TMU NMT System with Automatic Post-Editing by Multi-Source Levenshtein Transformer for the Restricted Translation Task of WAT 2022

Seiichiro Kondo and **Mamoru Komachi**
Tokyo Metropolitan University
kondo-seiichiro@ed.tmu.ac.jp, komachi@tmu.ac.jp

Abstract

In this paper, we describe our TMU English–Japanese systems submitted to the restricted translation task at WAT 2022 (Nakazawa et al., 2022). In this task, we translate an input sentence with the constraint that certain words or phrases (called restricted target vocabularies (RTVs)) should be contained in the output sentence. To satisfy this constraint, we address this task using a combination of two techniques. One is lexical-constraint-aware neural machine translation (LeCA) (Chen et al., 2020), which is a method of adding RTVs at the end of input sentences. The other is multi-source Levenshtein transformer (MSLevT) (Wan et al., 2020), which is a non-autoregressive method for automatic post-editing. Our system generates translations in two steps. First, we generate the translation using LeCA. Subsequently, we filter the sentences that do not satisfy the constraints and post-edit them with MSLevT. Our experimental results reveal that 100% of the RTVs can be included in the generated sentences while maintaining the translation quality of the LeCA model on both English to Japanese (En→Ja) and Japanese to English (Ja→En) tasks. Furthermore, the method used in previous studies requires an increase in the beam size to satisfy the constraints, which is computationally expensive. In contrast, the proposed method does not require a similar increase and can generate translations faster.

1 Introduction

We participated in the restricted translation task at WAT 2022. In this task, we were given pairs of an input sentence and a list of restricted target vocabularies (RTVs), wherein words or phrases are stored in a random order. Next, we were asked to generate a translated sentence for the input sentence that contained all the RTVs in the corresponding list. This setting is intended for cases where a user wishes to translate technical terms or proper nouns consistently by specifying these words in advance.

Previous studies have shown that neural machine translation (NMT) models (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) exhibit high translation performance in machine translation. Additionally, studies to control output in NMT under terminological constraints have been conducted (Hasler et al., 2018; Dinu et al., 2019; Chen et al., 2020; Song et al., 2020). However, several of these studies were set up to be available a bilingual dictionary rather than only the desired words to output.

In the previous year, the first shared task of restricted translation was performed, for which Chousa and Morishita (2021) achieved the highest score (Nakazawa et al., 2021). Their proposed method combines a "soft" method (which does not ensure constraint satisfaction using data augmentation (Chen et al., 2020)) and a "hard" method (which ensures constraint word satisfaction using grid beam search (Hokamp and Liu, 2017; Post and Vilar, 2018)). Their results revealed that certain constraint terms could be satisfied with only soft methods. We speculated whether the constraints could be satisfied by correcting those that were not satisfied by automatic post-editing.

In this study, we tackled this task in two generation steps. First, we generated the translation by a soft method (lexical-constraint-aware NMT (LeCA)). Next, we filtered the sentences that did not satisfy the constraints and post-edited those with multi-source Levenshtein transformer (MSLevT) (Wan et al., 2020). In general, hard methods employs a computationally expensive decoding algorithm compared with conventional beam search. We adopted MSLevT, an efficient non-autoregressive model, as the automatic post-editting from the perspective of computational complexity. In addition, while performing post-editing in MSLevT, RTVs were provided as initial values. Subsequently, the sentences were generated with repeated modifications according to the Levenshtein

Proceedings of the 9th Workshop on Asian Translation, pages 51–58
October 17, 2022.

transformer process. The restriction of delete and insert operations to RTVs ensured that RTVs would appear in the output in the order provided as the initial value. Consequently, we had to determine the order of the RTVs in advance. We used the cosine similarity of the embedding of each word in LeCA's generated text and RTVs, which were obtained using fasttext (Bojanowski et al., 2017), to determine the order of the RTVs.

