Input Augmentation Improves Constrained Beam Search for Neural Machine Translation: NTT at WAT 2021

Katsuki Chousa* and **Makoto Morishita***
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan
{katsuki.chousa.bg, makoto.morishita.gr}@hco.ntt.co.jp

Abstract

This paper describes our systems that were submitted to the restricted translation task at WAT 2021. In this task, the systems are required to output translated sentences that contain all given word constraints. Our system combined input augmentation and constrained beam search algorithms. Through experiments, we found that this combination significantly improves translation accuracy and can save inference time while containing all the constraints in the output. For both En→Ja and Ja→En, our systems obtained the best translation performances in both automatic and human evaluations.

1 Introduction

This year, we participated in the restricted translation task at WAT 2021 (Nakazawa et al., 2021), in which we were asked to control a model so that the translation output would contain specified terms. Although the recent neural machine translation (NMT) model achieves excellent performance, controlling its output is still a challenging task. Figure 1 shows an overview of the task. Each sentence includes the target words (constraints) that must be contained in the output. We believe this task reflects a critical function, especially in practical applications. For example, users may want to control the translation of technical terms or proper nouns.

Several works have tried to control the NMT outputs, and these works can be divided into two categories: *hard* and *soft* methods. The hard lexically constrained method guarantees that all the target words are in the output. Current works achieve this by modifying the beam search algorithm to find the hypothesis that contains all of the target words (Hokamp and Liu, 2017; Post and Vilar, 2018). The hard method guarantees all constraints are satisfied, but its translation performance is sometimes lower than the conventional NMT. This is because it requires all given target words to be contained in the decoding step, which may disrupt the model inference.

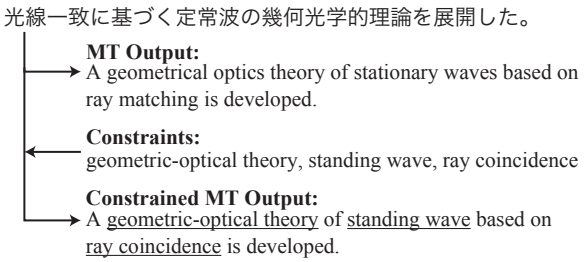

Figure 1: Overview of the restricted translation task

The soft lexically constrained method, on the other hand, does not guarantee that all target words are contained in the output. These methods usually modify or augment the input of the NMT model and try to output the given target words without changing the decoding algorithm (Song et al., 2019; Chen et al., 2020). Its decoding speed is usually faster than the hard method, but some of the constraints may not be satisfied.

Our submission aims to contain all of the specified target words with high translation accuracy. To achieve this goal, we applied both input augmentation and constrained beam search algorithms. To the best of our knowledge, this is the first work that combines these two methods. Through experiments, we found that this combination achieves quite high translation performance while containing all target words in the output and saving inference time. We submitted the systems to the English-to-Japanese (En→Ja) and Japanese-to-English (Ja→En) tasks, and we were ranked first in both language pairs in terms of BLEU scores and human evaluations.

*Equal contribution.

Proceedings of the 8th Workshop on Asian Translation, pages 53–61
Bangkok, Thailand (online), August 5-6, 2021.

2 Task Definition

Suppose we have a source sentence $X = (x_1, x_2, \ldots, x_S)$ with S tokens and a target sentence $Y = (y_1, y_2, \ldots, y_T)$ with T tokens. In a conventional machine translation approach, the problem of translation from X to Y can be solved by finding the best target sentence that maximizes the conditional probability

$$p(Y \mid X) = \prod_{t=1}^{T} p(y_t \mid y_{<t}, X). \quad (1)$$

In the restricted translation task, lists of target words are provided to represent word restrictions, and systems are required to output translations that contain all of the target words in each list. Here, the problem of translation with word constraints can be defined as

$$p(Y \mid X, C) = \prod_{t=1}^{T} p(y_t \mid y_{<t}, X, C), \quad (2)$$

where $C = (C_1, C_2, \ldots, C_N)$ is the provided word constraints with N phrases, and the constraints are given in random order.

The performance of systems in this task is evaluated through two metrics:

- Translation accuracy: BLEU (Papineni et al., 2002) is used for evaluation in this task.
- Consistency score: The percentage of sentences that correctly contain the given constraints over the entire test set.

