Restricted or Not: A General Training Framework for Neural Machine Translation

Zuchao Li[1,2], **Masao Utiyama**[3,*], **Eiichiro Sumita**[3], **and Hai Zhao**[1,2,*]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[3]National Institute of Information and Communications Technology (NICT), Kyoto, Japan
charlee@sjtu.edu.cn, {mutiyama,eiichiro.sumita}@nict.go.jp, zhaohai@cs.sjtu.edu.cn

Abstract

Restricted machine translation incorporates human prior knowledge into translation. It restricts the flexibility of the translation to satisfy the demands of translation in specific scenarios. Existing work typically imposes constraints on beam search decoding. Although this can satisfy the requirements overall, it usually requires a larger beam size and far longer decoding time than unrestricted translation, which limits the concurrent processing ability of the translation model in deployment, and thus its practicality. In this paper, we propose a general training framework that allows a model to simultaneously support both unrestricted and restricted translation by adopting an additional auxiliary training process without constraining the decoding process. This maintains the benefits of restricted translation but greatly reduces the extra time overhead of constrained decoding, thus improving its practicality. The effectiveness of our proposed training framework is demonstrated by experiments on both original (WAT21 En↔Ja) and simulated (WMT14 En→De and En→Fr) restricted translation benchmarks.

1 Introduction

Neural machine translation (NMT) has recently entered use because of rapid improvements in its performance (Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017). The translation mechanism of an NMT model is a black box because it is a special deep neural network model, which means that translation generation is uncontrollable (Moryossef et al., 2019; Mehta et al., 2020; Miyata and Fujita, 2021). Although uncontrollable (or unguaranteed) translation can satisfy basic requirements, it is unacceptable in some formal scenarios, particularly for key numbers, time, and proper nouns. To address this concern, the restricted translation task has been proposed (Hokamp and Liu, 2017; Post and Vilar, 2018; Song et al., 2019; Chen et al., 2020; Chousa and Morishita, 2021; Li et al., 2021). This restricts translation by forcing the inclusion of prespecified words and phrases in the generation output, which enables explicit control over the system output.

Lexically constrained (or guided) decoding (CD) (Post and Vilar, 2018; Hu et al., 2019b,a), a modification of beam search, has commonly been used in recent restricted translation studies. Although CD is a reasonable option for restricted translation, its slow decoding limits the practicality of restricted translation. Therefore, we propose a novel training framework for restricted translation that requires only minor changes to the ordinary translation model, to address the inconvenience of the decoding time overhead caused by additional constraints. In this framework, restricted machine translation is achieved by the model structure instead of the CD.

Specifically, we perform translation in two modes in the training framework: end-to-end translation and restricted translation, and reuse the self-attention and cross-attention in the decoder of the translation model. To make the restricted translation training mode adapt to the training data situation with only parallel sentences available, we propose the Sampled Constraints as Concentration (SCC) training approach. In this approach, we sample the target sequence to simulate the constraint words and impose additional penalties on the loss of these sampled words.

Because the restricted translation is embedded with the model structure and training objective in the translation model trained with our framework, restricted translation is performed without CD. Consequently, the inference speed is substantially increased, which greatly improves the practicality of restricted translation. Experimental re-

*Corresponding author. This work was partially funded by the Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Student Research Workshop, pages 245 - 251

sults show that our end-to-end translation model can achieve approximately the same performance as the end-to-end translation baseline; moreover, although it only requires unconstrained decoding, it can achieve performance competitive or even superior with that of the baseline with CD.

2 Our Training Framework

Our training framework comprises two training subprocesses: end-to-end translation and restricted translation. Recent restricted translation studies have focused mainly on the decoding phase, but we set out to integrate restricted translation into the training phase, which makes the motivation of our work completely different from that of previous studies. Our implementation is based on the existing mainstream Transformer NMT baseline; however, because the training method is independent of the baseline, our training framework can easily be generalized to other NMT models and language generation tasks. Due to space limitations, please refer to Appendix A.2 for training details.

2.1 End-to-end Translation Training

The most widely adopted form of machine translation is end-to-end translation, which usually employs an encoder–decoder architecture. In the training of end-to-end machine translation, given a source language input $\mathbf{X} = \{x_1, x_2, ..., x_n\}$ and target language translation $\mathbf{Y} = \{y_1, y_2, ..., y_m\}$, the model with parameter θ is trained to generate the target output sequence $\mathbf{Y}$ according to the source input sequence $\mathbf{X}$.