We submitted the system outputs to the En→Ja task and Ja→En tasks. We successfully included 100% of the constraint words in the system's output without significantly compromising the BLEU score of the LeCA model. We confirmed the effectiveness of the proposed method in reordering constraint words by calculating Spearman's rank correlation coefficient for the reordered constraint words and the constraint words in the reference.

2 System Overview

First, we used a baseline model called lexical-constraint-aware NMT (Chen et al., 2020), for translation that considers constraint words. However, because this method did not ensure that constraint words would appear in the generated text, automatic post-processing correction was performed on the sentences that failed to satisfy the constraints in the LeCA output to ensure that the constraints were satisfied. The automatic correction was performed by reordering the RTVs using fasttext (Bojanowski et al., 2017) and then, using MSLevT (Wan et al., 2020).

2.1 Lexical-Constraint-Aware NMT

The LeCA model is designed to induce the model to include pre-specified words in the generated sentences by data augmentation. In particular, the RTVs are concatenated at the end of the input sentence, thus ensuring that LeCA obtains the source sentence and RTVs simultaneously before the decoding step and is expected to be able to start decoding, taking into account constraint words. Furthermore, LeCA employs a pointer network, which is expected to copy the constraint words concatenated in the input sentence at the appropriate places while generating the translation.

2.2 Sorting RTV with fasttext

Synonyms of the constraint words and those close to the surface form of the constraint words tended to appear in the output of LeCA when the constraint words were not included in the output. Therefore, we addressed the reordering of the constraint words under the assumption that the words corresponding to the constraint words are included in the output of LeCA.

En	Ja
0.664	0.718

Table 1: Evaluation of the proposed RTV-sorting method by Spearman's rank correlation coefficient between the order of sorted RTVs and that of references.

We adopted the following steps to align each RTV with a word in the LeCA outputs.

1. We obtained word embeddings of each word (both RTV and LeCA output) via fasttext.
2. If the RTV is a phrase consisting of multiple words, its embedding is the average of the embeddings of each word that constitutes the RTV. Assuming that the number of words in the output range of LeCA corresponding to an RTV is equal to the number of words in the RTV, the embedding of the output words of LeCA is summarized by taking the average over the n-gram of the number of words in the RTV. We call the n-gram "word block" and regard the first word in the word block as the representative word.
3. Cosine similarity ranking is considered for the RTV and all the word blocks.
4. Essentially, the RTV is considered to correspond to the word block with the highest ranking. However, if the corresponding word block (representative word) overlaps with other RTVs, the one with a higher cosine similarity is assigned priority. The RTV discarded here is considered to correspond to the next highest ranking word block.

Note that in a few cases, the number of output words of LeCA was smaller than the total number of words of RTVs. In such cases, the RTV was reordered randomly. [1]

Table 1 lists the Spearman's rank correlation coefficients. There were calculated from the RTV order when the proposed method used, and the RTV order that appeared in the reference in the entire

[1] In our experiments, we observed only one case in the Ja→En validation data set.

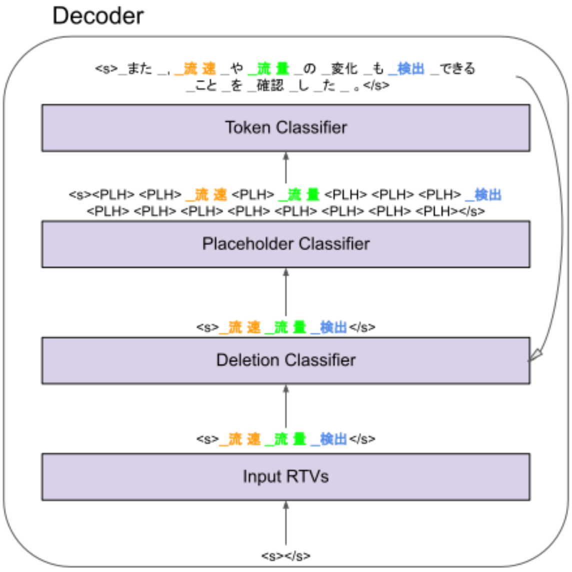
Decoder
<s>_また _, _流 速 _や _流 量 _の _変化 _も _検出 _できる _こと _を _確認 _し _た _ 。</s>
Token Classifier
<s><PLH> <PLH> _流 速 <PLH> _流 量 <PLH> <PLH> <PLH> _検出 <PLH> <PLH> <PLH> <PLH> <PLH> <PLH> <PLH> <PLH></s>
Placeholder Classifier
<s>_流 速 _流 量 _検出</s>
Deletion Classifier
<s>_流 速 _流 量 _検出</s>
Input RTVs
<s></s>