For the final ranking, the combined score of the above metrics is calculated as follows:

1. If the translation does not contain all of the constraints based on exact matching, replace the translation with an empty string.
2. Calculate BLEU scores with modified translations.

3 Data

3.1 Provided Data

In this task, we were asked to translate an English/Japanese scientific paper. As the in-domain training data, organizers provided ASPEC (Nakazawa et al., 2016), which contains three million parallel sentences. Since this corpus is ordered by the sentence-alignment quality, the sentences at the end might be noisy. Following a previous work (Morishita et al., 2017), we used only the first two million sentences as parallel sentences. We treated the final one million sentences as monolingual data and created a synthetic corpus (Sennrich et al., 2016). Based on a previous analysis (Morishita et al., 2019), we forward-translated it for the Japanese-English task and back-translated it for the English-Japanese task.

Architecture	Transformer (big)
Tied-embeddings	Tied the encoder/decoder embeddings and the decoder output layer
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$) (Kingma and Ba, 2015)
Learning Rate Schedule	Inverse square root decay
Warmup Steps	4,000
Max Learning Rate	0.001
Dropout	0.3
Gradient Clipping	1.0
Label Smoothing	$\epsilon_{ls} = 0.1$ (Szegedy et al., 2016)
Mini-batch Size	512,000 tokens (Ott et al., 2018)
Number of Updates	8,000 steps
Averaging	Save checkpoint for every 100 steps and take an average of last 8 checkpoints

Table 1: List of hyperparameters

3.2 Other Resources

We also trained the model with additional resources. As an additional parallel corpus, we used JParaCrawl (Morishita et al., 2020), which contains 10 million sentence pairs.

We also used CommonCrawl provided by the WMT 2020 news shared task (Barrault et al., 2020) as additional monolingual data. For CommonCrawl data, we chose the ten million English and Japanese sentences that are similar to the scientific domain based on the language model trained with ASPEC (Moore and Lewis, 2010). Then we further filtered out the following noisy sentences: (1) non-English/Japanese sentences with CLD2 [1], (2) excessively long sentences (more than 250 subwords), (3) sentences that contain out-of-vocabulary characters. After cleaning, we kept 7.9 million English and 9.2 million Japanese sentences. We then back-translated these sentences with the NMT model trained with ASPEC to make a synthetic corpus.

Setting	BLEU	Term%	Sent%
BASE	29.4	50.80	23.3
+ LCD (beam=60)	24.0	94.40	85.3
LeCA	42.2	87.64	72.02
+ LCD (beam=30)	43.9	94.34	85.21

Table 2: Comparison of translation accuracy and consistency score for each setting on Ja→En.

4 System Details

4.1 Base Model and Hyperparameters

As a baseline system, we employed the Transformer model with the big setting (Vaswani et al., 2017). Table 1 shows the detailed settings and hyperparameters. As an NMT implementation, we used `fairseq` (Ott et al., 2019), and modified it in the following experiments.

4.2 Lexically Constrained Decoding

We used the lexically constrained decoding (LCD) technique (Hokamp and Liu, 2017; Post and Vilar, 2018) to incorporate constraints at decoding time. In this task, the translations that do not satisfy the constraints lead to a substantial decrease in the final score. This technique is a hard lexically constrained method that uses grid beam search algorithm, and it guarantees that all word constraints appear in the target sentence.

To evaluate the effectiveness of this technique, we compared the baseline model (BASE) and the baseline with LCD (BASE+LCD). Here, we used two metrics for the consistency score: term% is the percentage of constraints that are correctly generated in the translations, and sent% is the percentage of sentences that contain all given constraints. Table 2 shows that the BASE+LCD significantly improves both term% and sent% on Ja→En. The reason why the two consistency scores of BASE+LCD are not 100% is due to the normalization on the tokenization, and this can be addressed by post-processing (§4.7).

However, BASE+LCD decreased the translation accuracy of the model. In preliminary experiments with the baseline models, we also found that the beam size needs to be larger than 60 to successfully generate all the constraints in this task. This is because the translations contain much repetition and the model never finishes generation before reaching the maximum output length.

[1] https://github.com/CLD2Owners/cld2

4.3 LExical-Constraint-Aware NMT

To ease the problem in LCD, we used the Lexical-Constraint-Aware NMT (LeCA) model (Chen et al., 2020), whose input is augmented by concatenating constraints and the source sentence together. This method can inform the model of what constraints are given before decoding time, and thus the model can properly decide where to output a constraint. LeCA is a one of the soft lexically constrained methods, which do not guarantee all constraints are in the output. However, in combination with LCD, we can guarantee the model always satisfies the constraints while keeping or improving the translation performance.

