Simultaneous Neural Machine Translation with Prefix Alignment

Yasumasa Kano and **Katsuhito Sudoh** and **Satoshi Nakamura**
Nara Institute of Science and Technology, Japan
kano.yasumasa.kw4@is.naist.jp

Abstract

Simultaneous translation is a task that requires starting translation before the speaker has finished speaking, so we face a trade-off between latency and accuracy. In this work, we focus on prefix-to-prefix translation and propose a method to extract alignment between bilingual prefix pairs. We use the alignment to segment a streaming input and fine-tune a translation model. The proposed method demonstrated higher BLEU than those of baselines in low latency ranges in our experiments on the IWSLT simultaneous translation benchmark.

1 Introduction

Simultaneous machine translation (SimulMT) is a task to start outputting translation before observing the whole input sentence. SimulMT is more difficult than the translation with the whole input sentence because it cannot use the latter part of the sentence as context. SimulMT has to decide whether to wait for more input or to output partial translation using the input so far, in real-time. The translation quality should become better if we can use longer inputs and *vice versa*. We have to handle such a trade-off between the quality and latency of the translation by decision *policies* to choose the next action between *read* (waiting for the next input segment) and *write* (outputting a translation segment) for a given input-output history (Gu et al., 2017). Neural Machine Translation (NMT) models used for SimulMT can be roughly categorized into *policy-dependent* and *policy-independent*.

A policy-dependent model is trained with the constraints given by the policy, in order to translate an input prefix into an output prefix. Ma et al. (2019) proposed a simple method with a fixed policy called *wait-k*, where the NMT first takes *k* read actions followed by alternating write and read actions until the end of the translation output. Arivazhagan et al. (2019) proposed a joint training framework for flexible policies and the corresponding NMT model using a latency-augmented loss function and Monotonic Infinite Lookback (MILk) attention.

In contrast, a policy-independent model is a standard NMT model to translate the whole input into the whole output and used for SimulMT along with a given policy in the inference. We can share one NMT model for different policies, so the quality-latency trade-off can be controlled easily. Dalvi et al. (2018) achieved some latency reduction with a small loss in BLEU by the use of a fixed policy called *STATIC-RW*. Ma et al. (2019) also applied their *wait-k* policy using a sentence-based NMT model, called *test-time wait-k*. Zhang et al. (2020) proposed a flexible policy to predict segment boundaries in an input. Once a boundary is found, the segment is translated using a sentence-based NMT model. The model based on their segmentation demonstrated better results in quality-latency trade-off than those using wait-k and MILk in Chinese-to-English SimulMT. Kano et al. (2021) proposed another flexible policy using simple rules with syntactic constituent label prediction and showed better performance than MU-based SimulMT in English-to-Japanese.

One problem in the use of a policy-independent model in SimulMT is the difference between training and inference conditions; the NMT model is trained in the sentence level but is used to translate the prefix of a sentence in inference. This causes unexpectedly long translation and hurts the quality of SimulMT (Kano et al., 2021). To mitigate the problem, we propose a method for data augmentation to fine-tune a policy-independent NMT model to the problem of prefix-to-prefix translation, called *Bilingual Prefix Alignment*. We use a pre-trained sentence-based NMT model to align source language prefix and target language prefix of sentences in the training corpus and collect prefix translation pairs. The proposed method demonstrated higher

Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022), pages 22 - 31

BLEU than baselines in low latency ranges, in our SimulMT experiments using IWSLT English-to-Japanese and English-to-German datasets.

2 Related Work

The problem of SimulMT has been tackled for a decade. In early attempts using statistical machine translation, decision policies were combined with the beam search decoding (Sankaran et al., 2010; Bangalore et al., 2012). Fujita et al. (2013) used phrase reordering probabilities used in phrase-based statistical machine translation for their decision policy. In later years, feature-based learned policies were proposed. Oda et al. (2014) proposed a feature-based policy optimization to maximize BLEU. Syntactic features also successfully used for the policies (Rangarajan Sridhar et al., 2013; Oda et al., 2015).

