Copied Monolingual Data Improves Low-Resource Neural Machine Translation

Anna Currey, Antonio Valerio Miceli Barone, and **Kenneth Heafield**
School of Informatics, University of Edinburgh
a.currey@sms.ed.ac.uk
{amiceli, kheafiel}@inf.ed.ac.uk

Abstract

We train a neural machine translation (NMT) system to both translate source-language text and copy target-language text, thereby exploiting monolingual corpora in the target language. Specifically, we create a bitext from the monolingual text in the target language so that each source sentence is identical to the target sentence. This copied data is then mixed with the parallel corpus and the NMT system is trained like normal, with no metadata to distinguish the two input languages.

Our proposed method proves to be an effective way of incorporating monolingual data into low-resource NMT. On Turkish↔English and Romanian↔English translation tasks, we see gains of up to 1.2 BLEU over a strong baseline with back-translation. Further analysis shows that the linguistic phenomena behind these gains are different from and largely orthogonal to back-translation, with our copied corpus method improving accuracy on named entities and other words that should remain identical between the source and target languages.

1 Introduction

Neural machine translation (NMT) systems require a large amount of training data to make generalizations, both on the source side (in order to interpret the text well enough to translate it) and on the target side (in order to produce fluent translations). This data typically comes in the form of parallel corpora, in which each sentence in the source language is matched to a translation in the target language. Recent work (Gulcehre et al., 2015; Sennrich et al., 2016b) has investigated incorporating monolingual training data (particularly on the target side) into NMT. This effectively converts machine translation into a semi-supervised problem that takes advantage of both labeled (parallel) and unlabeled (monolingual) data. Adding monolingual data to NMT is important because sufficient parallel data is unavailable for all but a few language pairs and domains.

In this paper, we introduce a straightforward method for adding target-side monolingual training data to an NMT system without changing its architecture or training algorithm. This method converts a monolingual corpus in the target language into a parallel corpus by copying it, so that each source sentence is identical to its corresponding target sentence. This copied corpus is then mixed with the original parallel data and used to train the NMT system, with no distinction made between the parallel and the copied data.

We focus on language pairs with small amounts of parallel data where monolingual data has the most impact. On the relatively low-resource language pairs of English↔Turkish and English↔Romanian, we find that our copying technique is effective both alone and combined with back-translation. This is the case even when no additional monolingual data is used (i.e. when the copied corpus and the back-translated corpus are identical on the target side). This implies that back-translation does not make full use of monolingual data in low-resource settings, which makes sense because it relies on low-resource (and therefore low-quality) translation in the reverse direction.

2 Related Work

Early work on incorporating monolingual data into NMT concentrated on target-side monolingual data. Jean et al. (2015) and Gulcehre et al. (2015) used a 5-gram language model and a recurrent neural network language model (RNNLM), respectively, to re-rank NMT outputs. Gulcehre et al. (2015) also integrated a pre-trained RNNLM into NMT by concatenating hidden states. Sennrich et al. (2016b) added monolingual target data directly to NMT using null source sentences and freezing encoder parameters while training with the monolingual data. Our method is similar, although instead of using a null source sentence, we use a copy of the target sentence and train the encoder parameters on the copied sentence.

Sennrich et al. (2016b) also created synthetic parallel data by translating target-language monolingual text into the source language. To perform this process, dubbed *back-translation*, they first trained an initial target→source machine translation system on the available parallel data. They then used this model to translate the monolingual corpus from the target language to the source language. The resulting back-translated data was combined with the original parallel data and used to train the final source→target NMT system. Since this back-translation method outperforms previous methods that only train the decoder (Gulcehre et al., 2015; Sennrich et al., 2016b), we use it as our baseline. In addition, our method stacks with back-translation in both the target→source and source→target systems; we can use source text to improve the back-translations and target text to improve the final outputs.

In the mirror image of back-translation, Zhang and Zong (2016) added source-side monolingual data to NMT by first translating the source data into the target language using an initial machine translation system and then using this translated data and the original parallel data to train their NMT system. Our method is orthogonal: it could improve the initial system or be used alongside the translated data in the final system. They also considered a multitask shared encoder setup where the monolingual source data is used in a sentence reordering task.