The input is constructed by concatenating the source sentence X and each phrase C_i in the constraints C with a separator symbol $\langle \texttt{sep} \rangle$, as follows:

$$[X, \langle \texttt{sep} \rangle, C_1, \langle \texttt{sep} \rangle, C_2, \ldots, C_N, \langle \texttt{eos} \rangle], \quad (3)$$

where $\langle \texttt{eos} \rangle$ is the symbol indicating the end of the sentence.

To construct the input at training time, Chen et al. (2020) proposed a method that dynamically samples constraints from a reference sentence. They first sampled the number of constrained words k, and then they randomly sampled k target words (not subwords) as constraints from the reference. Here, we sampled the number of constrained words k from 0 to 14 following the distribution that is $p = 0.4$ for 0 and $p = 0.6/14 (= 0.04)$ for the other ones. The high probability for no constraint is to maintain the translation performance for unconstrained settings.

To handle such a source sequence, this method modifies the input representation of the encoder to distinguish the source sentence and each constraint. This representation is composed of three types of learned embeddings: token embeddings, positional embeddings, and segment embeddings, as shown in Fig. 2. The position of each constraint starts from the maximum length of the source sentences to avoid overlapping with the sentence. We assigned different values for the source sentence and each constraint and fed it to the model with the segment embeddings. This method also introduces a pointer network architecture(Vinyals et al., 2015; See et al., 2017) that helps to generate constraints by copying from the source sequence. Finally, we updated the models with 10,000 steps for Ja→En and 12,000 steps for En→Ja and set the beam size to 30 for

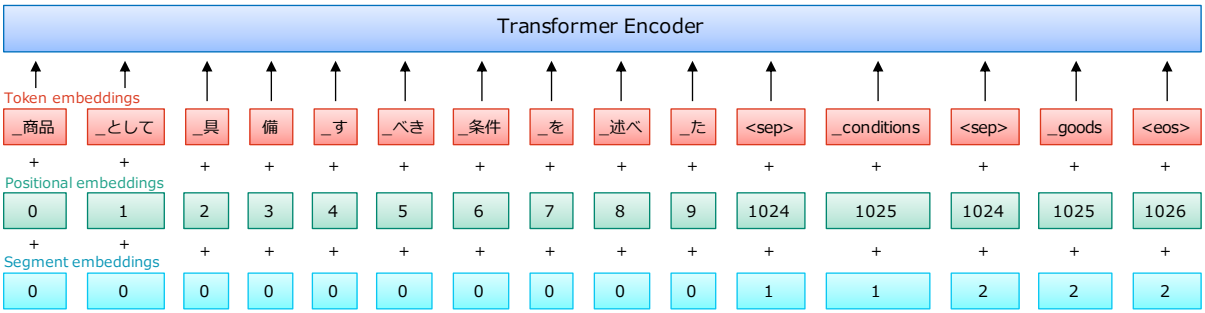

Figure 2: Input representation of the LeCA model.

Tokenizer	BLEU	Term%	Sent%
MeCab + ipadic	44.8	68.67	43.87
MeCab + NEologd	46.5	72.35	49.39

Table 3: Comparison of translation performance when changing the dictionary of tokenizer on En→Ja. The model setting is LeCA with a few updates.

LCD.

We evaluated the effectiveness of LeCA and LeCA with LCD (LeCA+LCD). Table 2 shows that LeCA achieved high translation accuracy and consistency scores. The input of both LeCA and BASE+LCD are the same, but the translation accuracy of LeCA is significantly better than that of BASE+LCD. Moreover, LeCA+LCD with a small beam size improves the translation accuracy and satisfies all of the constraints. This implies that inputting both a source sentence and constraints as source sequence is very effective for improving the performance in this task.