Recently, most SimulMT studies are based on NMT, and such methods can output more fluent translation than before. Among NMT-based SimulMT studies, one major approach is to train an NMT model optimized for given or jointly-learned policies. Wait-k (Ma et al., 2019) is a very simple fixed policy that waits for *k* input tokens first. Zheng et al. (2020) proposed an ensemble of different wait-*k*-based models for adaptive SimulMT. To make the policies more flexible, latency-augmented loss functions are used to jointly optimize accuracy and latency in the training of the SimulMT model (Raffel et al., 2017; Arivazhagan et al., 2019; Ma et al., 2020b).

Another approach employs such policies only in inference, using a standard sentence-based NMT model. Fixed policies can be applied to this approach easily (Dalvi et al., 2018; Ma et al., 2019). Cho and Esipova (2016) proposed greedy decoding with policies conditioned by the decoder's prediction, called *Wait-If-Worse* and *Wait-If-Diff*. Kano et al. (2021) proposed a rule-based policy using incremental prediction of the syntactic constituents. To learn segmentation policies from the bilingual corpus, reinforcement learning-based methods were proposed (Grissom II et al., 2014; Satija and Pineau, 2016; Gu et al., 2017; Alinejad et al., 2018). It is a straightforward way to optimize latency and accuracy jointly, but its training process is relatively complex and sometimes unstable. Instead of the joint learning of a segmentation policy and policy-dependent model, Zheng et al. (2019) proposed a method to find oracle read and write actions using a pre-trained NMT model. Zhang et al. (2020) also used a pre-trained NMT model to find segments called Meaningful Units (MUs).

This work is motivated by Dalvi et al. (2018) and Zhang et al. (2020) and extends them with Bilingual Prefix Alignment using a pre-trained NMT model. Our method finds appropriate segment boundaries based on the similarity between reference and translation hypothesis for given prefix segments in a different way from Zhang et al. (2020). We also fine-tune the pre-trained NMT model using the bilingual prefix pairs, which is a more sophisticated way than Dalvi et al. (2018)[1].

3 Simultaneous Machine Translation

A sentence-level NMT is formulated as follows, letting $\boldsymbol{x} = x_1, x_2, ..., x_n$ be an input sentence and $\boldsymbol{y} = y_1, y_2, ..., y_m$ be its translation:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{m} P(y_t|\boldsymbol{x}, \boldsymbol{y}_{<t}). \tag{1}$$

SimulMT takes a prefix of the input for its incremental decoding, formulated as follows:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{m} P(y_t|\boldsymbol{x}_{\leq g(t)}, \boldsymbol{y}_{<t}), \tag{2}$$

where $g(t)$ is a monotonic non-decreasing function that represents the number of input tokens read by the t-th step so that $\boldsymbol{x}_{\leq g(t)}$ means an input prefix given so far, and $\boldsymbol{y}_{<t}$ is a prefix translation by the previous step. This means that we can obtain a pair of a input prefix and the corresponding prefix translation $(\boldsymbol{x}_{\leq g(t)}, \boldsymbol{y}_{\leq t})$ at t-th step.

In this work, we use chunk-based incremental decoding (Kano et al., 2021), in which we translate an input prefix from the beginning. It is similar to an approach called *re-translation* (Niehues et al., 2016; Arivazhagan et al., 2020), but we force the decoder to follow already translated output prefixes in the same way as the teacher forcing in NMT training.

4 Proposed Method

Figure 1 shows the whole translation process of the proposed method at the inference step. We propose Prefix Alignment for training a segmentation policy and fine-tuning a sentence-level NMT model for the policy-dependent SimulMT. Suppose we have a

[1] Note that the authors reported they obtained no performance improvement by the fine-tuning.