More recent approaches have used both source and target monolingual data while simultaneously training source→target and target→source NMT systems. Cheng et al. (2016) accomplished this by concatenating source→target and target→source NMT systems to create an autoencoder. Monolingual data was then introduced by adding an autoencoder objective. This can be interpreted as back-translation with joint training. He et al. (2016) similarly used a small amount of parallel data to pre-train source→target and target→source NMT systems; they then added monolingual data to the systems by translating a sentence from the monolingual corpus into the other language and then translating it back into the original language, using reinforcement learning with rewards based on the language model score of the translated sentence and the similarity of the reconstructed sentence to the original. Our approach also employs an autoencoder, but rather than concatenate two NMT systems, we have flattened them into one standard NMT system.

Our approach is related to multitask systems. Luong et al. (2016) proposed conjoined translation and autoencoder networks; we use a single shared encoder. Further work used the same encoder and decoder for multi-way translation (Johnson et al., 2016). We have repurposed the idea to inject monolingual text for low-resource NMT. Their work combined multiple translation directions (e.g. French→English, German→English, and English→German) into one system. Our work combines e.g. English→English and Turkish→English into one system for the purpose of improving Turkish→English quality. They used only parallel data; our goal is to inject monolingual data.

3 Neural Machine Translation

We evaluate our approach using sequence-to-sequence neural machine translation (Cho et al., 2014; Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014) augmented with attention (Bahdanau et al., 2015). We briefly explain these models here.

Neural machine translation is an end-to-end approach to machine translation that learns to directly model $p(y \mid x)$ for a source-target sentence pair (x, y). The system consists of two recurrent neural networks (RNNs): the encoder and the decoder. In our experiments, the encoder is a bidirectional RNN with gated recurrent units (GRUs) that maps the source sentence into a vector representation. The decoder is an RNN language model conditioned on the source sentence. This is aug-

mented with an attention mechanism, which assigns weights to each of the words in the source sentence when modeling target words. This model is trained to minimize word-level cross-entropy loss; at test time, translations are generated using beam search.

4 Copied Monolingual Data for NMT

We propose a method for incorporating target-side monolingual data into low-resource NMT that does not rely heavily on the amount or quality of the parallel data. We first convert the target-side monolingual corpus into a bitext by making each source sentence identical to its target sentence; i.e., the source side of the bitext is a copy of the target side. We refer to this bitext as the *copied corpus*. The copied corpus is then mixed with the bilingual parallel corpus and no distinction is made between the two corpora. Finally, we train our NMT system with a single encoder and decoder using this mixed data. We are able to use the same encoder for both the parallel and the copied source sentences because we use byte pair encoding (Sennrich et al., 2016c) to represent the source and target words in the same vocabulary.

This copying method can also be combined with the back-translation method of Sennrich et al. (2016b). This is done by shuffling the parallel, back-translated, and copied corpora together into a single dataset and training the NMT system like normal, again making no distinction between the three corpora during training. We experiment with using the same monolingual data as the basis for both the back-translated and copied corpora (so that the target sides of the back-translated and copied corpora are identical) and with using two separate monolingual datasets for these purposes. Note that in the former case, each sentence in the original monolingual corpus occurs twice in the training data.

5 Experiments

5.1 Experimental Setup

5.1.1 Training Details

We train attentional sequence-to-sequence models (Bahdanau et al., 2015) implemented in Nematus (Sennrich et al., 2017). We use hidden layers of size 1024 and word embeddings of size 512. The models are trained using Adam (Kingma and Ba, 2015) with a minibatch size of 80 and a maximum sentence length of 50. We apply dropout (Gal and Ghahramani, 2016) in all of our EN↔TR and EN↔RO systems with a probability of 0.1 on word layers and 0.2 on all other layers. No dropout is used for EN↔DE. For all models, we use early stopping based on perplexity on the validation dataset. We decode using beam search on a single model with a beam size of 12, except for EN↔DE where we use a beam size of 5. For the experiments which use back-translated versions of the monolingual data, the target→source systems used to create the back-translations have the same setup as those used in the final source→target experiments.