4.4 Pre-process

Since constraints that are sampled from the reference are given as not a subword but a word, we need to separate the sentence into words. To do this, we first tokenized both the input and output sentences. For English, we simply applied the tokenizer scripts available in the `Moses` toolkit (Koehn et al., 2007). We used the Moses `truecaser` when the target language is English. For Japanese, we use the `MeCab` tokenizer (Kudo, 2006) with the `mecab-ipadic-NEologd` (Sato, 2015) dictionary. This dictionary contains many neologisms and thus it helps in handling named entities or technical terms, which are included in ASPEC but cannot be tokenized correctly using the default system dictionary. We compared the LeCA performance of `mecab-ipadic-NEologd` with the default system dictionary on an En→Ja task. Table 3 shows that `mecab-ipadic-NEologd` significantly improved translation accuracy and consistency scores. We confirmed that using `mecab-ipadic-NEologd` is the best option for LeCA on this task.

Then, we trained subword encoding models using the `sentencepiece` implementation (Kudo and Richardson, 2018). According to an earlier work (Morishita et al., 2019), a smaller vocabulary size (e.g., 4,000) is empirically superior to the commonly used ones (e.g., 32,000). On the other hand, larger vocabulary size is preferred for an LCD to keep the number of constraint tokens small. This is because a large number of tokens requires a large beam size of the LCD and increases the inference time. Finally, we found in a preliminary experiment that a vocabulary size of 32,000 achieved the best results, so we used a joint subword vocabulary with 32,000 tokens. For training data, we applied the Moses `clean-corpus-n` scripts to remove sentence pairs that are either too long or too different int their lengths[2].

4.5 Fine-Tuning and Data Selection

The synthetic corpora (e.g., ASPEC last 1M and CommonCrawl) contain noisy sentence pairs, and the domain of JParaCrawl is different from that of ASPEC, a scientific paper domain. We used these corpora to make the translations more fluent. The model was initially pre-trained with these corpora and the first 2M sentence pairs of ASPEC for 12,000 updates. We then fine-tuned the pre-trained model using only the first 2M sentence pairs of ASPEC for 2,000 steps. For the pre-training, we oversampled ASPEC three-times to keep roughly the same number of sentences as the synthetic cor-

[2]We set the minimum length to 1, the maximum length to 250, and the maximum ratio of lengths to 9.

Setting	BLEU Ja→En	BLEU En→Ja
ASPEC 2M	44.34	—[3]
+ synth 1M	44.26	56.57
after pre-training	44.28	56.47

Table 4: Effectiveness of fine-tuning. The model settings are LeCA+LCD.

Model type	BLEU En→Ja	BLEU Ja→En
Single model	55.49	43.44
8 Ensemble	56.57	44.34

Table 5: Effectiveness of ensembling models. The model settings are LeCA+LCD.

pora.

We searched for an effective setting to use the training data. Table 4 shows the results. The model using only ASPEC 2M for En→Ja and the model using ASPEC 2M and forward-translated ASPEC last 1M for Ja→En achieved the highest translation accuracies. For both En→Ja and Ja→En, the models trained on ASPEC 2M after pre-training achieved comparable results to the best ones. Since these models are trained on large amounts of parallel sentence pairs, they might be expected to produce more natural output than the best ones and thus be preferred by humans. Therefore, we decided to submit these four models for human evaluation.

4.6 Ensemble

We applied a model ensemble technique to improve the translation accuracy. First, eight models were trained with different random seeds. We then computed the average scores of these models and generated hypotheses based on these scores using beam search decoding.

Table 5 shows the effectiveness of ensembling models. Ensembling the eight models shows a significant improvement over the single model.

4.7 Post-processing

For the submission, we need to match the tokenization to the reference constraints. To achieve this, we fixed the terms that are not matched to the constraints due to tokenization issues. Specifically, for each unmatched constraint, we removed spaces in both the output and the constraint, and then replaced the constraint in the output with the reference-spaced constraint. In some cases, we found that constraints may contain out-of-vocabulary (OOV) characters, resulting in translation failure[4]. The model outputs the special OOV tokens for these sentence, and thus we replaced them with correct characters in the reference constraint.

5 Official Results

Table 6 shows the automatic evaluated performance of our systems on the test set. These scores were measured in the evaluation server[5]. The best systems improved the BLEU score by +11.93 pts for En→Ja and +15.04 pts for Ja→En against the BASE. Our systems achieved the best BLEU score for both En→Ja and Ja→En subtasks.

Table 7 shows the official results of our systems[6]. For both En→Ja and Ja→En, our systems achieved the best scores in the final ranking. Our submissions did not drop the scores from the BLEU, while the other participants dropped it. This means that our team only succeeded in implementing systems whose translation output could contain all the specified terms. Our systems also achieved the best performance in terms of human evaluations for both En→Ja and Ja→En. Notably, our scores are better than the reference ones even for Ja→En. This implies that constrained translation can yield human-parity performance when the system can receive appropriate terms in the target language.

