NTT's Machine Translation Systems for WMT19 Robustness Task

Soichiro Murakami[1]*, **Makoto Morishita**[2]*, **Tsutomu Hirao**[2] and **Masaaki Nagata**[2]
[1] Service Innovation Department, NTT DOCOMO, INC., Japan
[2] NTT Communication Science Laboratories, NTT Corporation, Japan
souichirou.murakami.cr@nttdocomo.com
{makoto.morishita.gr, tsutomu.hirao.kp,
masaaki.nagata.et}@hco.ntt.co.jp

Abstract

This paper describes NTT's submission to the WMT19 robustness task. This task mainly focuses on translating noisy text (e.g., posts on Twitter), which presents different difficulties from typical translation tasks such as news. Our submission combined techniques including utilization of a synthetic corpus, domain adaptation, and a placeholder mechanism, which significantly improved over the previous baseline. Experimental results revealed the placeholder mechanism, which temporarily replaces the non-standard tokens including emojis and emoticons with special placeholder tokens during translation, improves translation accuracy even with noisy texts.

1 Introduction

This paper describes NTT's submission to the WMT 2019 robustness task (Li et al., 2019). This year, we participated in English-to-Japanese (En-Ja) and Japanese-to-English (Ja-En) translation tasks with a constrained setting, i.e., we used only the parallel and monolingual corpora provided by the organizers.

The task focuses on the robustness of Machine Translation (MT) to noisy text that can be found on social media (e.g., Reddit, Twitter). The task is more challenging than a typical machine translation task like the news translation tasks (Bojar et al., 2018) due to the characteristics of noisy text and the lack of a publicly available parallel corpus (Michel and Neubig, 2018). Table 1 shows example comments from Reddit, a discussion website. Text on social media usually contains various noise such as (1) abbreviations, (2) grammatical errors, (3) misspellings, (4) emojis, and (5) emoticons. In addition, most provided parallel corpora are not related to our target domain, and the amount of in-domain parallel corpus is still limited as compared with parallel corpora used in the typical MT tasks (Bojar et al., 2018).

(1) I'll let you know bro, thx
(2) She had a ton of rings.
(3) oh my god it's beatiful
(4) Thank you so much for all your advice!!😭💕
(5) (\ ∗ ´ ∀ ` ∗) so cute

Table 1: Example of comments from Reddit.

To tackle this *non-standard* text translation with a low-resource setting, we mainly use the following techniques. First, we incorporated a placeholder mechanism (Crego et al., 2016) to correctly copy special tokens such as emojis and emoticons that frequently appears in social media. Second, to cope with the problem of the low-resource corpus and to effectively use the monolingual corpus, we created a synthetic corpus from a target-side monolingual corpus with a target-to-source translation model. Lastly, we fine-tuned our translation model with the synthetic and in-domain parallel corpora for domain adaptation.

The paper is organized as follows. In Section 2, we present a detailed overview of our systems. Section 3 shows experimental settings and main results, and Section 4 provides an analysis of our systems. Finally, Section 5 draws a brief conclusion of our work for the WMT19 robustness task.

2 System Details

In this section, we describe the overview and features of our systems:

- Data preprocessing techniques for the provided parallel corpora (Section 2.2).
- Synthetic corpus, back-translated from the

*Equal contribution.

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 743–750
Florence, Italy, August 1-2, 2019.

	# sentences	# words
MTNT (for En-Ja)	5,775	280,543
MTNT (for Ja-En)	6,506	128,103
KFTT	440,288	9,737,715
JESC	3,237,376	21,373,763
TED	223,108	3,877,868

Table 2: The number of training sentences and words on the English side contained in the provided parallel corpora.

provided monolingual corpus, and noisy data filtering for its data. (Section 2.3).

- Placeholder mechanism to handle tokens that should be copied from a source-side sentence (Section 2.4).