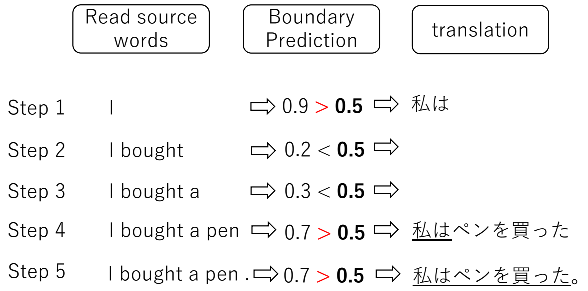

Figure 1: The translation process of the proposed method from English to Japanese. The threshold of boundary probability is 0.5 in this case. The underlined part is the forced output prefix.

pre-trained NMT model and a bilingual corpus for fine-tuning the model for SimulMT. The proposed method consists of the following steps:

1. Collect prefix translation pairs using the pre-trained model
2. Find reference prefixes corresponding to the prefix translation pairs
3. Train a boundary prediction model
4. Fine-tune the NMT model

Their details are described in the following subsections.

4.1 Collecting Prefix Translation Pairs

In this step, we collect *prefix translation pairs* from the bilingual corpus using the pre-trained NMT model. For every source language sentence in the bilingual corpus, we extract prefix translation pairs using NMT results of the source language sentence, by the following procedure. First, we translate the source language sentence $\boldsymbol{x}$ into the target language sentence $\boldsymbol{y}$ using the NMT model. Then, we translate a prefix of $\boldsymbol{x}$ with one word[2], $\boldsymbol{x}_{|w|\leq 1}$, into a target language prefix $\bar{\boldsymbol{y}}^{(1)}$. Here, if the *longest common prefix* $\bar{\boldsymbol{y}}^{(1)}_{lcp}$ between $\boldsymbol{y}$ and $\bar{\boldsymbol{y}}^{(1)}$ is not empty, we extract the pair $(\boldsymbol{x}_{|w|\leq 1}, \bar{\boldsymbol{y}}^{(1)}_{lcp})$ as a prefix translation pair. We iterate this prefix translation pair extraction with enlarging the prefix length one by one; we translate the i-word prefix $\boldsymbol{x}_{|w|\leq i}$ into $\bar{\boldsymbol{y}}^{(i)}$ and check $\bar{\boldsymbol{y}}^{(i)}_{lcp}$. In the iteration, we may obtain the same longest common prefix with different source language prefixes. We just extract the first appearance and ignore the rest with longer source language prefixes in such cases. Furthermore, once we extract a prefix translation pair $(\boldsymbol{x}_{|w|\leq i}, \bar{\boldsymbol{y}}^{(i)}_{lcp})$, we use the target language prefix $\bar{\boldsymbol{y}}^{(i)}_{lcp}$ as a forced output prefix and applied it to update the sentence-level translation $\boldsymbol{y}$ and to generate prefix translation $\bar{\boldsymbol{y}}^{(j)}$ for $j > i$. This is because the translation for longer prefixes or the whole sentence may change by a beam search when a forced output prefix is given.

Our prefix extraction strategy is different from that by Zhang et al. (2020), in which the whole prefix translation $\bar{\boldsymbol{y}}^{(i)}$ should be a prefix of the sentence-level translation $\boldsymbol{y}$, not taking the longest common prefix as in this work.

Figure 2 shows an example. The first prefix translation ends with a punctuation mark, so Meaningful Unit (Zhang et al., 2020) cannot extract the first prefix as the pair because the mark does not match with the end of prefix of full-sentence translation. In contrast, the proposed method can extract the matched target prefix by ignoring the latter part of the prefix translation. Therefore, the proposed method identifies more boundaries than Meaningful Unit.

Another difference from Meaningful Unit relates to the extraction strategy above. Since the original pre-trained NMT model often generates unnecessary tokens like punctuation marks at prefix boundaries, we fine-tune the pre-trained model using the extracted prefix pairs to avoid such problems.