6 Analysis

Figure 3 shows the example translation of the baseline and LeCA with lexically constrained decoding. Underlines in Figure 3 show the terms that match the constraints. Obviously, the baseline model generated the same term repeatedly and failed to translate while all of the constraints were satisfied. The baseline model appears to struggle with generating

[3] In a preliminary experiment on En→Ja, we found that a model using synthetic data was superior to that using only ASPEC 2M. However, we did not compare the three settings under the same conditions.

[4] We found that two percent of the lines in the test set include OOV characters.

[5] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/index.html

[6] The results of all participants are reported in https://sites.google.com/view/restricted-translation-task/#h.g3vfoh2oljpq

ID	Setting	BLEU En→Ja	BLEU Ja→En
(a)	BASE (§4.1)	44.64	29.30
(b)	BASE + LCD (§4.2)	45.38	23.22
(c)	LeCA (§4.3)	53.79	41.88
(d)	LeCA + LCD	55.49	43.33
(e)	(d) × 8 ensemble (§4.6)	**56.57**	**44.34**
(f)	[(d) + fine-tuning (§4.5)] × 8	56.47	44.28

Table 6: The performance of the submitted systems. According to §4.5, we used only ASPEC 2M for En→Ja and ASPEC 2M + synth 2M for Ja→En. For En→Ja, we show BLEU scores with `MeCab` tokenizer. Bold values indicate the highest score in each column.

	Automatic Eval.		Human Eval.			
Language pair	Final score	(Rank)	DA	(Rank)	CA	(Rank)
En→Ja	57.2	(1)	77.5	(1)	79.7	(1)
Ja→En	44.1	(1)	75.6	(1)	74.4	(1)

Table 7: Official results of our team. The definition of the final score is described in §2. Human evaluations are based on source-based direct assessment (DA) (Cettolo et al., 2017; Federmann, 2018) and source-based contrastive assessment (CA) (Sakaguchi and Van Durme, 2018; Federmann, 2018).

the constraint "superconductivity single phase autotransformer." One likely reason for this is that the baseline model generated a phrase that was quite similar to the constraint in the early phase (marked with a wavy line in Figure 3), and thus the model considered the constraint as translated.

In contrast, LeCA+LCD successfully translated the sentence with the constraints. We believe this is because the LeCA model correctly gives higher scores to the constraint phrases compared to the baselines, helping to generate a sentence with constraints.

Figure 4 shows the BLEU scores of En→Ja translation decoding with various beam sizes. As mentioned in §4.2, the beam size of BASE+LCD needs to be larger than 60 to successfully generate all of the constraints. In contrast, LeCA+LCD can generate all of the constraints and improve the translation accuracy even when their beam size is quite small. This result indicates that the output of LeCA is helpful for LCD to score the candidates and that LeCA can save inference time.

7 Related Work

Hokamp and Liu (2017) proposed Grid Beam Search (GBS), an extended beam search algorithm that forces the NMT model to output pre-specified lexical constraints of words or phrases. At each decoding step, a beam is allocated to each number of constraints, and the top-k candidates that contain n constraints are selected for the n^{th} beam. Translations that satisfy the constraints appear in the beam corresponding to the number of constraints. The beam size changes depending on the number of constraints for each sentence, which makes batch decoding difficult. Post and Vilar (2018) proposed Dynamic Beam Allocation (DBA), which dynamically allocates the beam with a fixed size and improves decoding more efficiently. However, the distribution of the number of constraint tokens in the experiments of these papers was much smaller than that of this task, and we found these methods did not perform well on this task.

Song et al. (2020) and Chen et al. (2021) proposed lexically constrained decoding given explicit alignment guidance between the constraints and the source text. Alignments were induced from an additional alignment head or attention weights (Garg et al., 2019), but these methods assumed that gold alignments are given as constraints. To apply these methods to this task, we would have to use an automatic alignment method (e.g., GIZA++, FastAlign) to obtain the alignments, and the translation accuracy might suffer due to alignment error.