Figure 1: Decoder of Levenshtein transformers. The decoder repeats deletion, insertion, and replacement until the sentence is complete. This figure shows an example given three RTVs. The colored characters represent these. The generated Japanese sentence means "And, it was confirmed to enable also to detect change of flow rate and volume.", corresponding to "流速", "流量", and "体積" for "flow rate", "volume", and "detection", respectively.

test set. A positive correlation was observed, thus verifying the effectiveness of the proposed method.

2.3 Automatic Post-Editing by multi-source Levenshtein transformer

MSLevT has two encoders: one encoder is fed with the source sentence and the other with the output of the LeCA. Tebbifakhr et al. (2018) contends that in the APE task, a better representation of attention can be obtained by concatenating the outputs of two such encoders and subsequently passing them as an attention key.

Moreover, the decoder is provided with RTVs in parallel as initial values, and it operates similar to a Levenshtein transformer (Gu et al., 2019) (See Figure 1). The Levenshtein transformer generates sentences by repeating three phases, namely delete tokens, insert placeholders, and replace placeholders with new tokens, until the generated sentences stop varying or the number of iterations attains a pre-defined max-iteration. In the task setting in this study, both the deletion of RTVs given in the initial step and insertion of placeholders into the RTVs are undesirable. Therefore, we designed the model to prohibit these operations while generating the outputs.

2.4 Post-processing

We performed post-processing because the output of the model needed to be matched with that of the reference for submission. In particular, English words, certain symbols, and spaces in the Japanese text were normalized to full-width characters. In addition, in some cases, the model failed to recognize out-of-vocabulary characters in the constraint words that were not included in the training data and output special tokens. For these cases, we replaced the spans of constraint words that contained special tokens with constraint words.

3 Experimental Setup

3.1 Dataset

We used the provided ASPEC (Nakazawa et al., 2016) dataset. This dataset contains three million parallel sentences as training data; 1,790 parallel sentences as validation data; and 1,812 parallel sentences as test data. ASPEC training sentences are ordered by sentence alignment scores. Therefore, the sentences at the end are considered relatively noisy data. Morishita et al. (2017) reported that the translation quality of training with the original three million corpus is less than that of training with only the first two million sentences. Therefore, we used only the first two million sentences as training data.

Referring to Chousa and Morishita (2021) and Morishita et al. (2019), we tokenized both Japanese and English sentences using MeCab (Kudo et al., 2004) with the `mecab-ipadic-NEologd`[2] dictionary and `mosestokenizer` [3], respectively. Next we split these into subwords using sentencepiece (Kudo and Richardson, 2018), where the vocabulary size was set to 4,000.

3.2 Evaluation

Based on the official evaluation, we evaluated the outputs of our system using two metrics: the BLEU score (Papineni et al., 2002) and consistency score.

BLEU score. The BLEU score is calculated based on the n-gram matching rate between hypothesis and reference. We calculated it by SACREBLEU (Post, 2018).

[2] https://github.com/neologd/mecab-ipadic-neologd

[3] https://pypi.org/project/mosestokenizer/

	En→Ja			Ja→En		
	BLEU	RIBES	AMFM	BLEU	RIBES	AMFM
LeCA	51.3	0.873	0.800	39.3	0.796	0.653
LeCA + MSLevT	49.6	0.869	0.786	39.5	0.800	0.641
LeCA + MSLevT (dist→org)	49.9	0.870	0.786	39.6	0.800	0.638
LeCA + MSLevT (dist+org)	50.0	0.869	0.789	39.6	0.799	0.640
LeCA (× 5 ensemble) + MSLevT (dist+org)	52.2	0.877	0.789	41.3	0.808	0.654

Table 2: Results of the official score. Herein, "dist→org" implies that the model is pretrained with distilled data for ten steps and then finetuned by original data; and "dist+org" implies that the model is trained with mixed data consisting of distilled and original data.