2.1 NMT Model

Neural Machine Translation (NMT) has been making remarkable progress in the field of MT (Bahdanau et al., 2015; Luong et al., 2015). However, most existing MT systems still struggle with noisy text and easily make mistranslations (Belinkov and Bisk, 2018), though the Transformer has achieved the state-of-the-art performance in several MT tasks (Vaswani et al., 2017).

In our submission system, we use the Transformer model (Vaswani et al., 2017) without changing the neural network architecture as our base model to explore strategies to tackle the robustness problem. Specifically, we investigate how its noise-robustness against the noisy text can be boosted by introducing preprocessing techniques and a monolingual corpus in the experiments.

2.2 Data Preprocessing

For an in-domain corpus, the organizers provided the MTNT (Machine Translation of Noisy Text) parallel corpus (Michel and Neubig, 2018), which is a collection of Reddit discussions and their manual translations. They also provided relatively large out-of-domain parallel corpora, namely KFTT (Kyoto Free Translation Task) (Neubig, 2011), JESC (Japanese-English Subtitle Corpus) (Pryzant et al., 2017), and TED talks (Cettolo et al., 2012). Table 2 shows the number of sentences and words on the English side contained in the provided parallel corpora.

	# sentences	# words
MTNT (Japanese)	32,042	943,208
MTNT (English)	81,631	3,992,200

Table 3: The number of training sentences and words contained in the provided monolingual corpus.

Yamamoto and Takahashi (2016) pointed out that the KFTT corpus contains some inconsistent translations. For example, Japanese era names are only contained in the Japanese side and not translated into English. We fixed these errors by the script provided by Yamamoto and Takahashi (2016)[1].

We use different preprocessing steps for each translation direction. This is because we need to submit tokenized output for En-Ja translation, thus it seems to be better to tokenize the Japanese side in the same way as the submission in the preprocessing steps, whereas we use a relatively simple method for Ja-En direction.

For Ja-En, we tokenized the raw text into subwords by simply applying `sentencepiece` with the vocabulary size of 32,000 for each language side (Kudo, 2018; Kudo and Richardson, 2018). For En-Ja, we tokenized the text by KyTea (Neubig et al., 2011) and the Moses tokenizer (Koehn et al., 2007) for Japanese and English, respectively. We also truecased the English words by the script provided with Moses toolkits[2]. Then we further tokenized the words into subwords using joint Byte-Pair-Encoding (BPE) with 16,000 merge operations[3] (Sennrich et al., 2016b).

2.3 Monolingual Data

In addition to both the in-domain and out-of-domain parallel corpora, the organizers provided a MTNT monolingual corpus, which consists of comments from the Reddit discussions. Table 3 shows the number of sentences and words contained in the provided monolingual corpus.

As NMT can be trained with only parallel data, utilizing a monolingual corpus for NMT is a key

[1] https://github.com/kanjirz50/mt_ialp2016/blob/master/script/ja_prepro.pl

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

[3] Normally, Japanese and English do not share any words, thus using joint BPE does not seem effective. However, for this dataset, we found that Japanese sentences often include English words (e.g., named entities), so we use joint BPE even for this language pair.

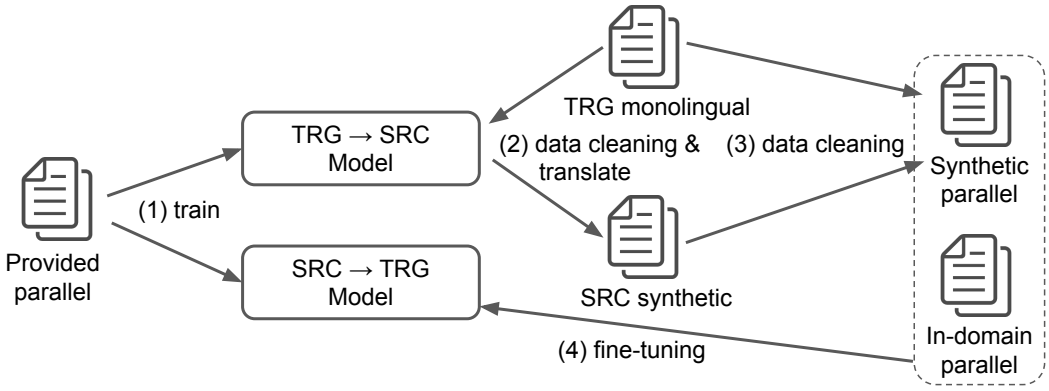

Figure 1: Overview of back-translation and fine-tuning.