Susanto et al. (2020) proposed non-autoregressive NMT for lexically constrained

Source	分路巻線のみに補助巻線を持つ超電導単相単巻変圧器を試作した。
Reference	Superconductivity single phase auto-transformer with auxiliary winding only at the shunt winding was produced experimentally.
Constraints	shunt winding, auxiliary winding, superconductivity single phase auto-transformer
Base+LCD	We have developed a superconducting single - phase transformer with auxiliary windings only in the shunt windings, in which the auxiliary windings are connected to the shunt windings of the single - phase transformer, and the auxiliary windings are connected to the shunt windings of the single - phase transformer with auxiliary windings of the auxiliary windings of the auxiliary windings of the auxiliary windings of the auxiliary windings of the auxiliary windings of the auxiliary windings of the auxiliary windings of the auxiliary windings. Superconductivity single phase auto - transformer is assisted by the auxiliary windings of the auxiliary windings.
LeCA+LCD	A superconductivity single phase auto-transformer with auxiliary winding only in the shunt winding was produced experimentally.

Figure 3: Example translation: Underlines show the matched constraints, and wavy lines show the phrases that the models fail to match.

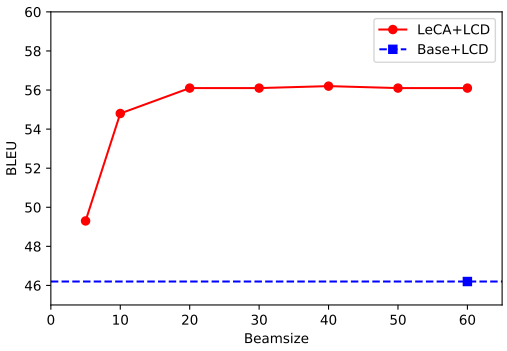

Figure 4: BLEU scores of En→Ja translation decoding with various beam sizes. The BLEU scores are calculated with `sacreblue` (Post, 2018)

translation. They used the Levenshtein Transformer (Gu et al., 2019), which inserts and deletes tokens at each time step, starting from the given constraints as the initial state. They assumed that the order of the given constraints is the same as the order in the reference, but the given constraints in this task appear in random order. Furthermore, they have not achieved comparable translation accuracy to the auto-regressive approaches.

Some works augment the input sequence with constraints. Song et al. (2019) augmented the source sentence by replacing or appending constraints with its corresponding source phrase through leveraging an SMT phrase table. Chen et al. (2020) proposed a simple yet effective augmentation method that appends constraints after the source sentence. Although the decoding speed is fast, Song et al. (2019) relied on the quality of the SMT phrase table. Furthermore, neither of the works could guarantee that the translation would contains all constraints.

8 Conclusion

This paper described the systems that were submitted to the WAT 2021 restricted translation task. We submitted systems for both En→Ja and Ja→En, and both of our systems won the best translation accuracy as assessed by BLEU, the consistency score, and human evaluations. We also confirmed that the data augmentation method makes lexically constrained decoding more effective and, furthermore, that combining data augmentation and constrained decoding significantly improves translation accuracy.

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the 5th Conference on Machine Translation (WMT)*, pages 1–55.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuitho Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 2–14.

Guanhua Chen, Yun Chen, and Victor OK Li. 2021. Lexically constrained neural machine translation with explicit alignment guidance. *Proceedings of the AAAI Conference on Artificial Intelligence*.

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, IJCAI-20*, pages 3587–3593. International Joint Conferences on Artificial Intelligence Organization. Main track.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 86–88, Santa Fe, New Mexico. Association for Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 11179–11189. Curran Associates, Inc.

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.

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. 2006. MeCab: yet another part-of-speech and morphological analyzer. http://mecab.sourceforge.net.

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.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 220–224.

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 (WAT)*, pages 89–94.

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 (WAT 2019)*, pages 99–105.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2020. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC)*, pages 3603–3609.

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, and Sadao Oda, Yusuke Kurohashi. 2021. Overview of the 8th workshop on Asian translation. In *Proceedings of the 8th Workshop on Asian Translation*, Bangkok, Thailand. 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 10th International Conference on Language Resources and Evaluation (LREC)*.

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 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 3rd Conference on Machine Translation (WMT)*, pages 186–191.

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.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 208–218, Melbourne, Australia. Association for Computational Linguistics.

Toshinori Sato. 2015. Neologism dictionary based on the language resources on the web for mecab.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

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 AAAI Conference on Artificial Intelligence*, volume 34, pages 8886–8893.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 2818–2826.

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 (NeurIPS)*, pages 6000–6010.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.