	En→Ja		Ja→En	
	FS	AS	FS	AS
LeCA	37.6	4.24	23.0	4.22
LeCA + MSLevT (dist+org)	50.5	4.19	38.1	4.14
LeCA (× 5 ensemble) + MSLevT (dist+org)	52.7	4.18	40.8	4.31

Table 3: Results of human evaluation. Herein, FS denotes final score; and AS denotes adequacy scores on a 5-point scale.

Consistency score. The consistency score is the percentage of sentences in the test corpus that could be translated by including the given RTVs in the output. Whether or not an RTV is included in a sentence is determined by an exact match. While evaluating English sentences, we lowercased hypotheses and references, and performed character-based sequence matching (including white spaces).

Final score. For the final ranking, the score was calculated by combining the BLEU and consistency scores. In particular, the BLEU score was calculated with only the exact match sentences. Essentially, translations that did not satisfy the constraints were replaced to empty the string before measuring the BLEU score.

3.3 Model

LeCA. We used the Transformer big model. The implementation was based on that of Chen et al. (2020). The hyperparameters were based on the previous work of Chousa and Morishita (2021), with a learning rate of 0.001, max-token of 4,000, mini-batch size of 512,000 tokens, and the Adam optimizer.

fasttext. We used fasttext, which is available as a Python module. [4] Fasttext was learned from scratch using three million sentences of training data for Japanese and English.

multi-source LevT. We used an almost identical model and hyperparameters used in the previous study of Wan et al. (2020). However, their implementation could adversely affect the RTV when LevT performs delete and insert operations. Therefore, we modified the implementation to prohibit delete and insert operations on the RTV, referring to the implementation of Susanto et al. (2020).

In general, non-autoregressive models are known to improve the BLEU score by performing knowledge distillation (Gu et al., 2018; Zhou et al., 2020). Therefore, we prepared distilled data (which is LeCA's output as reference) for the training step. We used distilled data in two strategies, as follows. One is wherein the model is pretrained on the distilled data for ten steps and then finetuned by the original data. The other is wherein the model is trained with mixed data consisting of the distilled and original data.

4 Results

4.1 Official Evaluation

Official score Table 2 lists the official BLEU, RIBES (Isozaki et al., 2010), and AMFM (Banchs et al., 2015) scores, calculated in the evaluation server for our submissions. The results revealed

[4] https://fasttext.cc/docs/en/python-module.html

	En→Ja			Ja→En		
Model	BLEU	CS	FS	BLEU	CS	FS
LeCA	**52.0**	0.805	36.0	39.0	0.719	19.6
MSLevT	35.8	**1.000**	35.8	32.6	**1.000**	32.6
MSLevT (dist→org)	37.5	**1.000**	37.5	32.2	**1.000**	32.2
MSLevT (dist + org)	44.4	**1.000**	44.4	**39.4**	**1.000**	**39.4**
LeCA + MSLevT	50.1	**1.000**	50.1	39.3	**1.000**	39.3
LeCA + MSLevT (dist + org)	50.5	**1.000**	**50.5**	39.3	**1.000**	39.3

Table 4: Results of our evaluation. Herein, "dist→org" implies that the model is pretrained on the distilled data for ten steps and then, finetuned by the original data; "dist+org" implies that the model is trained with mixed data consisting of the distilled and original data; and CS and FS denote consistency score and final score, respectively.