4.2 Prefix Alignment with References

Since the prefix translations obtained through the process above are NMT results and different from their references in general, we also extract corresponding reference prefixes from the bilingual corpus. We use BERTScore (Zhang* et al., 2020) to find the correspondence between an NMT-based prefix and a reference prefix, varying the length of the reference prefix. We choose the reference prefix that has the largest BERTScore F-measure as the corresponding one to a given NMT-based prefix. Using this correspondence, we can align a source language prefix and its reference counterpart to make bilingual prefix alignment.

4.3 Training a Boundary Predictor

We train a boundary predictor for the chunk-based SimulMT using the extracted source language pre-

[2]Here, we use the word-based prefix length even though we use subwords. Thus, $\boldsymbol{x}_{|w|\leq 1}$ may consists of one or more subwords.

Source Prefix	Source prefix Translation	Full-sentence translation	Extracted Target Prefix	Boundary
I	私は。	私はペン買った。	私は	1
I bought	私は買った。	私はペンを買った。		0
I bought a	私は買った。	私はペンを買った。		0
I bought a pen	私はペンを買った	私はペンを買った。	私はペンを買った	1
I bought a pen .	私はペンを買った。	私はペンを買った。	私はペンを買った。	1

Figure 2: Extract Prefix Alignment

fixes. It is a binary classifier, and its training data consist of pairs of a source language sentence prefix and the boundary label. The label is set to 1 for the prefixes in the extracted prefix translation pairs and 0 for the other possible prefixes of the corresponding source sentence, as shown in Figure 2.

4.4 Fine-Tuning a SimulMT Model

We fine-tune the pre-trained NMT model using the extracted bilingual prefix pairs for our SimulMT model. The model is used to translate an input incrementally in the chunk-based manner as presented in Section 3.

5 Experimental Setup

We conducted experiments on English-to-German (En-De) and English-to-Japanese (En-Ja) simultaneous translation to compare the proposed method with the baselines in the quality-latency trade-off.

5.1 Dataset and Preprocessing

In En-De translation, we used WMT 2014 training set (4.5 M sentence pairs) for pre-training and IWSLT 2017 training set (206 K sentence pairs) for fine-tuning. We used IWSLT dev2010, tst2010, tst2011 and tst2012 (5,589 sentence pairs in total) for the development dataset. We used 1,080 sentence pairs from IWSLT tst2015 for the evaluation.

In En-Ja translation, we used WMT 2020 (17.9 M sentence pairs) for pre-training and IWSLT 2017 (223 K sentence pairs) for fine-tuning dataset. We used IWSLT dev2010, tst2011, tst2012, and tst2013 (5,312 sentence pairs in total) for development dataset. We used 1,442 sentence pairs from IWSLT dev2021 for the evaluation.

Prefix translation pairs are collected only from the IWSLT dataset. We tokenized Japanese sentences using MeCab (Kudo, 2005). English and German sentences were tokenized using `tokenizer.perl` in Moses (Koehn et al., 2007). We prepared a shared subword vocabulary with 16 K entries based on Byte Pair Encoding (BPE) (Sennrich et al., 2016) for each language pair.

5.2 Model Settings

We mainly compared the following four methods in the experiments:

Prefix Alignment The proposed method has a hyperparameter to adjust latency, the threshold of boundary probability output by the boundary predictor. We used 0.5 as the default value for the binary classification and tried the following values for further investigation: [0.1, 0.15,..., 0.95], [0.99, 0.991, 0.992,..., 0.999], and [0.9991, 0.9992,..., 0.9999]. We also compared a one look-ahead boundary predictor that took one future word as the input at the cost of the delay in one word (`PA-1`), in addition to a standard (no look-ahead) boundary predictor (`PA-0`).

Meaningful Unit We used the same boundary probability thresholds as in PA. We implemented the refined version of MU-based method to translate with low latency following (Zhang et al., 2020), but did not apply the removal of monotonic translation examples following Kano et al. (2021). We also compared one look-ahead (`MU-1`) and no look-ahead (`MU-0`) boundary predictors.

Incremental Constitutent Label Prediction (`ICLP`) Following Kano et al. (2021), we used a one look-ahead label predictor. We segmented the input sequence based on their rules with the predicted labels `VP` and `S`. The minimum segment length adjusts latency. The range is [1, 2, 3, ..., 29].

`Wait-k` We tried [2, 4, 6, ..., 30] for the hyperparameter k.

NMT Settings We trained a standard NMT model (`full-sentence`) using WMT and

IWSLT training dataset. This model was used for MU, PA and ICLP as the pre-trained NMT model.

All the NMT models were based on Transformer-base (Vaswani et al., 2017) implemented with fairseq (Ott et al., 2019). Their hyperparameter settings basically followed the official baseline for IWSLT 2021[3], for both pre-training and fine-tuning. The models were saved on checkpoints in every 5,000 updates for pre-training and every 200 updates for fine-tuning. We applied early stopping with the patience for four checkpoints, based on the loss on the development set. We set the learning rate to 0.0007, minibatch size to 4,096 with the parameter update frequency of 4. We applied a chunk-based beam search for the methods other than wait-k, in which the low-scored hypotheses out of the specified beam size were eliminated at the end of the chunk. We used greedy-decoding for wait-k, due to the nature of its model.

Boundary Predictor The boundary predictors for the chunk-based methods were implemented similarly using BERT (Devlin et al., 2019) with a pre-trained model bert-base-uncased and the corresponding subword tokenizer from Huggingface transformers (Wolf et al., 2020). We set the learning rate to 5e-5 and the batch size to 512 instances. The models were saved at every epoch, and we applied early stopping with patience for three epochs based on the loss on the development set.

5.3 Evaluation Metrics

We used BLEU (Papineni et al., 2002) and Average Lagging (AL) (Ma et al., 2019) for our quality and latency evaluation metrics. They were calculated using SimulEval (Ma et al., 2020a) and drawn in scatterplots to show the quality-latency trade-off.

6 Results

6.1 English-to-German

Figure 3 shows the BLEU and AL results in English-to-German simultaneous translation. The proposed method (PA-0 and PA-1) showed best performance among the compared methods. On the other hand, the other chunk-based SimulMT (MU-0, MU-1, and ICLP) did not outperform

[3]https://github.com/pytorch/fairseq/blob/master/examples/simultaneous_translation/docs/enja-waitk.md, https://github.com/pytorch/fairseq/issues/346

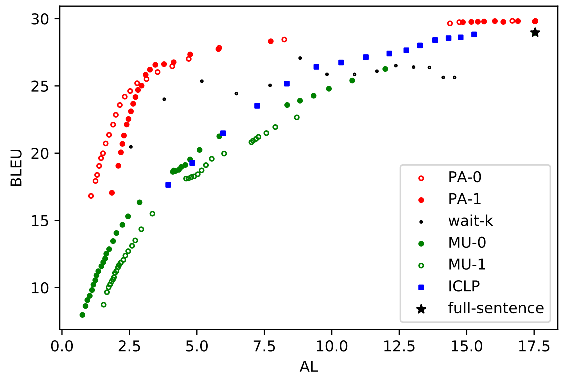

Figure 3: BLEU and Average Lagging (En-De)

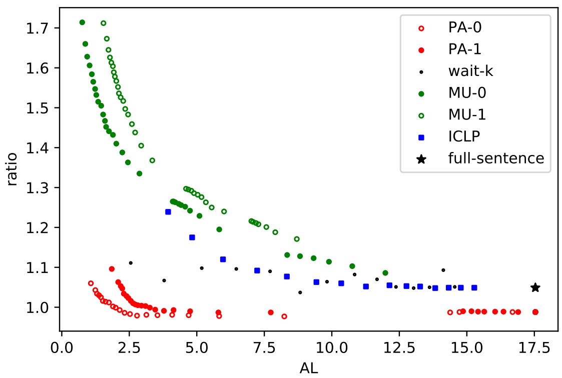

Figure 4: Length ratio and Average Lagging (En-De)

Wait-k. We can also see the look-ahead boundary prediction did not improve BLEU both for PA and MU but increased AL.