Taking the Transformer model as an example, the encoder is composed of the multi-head self-attention module, whose purpose is to vectorize and contextualize the source input sequence. This module can be formalized as:

$$\mathbf{H}^X = \textit{SelfAttn}_{enc}(\mathbf{X} + \textit{Pos}(\mathbf{X})),$$

where $\textit{Pos}(\cdot)$ represents the position encoding of a sequence, $\textit{SelfAttn}_{enc}$ denotes the stacked multi-head self-attention encoder, and $\mathbf{H}^X$ is the contextualized source representation. A typical decoder comprises two main components: self-attention and cross-attention. In the self-attention component, the target representation is encoded with similar multi-head attention structures,

$$\hat{\mathbf{H}}^Y = \textit{SelfAttn}_{dec}(\textit{IncMask}(\hat{\mathbf{Y}} + \textit{Pos}(\hat{\mathbf{Y}}))),$$

where $\hat{\mathbf{Y}} = \{\textit{BOS}, y_1, y_2, ..., y_{m-1}\}$ is the shifted version of the target sequence $\mathbf{Y}$, $\textit{SelfAttn}_{dec}$ denotes the stacked multi-head self-attention layers (similar to the encoder), and *IncMask* is the extra incremental mask adopted because the sequence on the decoder side is generated incrementally. The target representation is fed to the cross-attention component, as a query, and the source representation is used as the key and value to obtain the final representation, which is then mapped to the target vocabulary space through a linear and softmax layer. The final predicted probabilities can be written as follows:

$$P(\mathbf{Y}) = \textit{Softmax}(\textit{CrossAttn}(\hat{\mathbf{H}}^Y, \mathbf{H}^X)).$$

The model parameter θ is optimized by minimizing the negative log-likelihood of the gold tokens, according to their predicted probabilities:

$$\begin{aligned} \mathcal{L}_{\text{E2E}} &= -\sum_{i=1}^{m} log P(y_i) \\ &= -\sum_{i=1}^{m} log P(y_i|\mathbf{X}; \hat{\mathbf{Y}}_{<i}; \theta), \end{aligned} \tag{1}$$

where $\hat{\mathbf{Y}}_{<i}$ indicates the sequence before token y_i. In the inference stage, greedy (or beam) search is employed to generate the translation sequence according to predicted probabilities $P(y_i) = P(y_i|\mathbf{X}; \tilde{\mathbf{Y}}_{<i}; \theta)$, where $\tilde{\mathbf{Y}}$ is the generated token sequence.

2.2 Restricted Translation Training

In recent work on restricted translation, CD, a modification of beam search, has generally been adopted. In CD, $P(y_i)$ remains unchanged and external search processes are employed, which increases the decoding time overhead. In this paper, we focus on improving the efficiency of restricted translation by modifying $P(y_i)$ to eliminate the additional search processes. Given the constrained word sequence $\mathbf{C} = \{c_1, c_2, ..., c_k\}$, CD adds additional terms to the predicted probability of the model, and $\mathbf{C}$ is treated as an additional input prompt. The output probability $P(y_i)$ then becomes:

$$P(y_i) = P(y_i|\mathbf{X}; \mathbf{C}; \hat{\mathbf{Y}}_{<i}; \theta).$$

According to this change in the form of probability, we made a simple modification to the workflow of the model, keeping the model structure unchanged. First, we encoded the constrained word sequence with the self-attention component of the

decoder. Because the input order of the constrained word sequence is usually inconsistent with the word order of the target sequence, we removed the positional encoding, taking advantage of the position invariance of the self-attention layer. In addition, these constrained words are visible during the entire translation generation process, so there is no need to use the incremental mask strategy. Finally, the constrained words representation is as follows:

$$\mathbf{H}^C = \mathit{SelfAttn}_{dec}(\mathbf{C}).$$

Regarding such a representation as an additional context, outside of the source representation, the predicted probability of the model can be written as:

$$P(\mathbf{Y}) = \mathit{Softmax}(\mathit{CrossAttn}(\hat{\mathbf{H}}^Y, \mathbf{H}^X) + \mathit{CrossAttn}(\hat{\mathbf{H}}^Y, \mathbf{H}^C)). \quad (2)$$

2.3 Sampled Constraints as Concentration

The training of end-to-end NMT models generally uses parallel sentences between source and target languages, whereas restricted machine translation requires an additional constraint sequence. To hide the difference between restricted translation training and testing, we propose the SCC training strategy.