		En→Ja		Ja→En	
	beam size	sec/sent	ratio	sec/sent	ratio
LeCA	5	0.094	×1.00	0.099	×1.00
	30	0.221	×2.35	0.228	×2.30
LeCA + MSLevT (proposed)	5	0.115	×1.22	0.126	×1.27

Table 5: Inference time on GPU.

that LeCA's scores were higher than those of LeCA+MSLevT. However, LeCA's output did not include 100% constraints. The use of distilled data for training MSLevT tended to be marginally more effective. The reason for the marginal improvement in scores may be that few sentences required automatic post-processing in MSLevT.

Human Evaluation Table 3 lists the human evaluation and official final scores (Nakazawa et al., 2022). Human evaluation performed adequacy scores on a 5-point scale by the WAT organization. Our proposed method has higher Final Scores [5] because it reliably includes RTVs in the output, but the adequacy of the human evaluation tends to be marginally lower.

4.2 Our Evaluation

Table 4 lists the scores obtained in our evaluation.

English to Japanese Although LeCA achieved the highest BLEU score, the consistency score was 0.805, and the final score was significantly lower by 16.0. In contrast, "MSLevT" (which is the result of passing the LeCA's output through MSLevT) exhibited a significant decrease in BLEU, although all the RTVs could be output. However, our proposed combined approach ("LeCA + MSLevT") maintained BLEU scores comparable to those of LeCA and the consistency score was 1.000.

With regard to the effectiveness of the distillation data for MSLevT, training the model with mixed data consisting of the distilled and original data is the most effective approach for improving the BLEU score. However, MSLevT's improvement by distilled data had a negligible impact on "LeCA + MSLevT" (by 0.4 points). The likely cause of this is that the revision of only the 20% texts by MSLevT is not influenced by the presence or absence of distilled data. An analysis of this aspect is for future work.

Japanese to English Although LeCA achieved a BLEU score of 39.0, the consistency score was 0.719, and the final score was significantly lower by 19.4. In contrast, "MSLevT" exhibited a decrease in BLEU, although all the RTVs could be output. However, our proposed combined approach ("LeCA + MSLevT") maintained BLEU scores comparable to those of LeCA and the consistency score was 1.000 (similar to En→Ja).

Moreover, the effectiveness of the distillation data for MSLevT exhibited a trend similar to that of En→Ja. However, the BLEU score of "MSLevT" was higher than those of "LeCA" and "LeCA +

[5] The evaluation by the organizer in the ja-en test set showed that consistency score did not reach 100%. We found that this was due to the inclusion of escape sequences in 39 sentences at submission.

MSLevT (dist + org)." This implies that for English texts, applying all the LeCA outputs to MSLevT is more effective compared with being selective.

4.3 Inference Time

In the previous study by Chousa and Morishita (2021), the authors used grid beam search to generate translations. However, they reported that the method generated repetitions when the beam size was small and could not generate all the constraint words. Therefore, they performed a preliminary experiment and determined the beam size as 30 to generate a translation that included all the constraint words. However, larger beam sizes require more inference time. In contrast, our method can satisfy the RTVs without increasing the beam size.

Table 5 lists the time required for inference by LeCA with beam sizes of 5 and 30, and that by the proposed method with 5. The experiments verified that the time required to generate the translations by the proposed method was significantly shorter than that by LeCA with a beam size of 30.

5 Related Work

NMT with terminology constraints have been studied widely. In particular, the Machine Translation using Terminologies task in WMT2021 (Akhbardeh et al., 2021) had a setting that was highly similar to that in this study. Unlike this study, WMT's task provided terminology dictionaries. Consequently, such setting-specific approaches were observed. For example, Wang et al. (2021) employed a method of replacing words in the input sentence that corresponded to constrained source words with the constrained target words. Furthermore, Ailem et al. (2021) used a selective constraint word selection method during training based on dictionaries.

Bergmanis and Pinnis (2021) worked on a similar task in a setting that was marginally looser than that in this study. They differed from the other studies in that they focused on word conjugation as well, although their approach was to replace the constraining words in the input sentence with words from the target side. They added a process of lemmatizing the words to be replaced on the target side to ensure that the model could flexibly learn conjugations.