Language pair	Parallel	Monolingual
EN↔TR	207 373	414 746
EN↔RO	608 320	608 320
EN↔DE	5 852 458	10 000 000

Table 1: Number of parallel and monolingual training sentences for each language pair.

5.1.2 Data and Preprocessing

We evaluate our models on three language pairs: English (EN) ↔ Turkish (TR), English ↔ Romanian (RO), and English ↔ German (DE). As shown in Table 1, these pairs each have vastly different amounts of parallel data. All of these languages have a substantial amount of monolingual data available.

The EN↔TR and EN↔DE data comes from the WMT17 news translation shared task,[1] while the EN↔RO data comes from the WMT16 shared task (Bojar et al., 2016). We use all of the available parallel data for each language pair, and the monolingual data comes from News Crawl 2015 (EN↔RO) or News Crawl 2016 (EN↔TR and EN↔DE). To create our monolingual datasets we randomly sample from the full monolingual sets.

For all language pairs, we tokenize and truecase the parallel and monolingual training data; we also apply byte pair encoding (BPE) to split words into subword units (Sennrich et al., 2016c). For each language pair, we learn a shared BPE model with 90,000 merge operations. Both the BPE model and the truecase model are learned on parallel data only (not on monolingual data). For RO→EN, we remove diacritics from the source training data, following the recommendation by Sennrich et al. (2016a).

[1] http://statmt.org/wmt17

	EN→TR		TR→EN		EN→RO	RO→EN	EN→DE		DE→EN	
BLEU	**2016**	**2017**	**2016**	**2017**	**2016**	**2016**	**2016**	**2017**	**2016**	**2017**
baseline	12.8	14.2	18.5	18.3	23.8	34.5	**33.3**	**26.6**	40.1	33.8
+ copied	**14.0**†	**15.2**†	**18.9**‡	**18.6**‡	**24.5**†	**35.7**†	**33.3**	26.3	**40.2**	**34.0**

Table 2: Translation performance in BLEU with and without copied monolingual data. Statistically significant differences are marked with † ($p < 0.01$) and ‡ ($p < 0.05$).

5.2 Translation Performance

We evaluate our models compared to a baseline containing parallel and back-translated data on the newstest2016 (all language pairs) and newstest2017 (EN↔TR and EN↔DE) test sets. For each model, we report case-sensitive detokenized BLEU (Papineni et al., 2002) calculated using `mteval-v13a.pl`.

The BLEU scores for each language pair and each system are shown in Table 2. The only difference between the baseline and the + *copied* systems is the addition of the copied corpus during training. Note that the copied and the back-translated corpora are created using identical monolingual data, which means that in the + *copied* system, each sentence from the monolingual corpus occurs twice in the training data (once as part of the copied corpus and once as part of the back-translated corpus).

For EN↔TR and EN↔DE, we use about twice as much monolingual as parallel data, so the ratio of parallel to back-translated to copied data is 1:2:2. For EN↔RO, we use a 1:1:1 ratio. In addition, for EN↔DE, we oversample the parallel corpus twice in order to balance the parallel and monolingual data.

For EN↔TR and EN↔RO, we observe statistically significant improvements (up to 1.2 BLEU) when adding the copied corpus. This indicates that our copied monolingual method can help improve NMT in cases where only a moderate amount of parallel data is available. For EN↔DE, we do not see improvements from adding the copied data; we conjecture that this occurs because this is a high-resource language pair. However, the EN↔DE systems trained with the copied corpus also do not perform any worse that those without.

5.3 Fluency

Adding copied target-side monolingual data results in a significant improvement in translation performance as measured by BLEU for EN↔TR and EN↔RO. Motivated by a desire to better understand the source of these improvements, we further experiment with the outputs for each system described in section 5.2. In particular, we want to examine whether these gains are simply due to the monolingual data improving the fluency of the NMT system.

In order to evaluate the fluency of each system, we train 5-gram language models for each language using KenLM (Heafield, 2011). The models are trained on the full monolingual News Crawl 2015 and 2016 datasets. This data is preprocessed as described in section 5.1, except that no subword segmentation is used.