challenge to improve translation quality for low-resource language pairs and domains. Sennrich et al. (2016a) showed that training with a synthetic corpus, which is generated by translating a monolingual corpus in the target language into the source language, effectively works as a method to use a monolingual corpus. Figure 1 illustrates an overview of the back-translation and fine-tuning processes we performed. (1) We first constructed both of source-to-target and target-to-source translation models with the provided parallel corpus. (2) Then, we created a synthetic parallel corpus through back-translation with the target-to-source translation model. (3) Next, we applied filtering techniques to the synthetic corpus to discard noisy synthetic sentences. (4) Finally, we fine-tuned the source-to-target model on both the synthetic corpus and in-domain parallel corpus.

Before the back-translation, we performed several data cleaning steps on the monolingual data to remove the sentences including ASCII arts and sentences that are too long or short. To investigate whether each sentence contains ASCII art or not, we use a word frequency-based method to detect ASCII arts. Since ASCII arts normally consist of limited types of symbols, the frequency of specific words in a sentence tends to be locally high if the sentence includes an ASCII art. Therefore, we calculate a standard deviation of word frequencies in each sentence of monolingual data to determine whether a sentence is like ASCII arts. More specifically, we first define a word frequency list $\mathbf{x_i}$ of the sentence i. For example, the word frequency list is denoted as $\mathbf{x}_i = [1, 1, 1, 1, 1]$ for the sentence i, "*That 's pretty cool .*" but as $\mathbf{x}_j = [1, 1, 1, 1, 3]$ for another sentence j, "*THIS IS MY LIFE ! ! !*". Note that the length of the list $\mathbf{x}_i$ is equal to the vocabulary size of the sentence i or j and each element of the list corresponds to the word frequency of a specific word. Second, we calculate the standard deviation σ_i of the word frequency list $\mathbf{x}_i$ for the sentence i. Finally, if σ_i is higher than a specific threshold, we assume that the sentence i contains an ASCII art and discard it from the monolingual data. We set the threshold to 6.0.

Moreover, since the provided monolingual data includes lines with more than one sentence, we first performed the sentence tokenization using the `spaCy`[4] toolkit. After that, we discarded the sentences that are either longer than 80 tokens or equal to 1 token.

Since a synthetic corpus might contain noisy sentence pairs, previous work shows that an additional filtering technique helps to improve accuracy (Morishita et al., 2018). We also apply a filtering technique to the synthetic corpus as illustrated in (3) in Figure 1. For this task, we use the `qe-clean`[5] toolkit, which filtered out the noisy sentences on the basis of a word alignment and language models by estimating how correctly translated and natural the sentences are (Denkowski et al., 2012). We train the word alignment and language models by using KFTT, TED, and MTNT corpora[6]. We use `fast_align` for word alignment and `KenLM` for language modeling (Dyer et al., 2013; Heafield, 2011).

2.4 Placeholder

Noisy text on social media often contains tokens that do not require translation such as emojis, "😄, 😎, ❤️", and emoticons, "m(_ _)m, (` · ω · ´),

[4] https://spacy.io
[5] https://github.com/cmu-mtlab/qe-clean
[6] Note that the JESC corpus is relatively noisy, thus we decided not to use it for cleaning.

\(^o^)/". However, to preserve the meaning of the input sentence that contains emojis or emoticons, such tokens need to be output to the target language side. Therefore, we simply copy the emojis and emoticons from a source language to a target language with a placeholder mechanism (Crego et al., 2016), which aims at alleviating the rare-word problem in NMT. Both the source- and target-side sentences containing either emojis or emoticons need to be processed for the placeholder mechanism. Specifically, we use a special token "<PH>" as a placeholder and replace the emojis and emoticons in the sentences with the special tokens.