Because restricted machine translation training requires additional given constraint sequences, we randomly sample the target sequence to obtain constrained words in this training strategy. However, this is insufficient. Because these additional target words are already exposed to the decoder, the generation of these tokens would become quite easy, and the goal of fully training the model would not be accomplished (i.e., there are shortcuts). This would have an undesirable impact on end-to-end translation (as when no constrained words are prespecified) and reduce the model's robustness, which is incompatible with our general training framework. Therefore, we propose additional concentration penalties for the losses of these exposed constrained tokens. Denoting the sampled sequence as $\mathbf{S}^Y_\alpha$, where α is the sampling ratio, and the penalty factor as γ, the final loss is:

$$\mathcal{L}_{\text{RT}} = -\sum_{i=1}^{m} \big((1 + \gamma \mathbb{1}(y_i \in \mathbf{S}^Y_\alpha)) \; log P(y_i | \mathbf{X}; \mathbf{S}^Y_\alpha; \hat{\mathbf{Y}}_{<i}; \theta)\big), \quad (3)$$

where $\mathbb{1}(\cdot)$ is the indicator function. Please refer to Appendix A.1 for an illustrated figure and more details.

3 Empirical Evaluation and Analysis

Our method was evaluated on the ASPEC (Nakazawa et al., 2016) En↔Ja benchmark and the WMT14 En→De and En→Fr benchmarks. The constrained words for the ASPEC En↔Ja test set were provided by the WAT21 restricted translation shared task and, for WMT14 En→De and En→Fr, we followed previous work by adopting random sampling to extract the constraints. We chose two typical Transformer model settings as our baseline: Transformer-base and Transformer-big, both of which are consistent with (Vaswani et al., 2017). During training, we set $\alpha = 0.15$ and $\gamma = 1.0$. For a fair comparison, the beam size was set to 10 and the batch size was fixed at 64.

We reported MultiBLEU scores in our experiments and calculated them using the Moses script. For En, De, and Fr, we use the default tokenizer provided by Moses (Hoang and Koehn, 2008), and for Ja, we adopted Mecab[1] for word segmentation. In the evaluation of WAT21 EN↔JA, we also reported a consistency metric – the Exact Match (EM) score - according to the WAT21 official instructions. This score is the ratio of sentences in the whole corpus that exactly match the given constraints. For the EM score evaluation, we use lowercase hypotheses and constraints, then use character-level sequence matching (including whitespaces) for each constraint in En, while for Ja, we use character-level sequence matching (including whitespaces) for each constraint without preprocessing. Please see Appendix A.3 for more preparation details.

3.1 Results and Analysis

We present the performance of the models on the WAT21 En↔Ja restricted translation tasks in Table 1. First, for both model architectures (Transformer-base (T-base) and Transformer-big (T-big)), the end-to-end translation performance (E2E) of our approach's models is almost the same as our baselines. This demonstrates that our training framework still maintains high end-to-end translation performance, even with restricted translation added, meaning it effectively supports both end-to-end translation and restricted translation simultaneously.

Second, on our end-to-end baselines, CD can also be used to accommodate restricted translation. Its very substantial gain in translation performance suggests that CD is a reasonable op-

[1] https://taku910.github.io/mecab/

Model	Alg.	En→Ja	Ja→En	Speed (sent./s)
T-base	E2E	41.82 [26.49]	28.18 [21.96]	53.98 / 63.39
	CD	47.11 [98.29]	31.55 [99.11]	0.74 / 0.78
Ours	E2E	41.87 [26.55]	28.20 [22.01]	53.95 / 63.40
	CAC	47.15 [60.26]	35.46 [56.68]	36.01 / 39.32
	CD+	**47.30 [98.56]**	**35.49 [99.30]**	0.73 / 0.81
T-big	E2E	43.33 [27.51]	29.45 [22.70]	29.68 / 32.53
	CD	47.89 [98.30]	32.04 [99.16]	0.68 / 0.71
Ours	E2E	43.40 [27.60]	29.41 [23.25]	29.55 / 32.21
	CAC	47.93 [60.77]	35.71 [57.42]	18.13 / 19.32
	CD+	**48.01 [98.60]**	**35.75 [99.44]**	0.65 / 0.70

Table 1: Performance on WAT21 En↔Ja test sets. In the form $a[b]$, a represents the BLEU score and b the EM score (see Appendix A.3).