We use these language models to measure perplexity on the outputs of the baseline systems (trained using parallel and back-translated data) and the + *copied* systems (trained using parallel, back-translated, and copied data). The language models are also queried on the reference translations for comparison. For all language pairs except EN↔RO, we concatenate newstest2016 and newstest2017 into a single dataset to find the perplexity.

Table 3 displays the perplexities for each system output and the reference. Interestingly, the perplexities for the baseline and the + *copied* systems are similar for all language pairs. In particular, improvements in BLEU (see Table 2) do not necessarily correlate to improvements in perplexity. This indicates that the gains from the + *copied* system may not solely be due to fluency.

5.4 Pass-through Accuracy

Since the copied monolingual data adds an autoencoder element to the NMT training, it is possible that the systems trained with copied data learn how to better pass through named entities and other relevant words than the baselines. In order to test this hypothesis, we detect words that are identical in each sentence in the source and the reference for the tokenized test data (excluding words that contain only one character and ignoring case). We then count how many of these words occur in the corresponding sentence in the translation output from each system. We calculate the pass-through

Perplexity	EN→TR	TR→EN	EN→RO	RO→EN	EN→DE	DE→EN
reference	700.0	146.7	202.4	118.1	231.0	116.5
baseline	**921.1**	**341.6**	**328.2**	248.4	**490.6**	317.3
+ copied	921.6	344.2	344.8	**245.5**	493.3	**314.2**

Table 3: Language model perplexities for the outputs of each NMT system.

Accuracy	EN→TR	TR→EN	EN→RO	RO→EN	EN→DE	DE→EN
baseline	77.3%	85.0%	71.5%	85.3%	78.5%	**91.4%**
+ copied	**82.0%**	**89.1%**	**78.5%**	**91.5%**	**78.6%**	91.1%

Table 4: Pass-through accuracy for the outputs of each NMT system.

accuracy as the percent of such words that appear in the output; these results are shown in Table 4.

For all language pairs except for EN↔DE, there is a large improvement in pass-through accuracy when the copied data is added during training. This closely mirrors the BLEU results discussed in section 5.2. These results suggest that a key advantage of using copied data is that the model learns to pass appropriate words through to the target output more successfully. Table 5 shows some examples of translations with improved pass-through accuracy for the + *copied* systems.

5.5 Additional EN-TR Experiments

In this section, we describe a number of additional experiments on EN→TR in order to investigate the effects of different experimental setups and aspects of the data. Note that the BLEU scores in this section are not directly comparable with those in Table 2, since a different subset of the monolingual data is used for some of these experiments. All BLEU scores reported in this section are on newstest2016 unless otherwise noted.

5.5.1 Double Back-Translated Data

In section 5.2, we report significant gains from our + *copied* systems over baselines trained on parallel and back-translated data for EN↔TR and EN↔RO, even while using the same monolingual data as the basis for both the copied and the back-translated corpora. However, in our experiments, we use particularly high-quality in-domain monolingual data. As a result, it is possible that these improvements are due to using this monolingual data twice (in the form of the back-translated and copied corpora) rather than to using the copied monolingual corpus.

In order to evaluate this, we consider an additional configuration in which we train using two copies of the same back-translated corpus (instead of using one copy of each of the back-translated corpus and the copied corpus). The results for this experiment are in Table 6. For both test sets, the + *copied* system performs better than the system with double back-translated data by about 1 BLEU point. This indicates that our copied corpus improves NMT performance, and that this is not simply due to the higher weight given to the high-quality monolingual data.

5.5.2 Different Copied Data

In our initial experiments, we use the same monolingual corpus to create the back-translated and the copied data. Here, we consider a variation in which we use different monolingual data for these purposes. This is done by cutting the monolingual corpus in half and back-translating only half of it, leaving the rest for copied data. Note that this means that the original monolingual corpus is the same size (twice the size of the parallel data; see Table 1), but each monolingual sentence only occurs once in the training data, rather than twice as before.

The results for these experiments are shown in Table 7. The baseline is trained on back-translations of all of the monolingual data, and the + *same copied* system contains the full copied corpus. The + *different copied* system uses different data for copying and back-translation. Both copied systems outperform the baseline, although the + *same copied* system does slightly better.