To leverage the placeholder mechanism, we need to recognize which tokens are corresponding to emojis or emoticons in advance. Emojis can easily be detected on the basis of Unicode Emoji Charts[7]. We detect emoticons included in both the source- and the target-side sentences with the `nagisa`[8] toolkit, which is a Japanese morphological analyzer that can also be used as an emoticon detector for Japanese and English text.

Moreover, we also replace ">" tokens at the beginning of the sentence with the placeholders because ">" is commonly used as a quotation mark in social media posts and emails and does not require translation.

2.5 Fine-tuning

Since almost all the provided corpora are not related to our target domain, it is natural to adapt the model by fine-tuning with the in-domain corpora. Whereas we use both the MTNT and synthetic corpora for Ja-En, we only use the MTNT corpus for En-Ja because the preliminary experiment shows that synthetic corpus does not help to improve accuracy for the En-Ja direction. We suspect this is due to the synthetic corpus not having sufficient quality to improve the model.

3 Experiments

3.1 Experimental Settings

We used the Transformer model with six blocks. Our model hyper-parameters are based on *transformer_base* settings, where the word embedding dimensions, hidden state dimensions, feed-forward dimensions and number of heads are 512, 512, 2048, and 8, respectively. The model shares the parameter of the encoder/decoder word embedding layers and the decoder output layer by three-way-weight-tying (Press and Wolf, 2017). Each layer is connected with a dropout probability of 0.3 (Srivastava et al., 2014). For an optimizer, we used Adam (Kingma and Ba, 2015) with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.98$. We use a root-square decay learning rate schedule with a linear warmup of 4000 steps (Vaswani et al., 2017). We applied mixed precision training that makes use of GPUs more efficiently for faster training (Micikevicius et al., 2018). Each mini-batch contains about 8000 tokens (subwords), and we accumulated the gradients of 128 mini-batches for an update (Ott et al., 2018). We trained the model for 20,000 iterations, saved the model parameters each 200 iterations, and took an average of the last eight models[9]. Training took about 1.5 days to converge with four NVIDIA V100 GPUs. We compute case-sensitive BLEU scores (Papineni et al., 2002) for evaluating translation quality[10]. All our implementations are based on the `fairseq`[11] toolkit (Ott et al., 2019).

After training the model with the whole provided parallel corpora, we fine-tuned it with in-domain data. During fine-tuning, we used almost the same settings as the initial training setup except we changed the model save interval to every three iterations and continued the learning rate decay schedule. For fine-tuning, we trained the model for 50 iterations, which took less than 10 minutes with four GPUs.

When decoding, we used a beam search with the size of six and a length normalization technique with $\alpha = 2.0$ and $\beta = 0.0$ (Wu et al., 2016). For the submission, we used an ensemble of three (En-Ja) or four (Ja-En) independently trained models[12].

3.2 Experimental Results

Table 4 shows the case-sensitive BLEU scores of provided blind test sets. Replacing the emoticons

[7] https://unicode.org/emoji/charts

[8] https://github.com/taishi-i/nagisa

[9] The number of iterations might seem to be too low. However, Ott et al. (2018) showed that we could train the model with a small number of iterations if we use a large mini-batching. We also confirmed the model had already converged with this number of iterations.

[10] We report the scores calculated automatically on the organizer's website http://matrix.statmt.org/.

[11] https://github.com/pytorch/fairseq

[12] Originally, we planned to submit an ensemble of four for both directions. However, we could train only three models for En-Ja in time. In this paper, we also report the score of ensembles of four for reference.

	Ja-En		En-Ja	
Baseline model	10.8		14.3	
+ placeholders	12.2	(+1.4)	15.0	(+0.7)
+ fine-tuning	11.9	(+1.1)	16.2	(+1.9)
+ synthetic	14.0	(+3.2)	—	
+ 4-model ensemble	14.9	(+4.1)	17.0	(+2.7)
Submission	14.8		17.0	

Table 4: Case-sensitive BLEU scores of provided blind test sets. The numbers in the brackets show the improvements from the baseline model.

	Improved	Degraded	Unchanged
Ja-En	9 (53%)	0 (0%)	8 (47%)
En-Ja	14 (82%)	1 (6%)	2 (12%)

Table 5: The number of improved/degraded sentences by applying the placeholder mechanism compared with the baseline model. We manually evaluated all sentences containing placeholders in terms of whether the emojis and emoticons are correctly copied to the output.

and emojis with the placeholders achieves a small gain over the baseline model, which was trained with the provided raw corpora. Also, additional fine-tuning with in-domain and synthetic corpora also leads to a substantial gain for both directions. For Ja-En, although we failed to improve the accuracy by fine-tuning the MTNT corpus only, we found that the fine-tuning on both the in-domain and synthetic corpora achieves a substantial gain. We suspect this is due to overfitting, and modifying the number of iterations might alleviate this problem. As described in Section 2.5, we did not use the synthetic corpus for the En-Ja direction. For the submission, we decoded using an ensemble of independently trained models, which boosts the scores.

4 Analysis

4.1 Effect of Placeholders

To investigate the effectiveness of using the placeholder mechanism, we compared the translation of the baseline to the model trained with the placeholders. We manually evaluated how correctly the emojis and emoticons were copied to the output. Table 5 shows the numbers of sentences on the MTNT test set that are improved/degraded by applying the placeholder mechanism. These result demonstrate that the placeholder mechanism could improve the translation of the noisy text, which frequently includes emojis and emoticons, almost without degradation.

Tables 6 and 7 show examples of translations in the Ja-En and En-Ja tasks, respectively. Both the emoji (😂) and the ">" token, which represents a quotation mark, were properly copied from the source text to the translation of *+placeholders*, whereas the baseline model did not output such tokens as shown in Tables 6 and 7. Thus, we can consider this to be the reason the placeholders contribute to improving case-sensitive BLEU scores over the baseline.

In our preliminary experiments, although we tried a method to introduce the placeholder technique to our systems at the fine-tuning phase, we found that it does not work properly with only the fine-tuning. This means that an NMT needs to be trained with the corpus pre-processed for the placeholder mechanism before the fine-tuning.

4.2 Effect of Fine-tuning

According to the comparison between *+fine-tuning* and *baseline* shown in Table 4, fine-tuning on the in-domain and synthetic corpus achieved a substantial gain in both directions. Accordingly, we can see that the sentence translated by *+fine-tuning* has a more informal style than those translated by *baseline* and *+placeholders* as presented in Tables 6 and 7.

4.3 Difficulties in Translating Social Media Texts

Challenges still remain to improving the model's robustness against social media texts such as Reddit comments. As we pointed out in Section 1, various abbreviations are often used. For example, the term, "東スポWeb" (literally *East Spo Web*) in

Input	Woah woah, hang on a minute, let's hear this guy out. Amazing title 😂
Reference	おいおい、ちょっと待てよ。こいつの言うことを聞いてみようぜ。凄いタイトルだ😂
Baseline	うわぁ ちょっと 待 っ て こいつ の 話 を 聞 い て み ま しょ う 驚 く よう な 名前 だっ た わ ね (Well wait a minute let's listen to this story It was an amazing name)
+ placeholders	ちょっと 待 っ て ください この 人 の 話 を 聞 い て み ま しょ う 素晴らし い タイトル だ😂 (Wait a minute, let's hear the story of this person It's a great title 😂.)
+ fine-tuning	うわー 、 うわー 、 ちょっと 待 っ て 、 この 男 の 話 を 聞 こ う ぜ 。 すご い タイトル だ😂 (Wow, wow, wait a minute and hear this guy talk. It's an amazing title 😂.)

Table 6: Translation results on the English-to-Japanese development set. English sentences corresponding to the Japanese translations are also given.