Model	En→De	En→Fr	Speed (sent./s)
(Vaswani et al., 2017)	28.40	41.80	–
T-big (Ours)	28.15	43.12	39.23 / 34.95
+CAC (*rand1*)	29.95	44.27	31.27 / 29.38
+CAC (*rand2*)	31.62	45.53	30.63 / 28.37
+CAC (*rand3*)	33.13	47.21	29.43 / 27.46
+CAC (*rand4*)	34.51	48.16	28.19 / 26.40
+CAC (*phr4*)	36.07	48.95	28.26 / 26.38

Table 2: Performance on WMT14 En→De and En→Fr test sets.

tion for restricted translation. However, under the same conditions, its decoding speed is much lower than that of ordinary decoding, which prevents it from being deployed at a large scale. In our proposed framework, restricted translation is successfully supported with constraints as context (CAC), without using CD. Like CD methods, our method obtains a similar and substantial performance improvement, but it does so without sacrificing too much decoding speed, which demonstrates that our proposed method is efficient and effective.

Because CAC employs constrained word sequences as additional context, it only imposes soft constraints on the decoder, whereas CD imposes hard constraints. However, because CAC and CD do not conflict, we combined the two as CD+ to produce better results. Our experimental findings attest to the effectiveness of this practice. Furthermore, CAC significantly outperforms CD in Ja→En. This may be due to the beam size of 10, which is insufficient for longer constrained sequences and limits CD performance (a larger beam size will be better, see Figure 1(a)), but our proposed CAC alleviates this shortcoming obviously. Furthermore, for the EM score, CD adheres to hard constraints that the given constrained word must appear in the translation, whereas CAC leverages soft constraints and instead focuses on the overall translation, resulting in a higher BLEU for CAC and a higher EM for CD. CD+, however, provides higher scores for both these metrics.

As in previous studies on restricted translation, we also investigated the impact of constrained words on restricted translation. The constrained words were sampled from the translation references of popular translation datasets (WMT14 En→De and En→Fr). There are five common sampling strategies: *rand1*, *rand2*, *rand3*, *rand4*, and *phr4*. *randk* means that the translation is sampled without replacement k times, and *phrk* means that k consecutive words are sampled. For a translation length less than k, an empty string is output because no constrained words are given.

Table 2 compares the end-to-end translation performance of our T-big model with that of Vaswani et al. (2017)'s model. Although we used the same model size and number of training steps, our model's performance was inferior on En→De but superior on En→Fr. This is a consequence of the use of a larger beam size and the potential benefits of restricted translation training on end-to-end translation. The results also show that the translation performance improved dramatically even when only one constrained word was used. This shows that our method of using constraints as a soft restriction is very effective, and it also demonstrates that translation can be improved substantially with some prior knowledge of translation. The disparities between *rand1* and *rand4* show that accurate prior knowledge of translation can lead to more accurate translation, as the translation uncertainty has been gradually reduced. Additionally, comparing *rand4* and *phr4* demonstrates that the continuous sampling of four constrained words can result in a greater performance improvement than the discrete sampling of four constrained words. This is because *phr4* generally carries more useful information than *rand4*.

3.2 Ablation Study

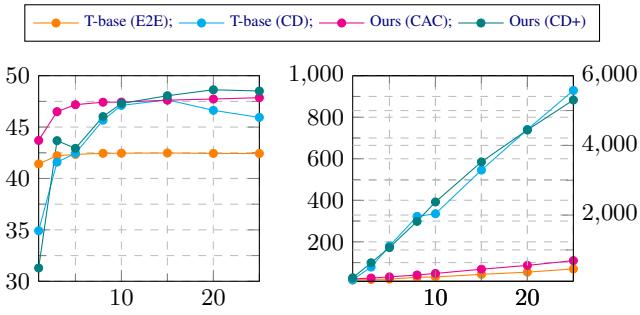
To further demonstrate the advantages of our method, we plotted the performance in BLEU score and total decoding time with different beam sizes in Figure 1. The results of BLEU score vs. beam size show that, for CD methods or variants (CD+), the translation improves at first as the beam size increases. However, after the beam size increases

Model	En→Ja	Ja→En	Speed (sent./s)
T-base (E2E)	41.82	28.18	53.98 / 63.39
Ours (CAC)	47.15	35.46	36.01 / 39.32
- *SCC*	45.63	33.05	36.05 / 39.25
- *RTT*	19.42	10.56	36.07 / 39.30
+ *CPos*	43.36	29.55	35.91 / 39.04
+ *IncMask*	43.78	29.61	35.79 / 38.93

Table 3: Results of ablation study on WAT21 En↔Ja test sets.

beyond a certain threshold, the translation performance decreases. Moreover, we have also observed that CD methods require a larger beam size to outperform beam search methods, and they perform worse when beam size is small; because taking additional constraint words into consideration requires more searching. There is no such issue with our CAC method that employs beam search, however.

Figure 1(b) depicts the total decoding time for various beam sizes. The test set contains 1,812 sentences. We use two y-axes , a larger-scale one on the right to accommodate and denote CD and CD+'s longer decoding times, and a smaller-scale one on the left to denote E2E and CAC's decoding times. The decoding time results show that our CAC method can come close to beam search, a practical restricted translation solution, but CD and CD+ are extremely slow in comparison.



(a) BLEU vs. beam size (b) Time vs. beam size

Figure 1: BLEU score vs. beam size and Decoding time vs. beam size on WAT21 En→Ja test set.

We conducted ablation studies on the model structures and training options of our proposed framework, as shown in Table 3. Using a general MLE loss in restricted translation training; without using SCC loss (-SCC); outperforms the baseline, which shows that the use of restricted translation training can effectively support restricted translation; however, including SCC loss still leads to an improvement over this. This reveals that imposing additional penalties on the loss of constrained words exposed to the decoder is an important design decision. We also evaluated complete removal of the restricted translation training and directly using the end-to-end translation training model for CAC decoding (-RTT). Our results show that the performance greatly suffered, which illustrates the necessity of using restricted translation training for the restricted translation of CAC decoding.

4 Related Work

Lexically constrained (or guided) decoding (CD), a modification of beam search, has commonly been used in recent restricted translation studies. Specifically, some prespecified words or phrases are forced in translation choice. However, although these approaches can theoretically achieve the goal of restricted translation, existing methods are very expensive in terms of decoding time; this limits the practicality of CD. Starting from (Post and Vilar, 2018), in which CD was introduced and utilized in NMT, attempts have been made to reduce the time overhead of CD by the use of dynamic beam allocation. Although the time complexity is formally consistent with that of general beam search, it remains too inefficient to be used on a large scale (Hu et al., 2019b). Hu et al. (2019a) further extended CD and improved the throughput of restricted translation systems by using batching in vectorized dynamic beam allocation. Although these efforts have improved the practicality of restricted translation, the decoding speed is still far less than that of ordinary decoding.

5 Conclusion

In this paper, we proposed novel training and decoding methods for restricted translation that do not use CD. Furthermore, we established a general training framework. With our framework, end-to-end translation and restricted translation can be implemented in the same model. Compared to using CD in the end-to-end translation model, we achieved better translation results, as well as smaller beam size and consistently higher decoding speed. We evaluated the framework on multiple benchmarks, and demonstrated the performance advantages of restricted translation. Using our training framework and decoding method, restricted translation can overcome the limitation of its extremely slow decoding speed and become practical.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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 2020*, pages 3587–3593. ijcai.org.

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.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Hieu Hoang and Philipp Koehn. 2008. Design of the Moses decoder for statistical machine translation. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 58–65, Columbus, Ohio. 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.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019a. Improved lexically constrained decoding for translation and monolingual rewriting. 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 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019b. PARABANK: monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6521–6528. AAAI Press.

Zuchao Li, Masao Utiyama, Eiichiro Sumita, and Hai Zhao. 2021. NICT's neural machine translation systems for the WAT21 restricted translation task. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 62–67, Online. Association for Computational Linguistics.

Sneha Mehta, Bahareh Azarnoush, Boris Chen, Avneesh Saluja, Vinith Misra, Ballav Bihani, and Ritwik Kumar. 2020. Simplify-then-translate: Automatic preprocessing for black-box translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8488–8495. AAAI Press.

Rei Miyata and Atsushi Fujita. 2021. Understanding pre-editing for black-box neural machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1539–1550, Online. Association for Computational Linguistics.

Amit Moryossef, Roee Aharoni, and Yoav Goldberg. 2019. Filling gender & number gaps in neural machine translation with black-box context injection. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 49–54, Florence, Italy. 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 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

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 (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

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.

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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.