5.5.3 Copied Data Without Back-translation

Our results in section 5.2 show that our copied corpus method stacks with back-translation to improve translation performance when there is not much parallel data available. In this section, we study whether the copied corpus can aid NMT when no back-translated data is used. If so, this would be advantageous, as the copied corpus method is much simpler to apply than back-

	RO→EN
source	... a afirmat Angel Ubide, analist șef în cadrul **Peterson** Institute for International Economics.
reference	... said Angel Ubide, senior fellow at the **Peterson** Institute for International Economics.
baseline	... "said Angel Ubide, chief analyst at the **Carson** Institute for International Economics.
+ copied	... "said Angel Ubide, chief analyst at **Peterson** Institute for International Economics.
source	Les **Dissonances** a aparut pe scena muzicala în 2004 ...
reference	Les **Dissonances** appeared on the music scene in 2004 ...
baseline	Les **Dissonville** appeared on the music scene in 2004 ...
+ copied	Les **Dissonances** appeared on the music scene in 2004 ...
	TR→EN
source	**Metcash**, Bay **Douglass**'ın yorumlarına bir yanıt vermeyi reddetti.
reference	**Metcash** has declined to respond publicly to Mr **Douglass**' comments.
baseline	**Metah** declined to give an answer to Mr. **Doug**'s comments.
+ copied	**Metcash** declined to respond to a response to Mr. **Douglass**'s comments.
source	PSV teknik direktörü Phillip **Cocu**, şöyle dedi: "Çok kötü bir sakatlanma."
reference	Phillip **Cocu**, the PSV coach, said: "It's a very bad injury."
baseline	PSV coach Phillip **Coker** said: "It was a very bad injury.
+ copied	PSV coach Phillip **Cocu** said: "It's a very bad injury."

Table 5: Comparison of translations generated by baseline and + *copied* systems.

BLEU	2016	2017
parallel + back-translated	12.4	14.2
parallel + double back-translated	13.1	14.1
parallel + back-translated + copied	**14.0**	**15.2**

Table 6: EN→TR translation performance when using the back-translated corpus twice vs. the back-translated and copied corpora.

	BLEU
baseline	12.4
+ same copied	**13.6**
+ different copied	13.3

Table 7: EN→TR translation performance when using the same or different data for copied and back-translated corpora.

translation and does not require the training of an additional target→source machine translation system. We experiment with both a small copied corpus (about 200k sentences) and a large copied corpus (about 400k sentences).

The results for systems trained with only parallel and copied data are in Table 8. Both the small copied corpus and the large copied corpus yield large improvements (2.3-2.6 BLEU) over using parallel data only, and their performance is only slightly worse (0.3-0.4 BLEU) than the corresponding systems trained with only back-translated and parallel data.

5.5.4 Source Monolingual Data

Although we have concentrated thus far on incorporating target-side monolingual data into NMT, source-side monolingual data also has the potential to help translation performance. In particular, a source copied corpus can be used when training the target→source system for back-translation. Here, we test this strategy on EN→TR NMT with EN monolingual data. For this purpose, we randomly sample about 400k English sentences (twice the size of the parallel corpus) from the News Crawl 2015 monolingual corpus.

The results for this experiment are shown in Table 9. Although both copied systems improve over the baseline, adding the EN monolingual data does not result in further improvement over the target-only copied model, despite taking much longer to train.

	BLEU
parallel only	9.4
parallel + small copied	11.7
parallel + large copied	12.0
parallel + small back-translated	**12.0**
parallel + large back-translated	**12.4**

Table 8: EN→TR translation performance without back-translated data. We include systems trained with parallel and back-translated data (without copied data) for comparison.

	BLEU
baseline	12.4
+ copied	**13.6**
+ EN data	**13.6**

Table 9: EN→TR translation performance with EN monolingual data.

BLEU	1:1	2:1	3:1
baseline	12.0	12.4	12.8
+ copied	**13.0**	**13.6**	**13.8**

Table 10: EN→TR translation performance with different amounts of monolingual data.