In the previous year, Chousa and Morishita (2021) achieved the highest score in the restricted translation task in WAT2021 (Nakazawa et al., 2021). Their proposed method combines LeCA and grid beam search (Hokamp and Liu, 2017; Post and Vilar, 2018). Although grid beam search can consistently output constraint words, it incurs high computational cost and is known to adversely affect translation accuracy if a sufficient beam width is not adopted. Chousa and Morishita (2021) demonstrated that this problem can be mitigated by combining grid beam search with LeCA.

6 Conclusion

We introduced an automatic post-editing approach for the restricted translation task of WAT 2022. In our experiments, 100% of the RTVs could be included in the generated sentences while maintaining the translation quality of LeCA. Furthermore, our method does not require any preliminary experiments to determine the beam size and can generate translations faster while satisfying constraints compared with existing methods using grid beam search.

References

Melissa Ailem, Jingshu Liu, and Raheel Qader. 2021. Lingua custodia's participation at the WMT 2021 machine translation using terminologies shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 799–803, Online. Association for Computational Linguistics.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

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*, San Diego, California.

Rafael E. Banchs, Luis F. D' Haro, and Haizhou Li. 2015. Adequacy–fluency metrics: Evaluating MT in the continuous space model framework. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):472–482.

Toms Bergmanis and Mārcis Pinnis. 2021. Facilitating terminology translation with target lemma annotations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3105–3111, Online. Association for Computational Linguistics.

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.

Guanhua Chen, Yun Chen, Yong Wang, and Victor O.K. Li. 2020. Lexical-constraint-aware neural machine translation via data augmentation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3587–3593. International Joint Conferences on Artificial Intelligence Organization.

Katsuki Chousa and Makoto Morishita. 2021. Input augmentation improves constrained beam search for neural machine translation: NTT at WAT 2021. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 53–61, Online. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada. OpenReview.net.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, volume 32, pages 11181–11191. Curran Associates, Inc.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. 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.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237, Barcelona, Spain. Association for Computational Linguistics.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2017. NTT neural machine translation systems at WAT 2017. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 89–94, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2019. NTT neural machine translation systems at WAT 2019. In *Proceedings of the 6th Workshop on Asian Translation*, pages 99–105, Hong Kong, China. Association for Computational Linguistics.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Raj Dabre, Shohei Higashiyama, Shantipriya Parida, Anoop Kunchukuttan, Makoto Morishita, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, and Sadao Kurohashi. 2022. Overview of the 9th workshop on Asian translation. In *Proceedings of the 9th Workshop on Asian Translation (WAT2022)*, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, and Sadao Kurohashi. 2021. Overview of the 8th workshop on Asian translation. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 1–45, Online. Association for Computational Linguistics.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2204–2208, Portorož, Slovenia. European Language Resources Association (ELRA).

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Kai Song, Kun Wang, Heng Yu, Yue Zhang, Zhongqiang Huang, Weihua Luo, Xiangyu Duan, and Min Zhang. 2020. Alignment-enhanced transformer for constraining NMT with pre-specified translations. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 8886–8893, New York City, New York. Association for the Advancement of Artificial Intelligence.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with Levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online. 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*, pages 3104–3112, Montreal, Canada. Curran Associates, Inc.

Amirhossein Tebbifakhr, Ruchit Agrawal, Matteo Negri, and Marco Turchi. 2018. Multi-source transformer with combined losses for automatic post editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 846–852, Belgium, Brussels. Association for Computational Linguistics.

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*, pages 5998–6008, Long Beach, California. Curran Associates, Inc.

David Wan, Chris Kedzie, Faisal Ladhak, Marine Carpuat, and Kathleen McKeown. 2020. Incorporating terminology constraints in automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1193–1204, Online. Association for Computational Linguistics.

Ke Wang, Shuqin Gu, Boxing Chen, Yu Zhao, Weihua Luo, and Yuqi Zhang. 2021. TermMind: Alibaba's WMT21 machine translation using terminologies task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 851–856, Online. Association for Computational Linguistics.

Chunting Zhou, Jiatao Gu, and Graham Neubig. 2020. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*.