Input	>男同士で物言えない奴のただの逆恨み
Reference	>Just misguided resentment from some fellow who can't speak amongst other men.
Baseline	A mere grudge against a man who can't say anything.
+ placeholders	> It's just a grudge against guys who can't say anything between men.
+ fine-tuning	>it's just inverted resentment for guys who can't say anything between men.

Table 7: Translation results on the Japanese-to-English test set.

the MTNT dataset should be translated to "*Tokyo Sports Website*" according to its reference, but our model incorrectly translated it to "*East Spoweb*". Such abbreviations that cannot be translated correctly without prior knowledge, such as "東スポWeb stands for 東京スポーツWebサイト (literally *Tokyo Sports Website*)", are commonly used on social media.

4.4 Use of Contextual Information

Some sentences need contextual information for them to be precisely translated. The MTNT corpus provides comment IDs as the contextual information to group sentences from the same original comment. We did not use the contextual information in our systems, but we consider that it would help to improve translation quality as in previous work (Tiedemann and Scherrer, 2017; Bawden et al., 2018). For example, in the following two sentences, "*Airborne school isn't a hard school.*" and "*Get in there with some confidence!*", which can be found in the MTNT corpus and have the same comment ID, we consider that leveraging their contextual information would help to clarify what "*there*" means in the latter and to translate it more accurately.

5 Conclusion

In this paper, we presented NTT's submission to the WMT 2019 robustness task. We participated in the Ja-En and En-Ja translation tasks with constrained settings. Through experiments, we showed that we can improve translation accuracy by introducing the placeholder mechanism, performing fine-tuning on both in-domain and synthetic corpora, and using ensemble models of Transformers. Moreover, our analysis indicated that the placeholder mechanism contributes to improving translation quality.

In future work, we will explore ways to use monolingual data more effectively, introduce contextual information, and deal with a variety of noisy tokens such as abbreviations, ASCII-arts, and grammar errors.

Acknowledgments

We thank two anonymous reviewers for their careful reading and insightful comments and suggestions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3rd International Conference on Learning Representations (ICLR).

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pages 1304–1313.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In Proceedings of the 6th International Conference on Learning Representations (ICLR).

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 3rd Conference on Machine Translation (WMT), pages 272–303.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: web inventory of transcribed and translated talks. In Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT), pages 261–268.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systran's pure neural machine translation systems. arXiv preprint arXiv:1610.05540.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The cmu-avenue french-english translation system. In Proceedings of the 7th Workshop on Statistical Machine Translation (WMT), pages 261–266.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pages 644–648.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In Proceedings of the 6th Workshop on Statistical Machine Translation (WMT), pages 187–197.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR).

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 (ACL), pages 177–180.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), pages 66–75.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 66–71.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir K. Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan M. Pino, and Hassan Sajjad. 2019. Findings of the first shared task on machine translation robustness. In Proceedings of the 4th Conference on Machine Translation (WMT).

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1412–1421.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 543–553.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In Proceedings of the 6th International Conference on Learning Representations (ICLR).

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2018. NTT's neural machine translation systems for WMT 2018. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 461–466.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL), pages 529–533.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pages 48–53.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In Proceedings of the 3rd Conference on Machine Translation (WMT), pages 1–9.

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 Association for Computational Linguistics (ACL), pages 311–318.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In Proceedings of the 15th European Chapter of the Association for Computational Linguistics (EACL), pages 157–163.

R. Pryzant, Y. Chung, D. Jurafsky, and D. Britz. 2017. JESC: Japanese-English Subtitle Corpus. arXiv preprint arXiv:1710.10639.

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 (ACL), pages 86–96.

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 (ACL), pages 1715–1725.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In Proceedings of the 3rd Workshop on Discourse in Machine Translation, pages 82–92.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), pages 6000–6010.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Kazuhide Yamamoto and Kanji Takahashi. 2016. Japanese orthographical normalization does not work for statistical machine translation. In Proceedings of the 20th International Conference on Asian Language Processing (IALP), pages 133–136.