5.5.5 Amount of Monolingual Data

Finally, we study the effectiveness of the copied monolingual corpus when the amount of monolingual data is varied. We consider three different monolingual corpus sizes: the same size as the parallel data (200k sentences; *1:1*), twice the size of the parallel data (400k sentences; *2:1*), and three times the size of the parallel data (600k sentences; *3:1*). We compare these different sizes for the baseline (parallel and back-translated data) and the + *copied* systems (parallel, back-translated, and copied data, where the back-translated and copied data are identical on the target side). Each smaller monolingual corpus is a subset of the larger monolingual corpora. Note that we do not oversample the parallel data to balance the different data sources.

Table 10 displays the results when different amounts of monolingual data are used. Note that we vary the amount of back-translated data in the baseline and of back-translated and copied data in the + *copied* system. For both the baseline and + *copied*, adding more monolingual data consistently yields small improvements (0.2-0.6 BLEU). In addition, the + *copied* system performs about 1.0 BLEU better than the baseline regardless of the amount of monolingual data. This is surprising since we do not oversample the parallel data at all. For the *2:1* and *3:1* cases, the systems see far less parallel than synthetic data, but the overall translation performances still improve.

6 Discussion

Our proposed method of using a copied target-side monolingual corpus to augment training data for NMT proved to be beneficial for EN↔TR and EN↔RO translation, resulting in improvements of up to 1.2 BLEU over a strong baseline. We showed that our method stacks with the previously proposed back-translation method of Sennrich et al. (2016b) for these language pairs. For EN↔DE, however, there was no significant difference between systems trained with the copied corpus and those trained without it. There was much more parallel training data for EN↔DE than for EN↔RO (nearly 10 times as much) and EN↔TR (about 28 times as much), so it is possible that the gains that would have come from the copied corpus were already achieved with the parallel data. Overall, the copied monolingual corpus either helped or was indifferent, so training with this corpus is not risky. In addition, it does not require any more monolingual data besides what is used for back-translation.

We initially assumed that the copied monolingual corpus was helping to improve the fluency of the target outputs. However, further study of the outputs did not necessarily support this assumption, as noted in section 5.3. Our method did improve accuracy when copying proper nouns and other words that are identical in the source and target languages; this is at least part of the explanation for the increases in BLEU score when using the copied corpus.

Subsequent experiments revealed various factors that influenced the effectiveness of the copied monolingual corpus. An unexpected finding was that doubling and tripling the size of the monolingual corpus (whether used as copied or back-translated data) resulted in small improvements (0.2-0.6 BLEU). We had originally thought that using much more monolingual than parallel data would result in a worse performance, since the system would see true parallel data less often than copied or back-translated data, but this did not turn out to be the case. Not having to limit the amount of monolingual data based on the availability of parallel data is an advantage for language pairs with much more monolingual than parallel data.

7 Conclusion

In this paper, we introduced a method for improving neural machine translation using monolingual data, particularly for low-resource scenarios. Augmenting the training data with monolingual data in which the source side is a copy of the target side proved to be an effective way of improving EN↔TR and EN↔RO translation, while not damaging EN↔DE (high-resource) translation. This technique could be used in combination with back-translation or with parallel data only. In addition, using much more monolingual than parallel data did not hinder performance, which is beneficial for the common case where a large amount of monolingual data is available but the language pair has little parallel data.

In the future, we plan on studying the effects of the quality of the monolingual data, since our copied corpus technique might in principle pose the risk of adding noise to the NMT system. In particular, we would like to apply a data selection method when creating the monolingual corpus, as the similarity of the monolingual and parallel data has been shown to have an effect on NMT (Cheng et al., 2016). We also hope to find an effective way of adding source monolingual training data. Finally, it would be interesting to do a manual evaluation of our method to confirm the BLEU and perplexity findings reported in sections 5.2 and 5.3.

Acknowledgments


This work was conducted within the scope of the Horizon 2020 Innovation Action *Health in My Language*, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644402. This work was partially funded by the Amazon Academic Research Awards program. We used Azure credits donated by Microsoft to The Alan Turing Institute. This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198. Association for Computational Linguistics.

Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Semi-supervised learning for neural machine translation. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 1965–1974. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in Neural Information Processing Systems 29*.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for WMT15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations*.

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

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin

Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the EACL 2017 Software Demonstrations*, pages 65–68. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of NAACL-HLT*, pages 86–96. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 1715–1725. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics.