optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e^{-9}$, a learning rate initialized at $3e^{-4}$, following a linear warm-up during 16k updates and decaying based on the inverse square root of the update number. Label smoothing was set to 0.1. During training, mean cross-entropy was evaluated on the entire validation data every 5,000 mini-batch updates and training was stopped after 5 consecutive times without an improvement of the mean cross-entropy. Then, we selected the best model that yielded the best BLEU score on the validation data. For decoding, we fixed the beam size at 12 and the length normalization at 1.0.

`Fairseq` Models trained with the `Fairseq` toolkit were based on Transformer *base* and Transformer *big*. The former used a dropout rate of 0.1 and batches containing approximately 12k tokens with parameters updated every 2 batches. The latter used a dropout rate of 0.3 and batches containing approximately 8k tokens with parameters updated every 8 batches. Both configurations shared decoder input and output embeddings, trained with half-precision float numbers, used ReLU activation functions, were optimized using the Adam optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e^{-9}$, a learning rate initialized at $1.7e^{-7}$, following a linear warm-up during 4k updates until reaching $5e^{-4}$ and decaying based on the inverse square root of the update number. Label smoothing with a parameter 0.1 was applied during training. Whereas *base* models were trained for 200 epochs, *big* models were trained for 100 epochs. The entire validation data was used for evaluation every epoch, while the best BLEU scores on this data allow for checkpoint saving. The parameters for decoding were fixed: a beam size of 4 and a length penalty of 0.6.

5 Training Data Augmentation

5.1 Parallel Data Alignment

We extracted additional training parallel data from the News Crawl monolingual corpora with the following procedure:

1. Jointly train bilingual word embeddings on

Configuration	En–Pl En→Pl	 Pl→En	 #sent. pairs	Ja→En	#sent. pairs
w/o NC	26.3	30.4	0	20.3	0
w/ NC	26.4	30.6	257.4k	20.3	244.1k

Table 3: Results obtained on the validation data with `Fairseq` Big with and without using the additional parallel data extracted from the News Crawl monolingual corpora, denoted "NC." The columns "#sent. pairs" indicate how many sentence pairs were extracted from the News Crawl monolingual corpora.

the provided parallel data using `Bivec` (Luong et al., 2015).

2. Make all possible bilingual sentence pairs from News Crawl corpora in the source and target languages.

3. For each sentence pair, compute the similarity between the source and target sentences using the bilingual word embeddings trained with `Bivec` simply by measuring cosine similarity over the averaged word embeddings in each sentence as proposed by Artetxe and Schwenk (2019).[6]

4. Finally, keep only the sentence pairs with a score higher than a threshold among $\{1.0, 1.0025, 1.05\}$ and select the value that results in the sentence pairs leading to the highest BLEU score on the validation data when mixing the selected sentence pairs with the original parallel data for training NMT.

Table 3 gives an overview of the results obtained with the additional sentence pairs extracted from News Crawl. We did not observe significant improvements as we could only extract a very small amount of useful sentence pairs. Nevertheless, we decided to keep these additional data to train our other NMT systems, since it did not appear harmful according to BLEU. However, as we report in Section 8, it was not the optimal choice to obtain the best results on the test data.

5.2 Backward and Forward Translation of Monolingual Data

Parallel data for training NMT can be augmented with synthetic parallel data, generated through a so-called back(ward)-translation, to significantly improve translation quality (Bertoldi and Federico, 2009; Bojar and Tamchyna, 2011; Sennrich et al., 2016a). We used the `Fairseq` Big system, trained on the provided parallel data and the aligned News Crawl sentence pairs, to translate target monolingual sentences into the source language. Then, these back-translated sentences were simply mixed with the original parallel data, putting the synthetic side on the source side, to train from scratch a new NMT system.

We also experimented with forward translation, i.e., with the synthetic part on the target side, and tagged back-translation (Caswell et al., 2019), which simply adds a tag at the beginning of each back-translation, as it has shown to lead to better results, especially when translating texts in their original language (Marie et al., 2020).

For English, we translated 50M sentences made of the entire News Crawl 2019 corpus and randomly added sentences from News Crawl 2018 corpus until we have 50M sentences. For Polish and Japanese, we translated the entire News Crawl corpora and added sentences from the Common Crawl corpus until we have 50M sentences.

For each configuration, i.e., back-translation, tagged back-translation, and forward translation, we also experimented with sub-samples of 12.5M (only with `Marian`), and 25M synthetic sentence pairs, in addition to using the entire 50M sentence pairs, for retraining the NMT systems. Table 4 gives an overview of the results for each configuration obtained on the validation data.

All configurations using back-translations (BT) and tagged back-translations (TBT) were better than the baseline system as expected. We also observed very small differences in BLEU when increasing the size of the back-translated data.

TBT improves over BT as expected (Caswell et al., 2019), but only for Pl→En and Ja→En. On the other hand, using forward translations significantly decreased BLEU scores, as expected, since it introduces NMT translations to the target side of the training data (Bogoychev and Sennrich, 2019), but again only for Pl→En and Ja→En. Our results for En→Pl across all configurations remained similar, which defies the findings of previous work on back-translation and forward translation. We give

[6] We used the "Ratio" version of the scoring function.

System	#sent. pairs	En→Pl	Pl→En	Ja→En
Marian Base	0	24.4	29.1	18.1
BT	12.5M	26.1	32.1	21.1
	25M	26.2	32.1	21.2
	50M	26.3	31.7	21.1
TBT	12.5M	26.1	30.3	21.1
	25M	26.1	32.3	21.2
	50M	26.3	32.2	21.4
FWD	12.5M	26.3	29.7	18.3
	25M	26.4	29.5	17.4
	50M	26.3	29.6	16.4

Table 4: Results of Marian Base on the validation data obtained using synthetic parallel data as back-translations (BT), tagged back-translations (TBT) or forward translations (FWD).

Feature	Description
NMT models	Scores given by each NMT model
LEX	Sentence-level translation probabilities, for both translation directions
LM	Scores given by a 4-gram language model trained on all the monolingual corpora in the target language
LEN	Difference between the length of the source sentence and the length of the translation hypothesis, and its absolute value

Table 5: Set of features used by our reranking systems. The "Feature" column refers to the same feature used in Marie and Fujita (2018). The numbers between parentheses indicate the number of scores in each feature set.

some plausible explanations for this peculiarity in our analysis in Section 8.

6 Combination of NMT systems

Our primary submissions for the tasks were the result of a simple combination of all our NMT systems through n-best list reranking. As demonstrated by Marie and Fujita (2018), it can significantly improve translation quality, even when there is a large difference in translation quality between the combined systems. Following Marie and Fujita (2018), our system combination works as follows.

6.1 Generation of n-best Lists

We first independently generated the 100-best translation hypotheses from each of all our NMT models, and additional 12-best, with Marian, or 4-best with Fairseq. We then merged all these lists generated by different systems, without removing duplicated hypotheses.

6.2 Reranking Framework and Features

We rescored all the hypotheses in the list with a reranking framework using features to better model the fluency and the adequacy of each hypothesis. This method can find a better hypothesis in these merged n-best lists than the one-best hypothesis originated by the individual systems. We chose KB-MIRA (Cherry and Foster, 2012) as a rescoring framework and used a subset of the features proposed in Marie and Fujita (2018). All the following features we used are described in detail by Marie and Fujita (2018). As listed in Table 5, it includes all scores given by all our NMT models. We computed sentence-level translation probabilities using the lexical translation probabilities learned by mgiza on all the parallel training data of our NMT systems. One 4-gram language model trained on the target language model was also used. 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. The reranking framework was trained on n-best lists generated by decoding the entire validation data.

7 Results

Our main results are presented in the Table 6. According to the validation data, Fairseq Base is as good as, or better than, Marian Base. Given this observation, we trained Fairseq Big and obtained even better results on the validation data. BLEU scores are improved by up to 4.1 BLEU points when using tagged back-translations (TBT) on the validation data. Overall, the best system is, as expected, the Reranker combining all our systems with additional features. How-

System	En→Pl		Pl→En		Ja→En	
	Validation	Test	Validation	Test	Validation	Test
Marian Base	24.4	21.2	29.1	31.9	18.1	19.1
Fairseq Base	24.4	21.8	30.3	32.4	19.4	18.7
Fairseq Big	26.4	21.9	30.9	31.5	20.4	19.3
Fairseq Big TBT	28.5	23.1	35.0	31.8	23.3	19.9
Reranker*	29.9	24.9	36.5	32.3	25.5	22.8

Table 6: Results of our main systems. Marian Base and Fairseq Base use the same training data and architecture. Fairseq Big uses Transformer Big and the additional training data extracted from News Crawl corpora. Fairseq Big TBT is retrained from scratch on the tagged back-translations generated by Fairseq Big. The system denoted with an "*" is our primary system.

ever, surprisingly, the results on the test data exhibited a significantly different pattern. For instance, Marian Base performed very closely to Fairseq Big on the test data. Even more strikingly, we observed only small differences in BLEU between Fairseq Big and Fairseq TBT. For instance, for Pl→En, while we have 4.1 BLEU points of improvements on the validation data, we have only 0.3 BLEU points of improvements on the test data. Also for this translation direction, Reranker outperforms Marian Base by 7.4 BLEU points on the validation data but by only 0.4 BLEU points on the test data.

To better understand the lack of correlation between our results on validation and test data, we propose an analysis in the next section.

8 Analysis

Table 7 presents all our results for En↔Pl on the validation data, separating the part in the original and non-original languages, and the test data.

One obvious observation from these results is that the BLEU scores on the test data are all very close to the score of the validation data in original language (Orig.). On the other hand, we also observe that the ranking of the systems given the BLEU score on the entire validation data does not correlate well with the ranking of the systems given the BLEU score on the validation data in the original language. It means that the translationese texts in the validation data had a negative impact on all our decisions for selecting the best framework, architecture, additional parallel sentences, and so forth, and that we could potentially had better results by taking our decisions by using only the original texts in the validation data.

Translationese texts are particularly harmful for training a Reranker, as we can observe for Pl→En. Using them as training data for the Reranker leads to significantly lower BLEU scores (#14) while training it only on the original texts of the validation data leads to our best BLEU score (#15). For this translation direction, we also observe large improvements of BLEU thanks to back-translations on the validation data that comes mainly from the translationese texts while translation quality drops when translating the original texts in the validation and test data, as expected. This was compensated by using tagged back-translations (#10-12) as suggested by Marie et al. (2020).

Our observations are very different for the reverse translation direction. For En→Pl, training Reranker on the entire validation data leads to the best BLEU score, and it drops only slightly when training only on the translationese texts. Even more surprisingly, using back-translations improves BLEU scores for both original and translationese texts while using tagged back-translations (#12) leads to BLEU scores identical to those obtained by using back-translations (#9) for the original texts. These peculiarities observed for Pl→En, associated with our observations in Section 5.2 that forward translations improves BLEU, are in contradiction with the findings in previous work as follows.

- Back-translations should decrease BLEU scores for original texts (Edunov et al., 2020).
- Tagged back-translations should improve BLEU scores for original texts (Marie et al., 2020).
- Forward translation should lead to lower BLEU scores for translationese texts (Bogoychev and Sennrich, 2019).

A possible explanation is that the texts denoted as "original" in the validation data and the test data,

#	System	Arch.	NC	BT	TBT	En→Pl Validation Orig.	En→Pl Validation Non-orig.	En→Pl Validation All	En→Pl Test	Pl→En Validation Orig.	Pl→En Validation Non-orig.	Pl→En Validation All	Pl→En Test
						Individual Systems							
1	Marian	Base				21.4	28.7	24.4	21.2	32.0	26.7	29.1	32.3
2	Marian	Base	✓			21.5	28.5	24.8	21.7	32.3	27.5	29.7	31.9
3	Fairseq	Base				20.9	28.7	24.4	21.8	33.1	27.9	30.3	32.4
4	Fairseq	Highway				21.1	29.6	25.0	21.8	32.6	28.5	30.6	32.6
5	Fairseq	Big				22.7	30.7	26.3	22.3	31.2	30.3	30.7	32.5
6	Fairseq	Big	✓			22.6	31.2	26.4	21.9	31.9	29.5	30.9	31.5
7	Marian	Big	✓	✓		22.1	31.3	26.3	21.9	29.1	34.1	32.0	29.5
8	Fairseq	Base	✓	✓		22.2	32.1	26.8	22.2	29.7	34.9	32.9	29.7
9	Fairseq	Big	✓	✓		23.6	34.4	28.5	23.7	30.2	36.5	33.9	29.5
10	Marian	Big	✓		✓	22.1	31.1	26.2	22.1	32.1	32.9	32.5	31.8
11	Fairseq	Base	✓		✓	22.3	32.0	26.7	22.2	31.8	34.0	33.2	31.7
12	Fairseq	Big	✓		✓	23.6	34.6	28.5	23.1	32.4	37.1	35.0	31.8
						System Combination							
13	Reranker*					25.6	35.2	29.9	24.9	33.1	38.7	36.5	32.3
14	Reranker	Non-orig.				23.2	35.3	29.3	23.1	28.8	39.6	34.7	28.4
15	Reranker	Orig.				25.4	33.9	29.1	24.7	35.0	33.8	34.4	34.8

Table 7: The system denoted with an "∗" is our primary system. The column "Arch." stands for the Transformer architecture, "NC" indicates the use of the News Commentary Corpus, "BT" and "TBT" indicate the use of back-translation and tagged back-translation, respectively. "Reranker Non-orig." and "Reranker Orig." are variants of Reranker that are trained on the validation data using only the part in the non-original and original languages, respectively, while Reranker, our primary system, was trained on the entire validation data.

that were prepared similarly, do have some characteristics of translationese that may come from the translation of texts not in their original language or the use of MT followed by post-editing. Comparing our results with the results of other participants will help us test this assumption.

9 Conclusion

We participated in three translation directions and for all of them we did experiments with several frameworks and architectures, also exploiting additional synthetic parallel data made from monolingual data. Combining all our systems led to significantly better BLEU scores on the validation data. However, our analysis revealed that the presence of translationese texts in the validation data led us to take sub-optimal choices that prevented us from obtaining significantly better BLEU scores on the test data. Selecting/validating systems on the test data should not be possible, or at least not an option. We thus suggest organizers to provide validation data that better matches the characteristics of the test data, e.g., removing translationese texts if none are in the test data.

Acknowledgments

We would like to thank Reviewer 1 for the detailed corrections and suggestions.

References

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3033, Brussels, Belgium. Association for Computational Linguistics.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece. Association for Computational Linguistics.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *arXiv preprint arXiv:1911.03362*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine*

Translation, pages 330–336, Edinburgh, Scotland. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Yekun Chai, Shuo Jin, and Xinwen Hou. 2020. Highway transformer: Self-gating enhanced self-attentive networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6887–6900, Online. 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.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, Online. Association for Computational Linguistics.

Moritz Hardt and Tengyu Ma. 2017. Identity matters in deep learning. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, USA.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. 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, Melbourne, Australia. 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, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. 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*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, USA. 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, Boston, USA. Association for Machine Translation in the Americas.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Tagged back-translation revisited: Why does it really work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5990–5997, Online. 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, Berlin, Germany. 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.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015a. Training very deep networks. In *Advances in Neural Information Processing Systems 28*, pages 2377–2385, Montréal, Canada. Curran Associates, Inc.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015b. Highway Networks. *arXiv preprint arXiv:1505.00387*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, Long Beach, USA. Curran Associates, Inc.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China. Association for Computational Linguistics.