Figure 4 shows the results in the length ratio between a translation result and its reference. The proposed method demonstrated better results in the translation length than the other methods. The other chunk-based SimulMT methods generated much longer translation results than the references and resulted in a large drop in BLEU due to the brevity penalty.

6.2 English-to-Japanese

Figure 5 shows the BLEU and AL results in English-to-Japanese simultaneous translation. This shows a large difference from the results in English-to-German; the proposed method outperformed the baselines in very small latency ranges around AL of 2, but showed worse BLEU in the large latency ranges.

Figure 6 shows the results in the length ratio. The proposed method generated shorter transla-

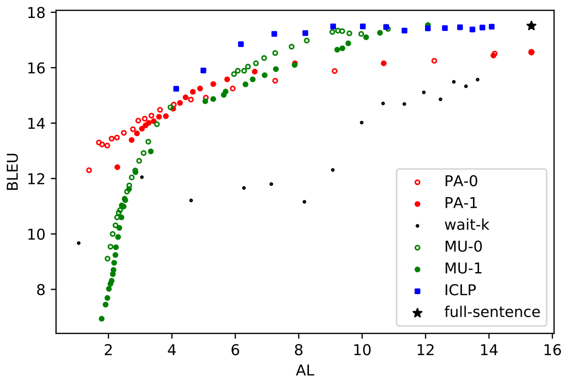

Figure 5: BLEU and Average Lagging (En-Ja)

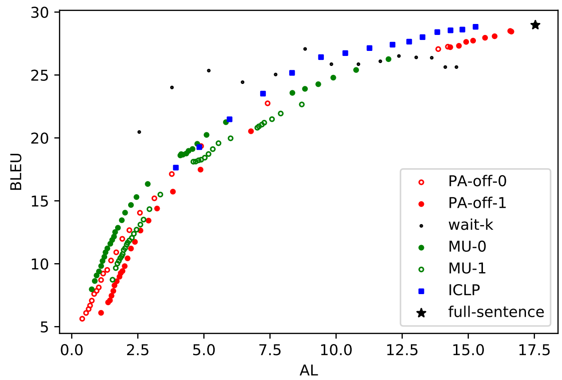

Figure 7: BLEU and Average Lagging (En-De) without PA-based NMT fine-tuning

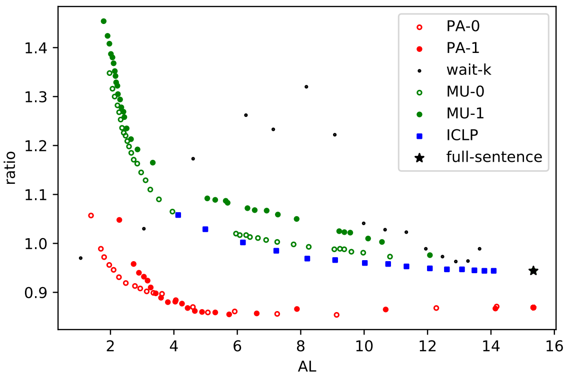

Figure 6: Length ratio and Average Lagging (En-Ja)

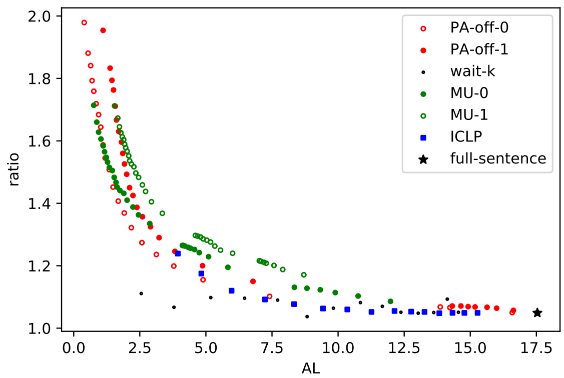

Figure 8: Length ratio and Average Lagging (En-De) without PA-based NMT fine-tuning

tion results especially with the large latency ranges, even though the other methods resulted in a better length ratio of around 1.0. The difference between the two language directions would come from the length issue; the full-sentence NMT resulted in the length ratio slightly larger than 1.0 in English-to-German and around 0.9 in English-to-Japanese. The proposed method encouraged to shorten the translation length in general so that it did not contribute to the BLEU improvement in English-to-Japanese.

7 Analysis

7.1 Effect of PA-based NMT fine-tuning

For the detailed analyses, we investigated the performance of the chunk-based SimulMT without the fine-tuning using the bilingual prefix pairs. Here, only the boundary predictor was used to segment the input for the chunk-based SimulMT. Figures 7, 8, 9, and 10 show the results by the proposed method with the pre-trained NMT model (`PAoff-0` and `PAoff-1`). They clearly show the proposed method does not work well without fine-tuning the NMT model; it resulted in a longer translation length so BLEU decreased due to the brevity penalty. These results suggest the segmentation policy in the chunk-based SimulMT should match the prefix translation models because a full-sentence translation model often generates a too-long translation result for a short prefix input.

7.2 Length Distribution in training dataset

	En-De	En-Ja
# Source prefixes	1,874,909	1,059,865
# Words in sentences	4,228,604	4,593,194

Table 1: Statistics of the training data

We investigated the length issue on the training data. Table 1 shows statistics of the IWSLT training set, in the number of source language prefixes extracted for the fine-tuning of the SimulMT models

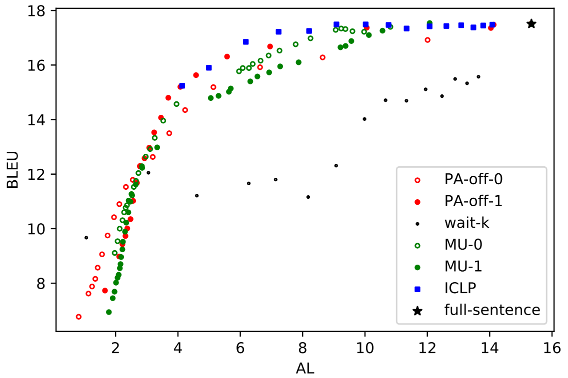

Figure 9: BLEU and Average Lagging (En-Ja) without PA-based NMT fine-tuning

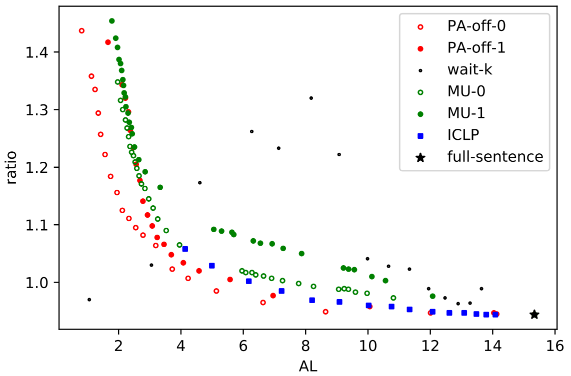

Figure 10: Length ratio and Average Lagging (En-Ja) without PA-based NMT fine-tuning

and the number of words in the whole sentences.

Even though the number of words is almost similar, the number of prefixes is largely different; that in En-De is almost two times larger than that in En-Ja. This is because of the large word order difference between English and Japanese, compared to that between English and German. The word order difference should cause poor prefix matches in the prefix translation pair extraction, so just a few short prefix pairs are found. Figure 11 shows the source prefix length distribution in the IWSLT training data. The peak of the En-Ja distribution is to the right of that of En-De distribution because of this word order difference. The number of the En-De shortest prefixes is more than three times larger than that of En-Ja ones. This large number of short prefixes contributed to the improvement of En-De SimulMT.

Figures 12 and 13 show the change of length distribution of the training data; blue bars represent

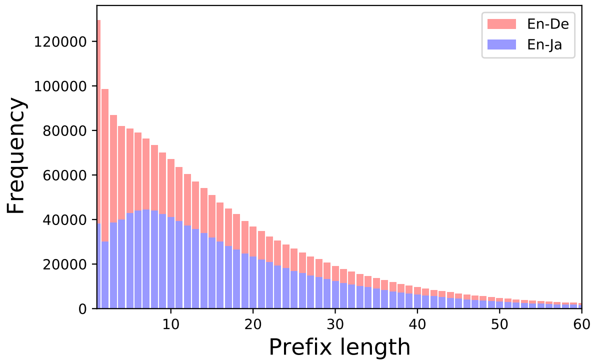

Figure 11: Source prefix length distribution in the IWSLT training data

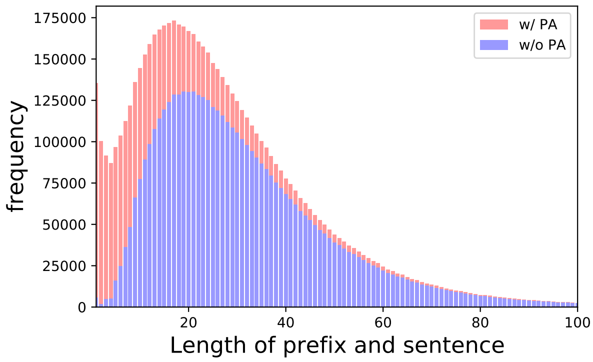

Figure 12: Source sentence length distribution in the training data (En-De)

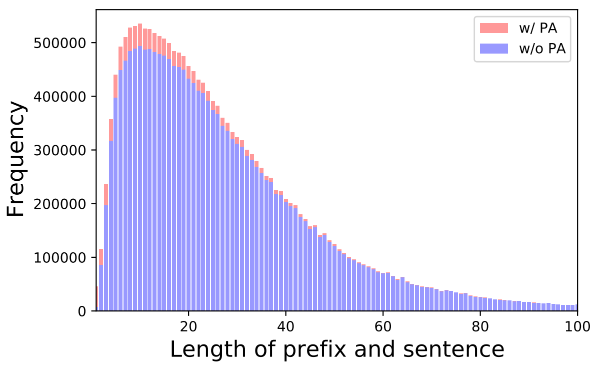

Figure 13: Source sentence length distribution in the training data (En-Ja)

the original distribution on the whole training data (WMT and IWSLT), and red bars represent that on the training data augmented by the additional prefix pairs. The change in English-to-German was much larger than that in English-to-Japanese, because of the large difference in the number of bilingual prefix pairs. These findings suggest the proposed method had a larger effect in English-to-German than English-to-Japanese.

8 Conclusion

We proposed a method to train the neural SimulMT model by extracting bilingual prefix pairs by Prefix Alignment. The proposed method outperformed the baselines in quality-latency trade-off in English-to-German simultaneous translation but showed mixed results in English-to-Japanese. We investigated the results in detail and found the difference in the translation length made a large effect on the results, caused by the performance of the sentence-level NMT model and the word order difference.

In future work, we extend the method to work for language pairs with the large word order differences such as English-Japanese, in the wide range of AL. The proposed method to extract source prefixes can be adapted to speech input. We applied this method to Speech-to-text simultaneous machine translation system submitted to the IWSLT 2022 Evaluation Campaign (Anastasopoulos et al., 2022; Fukuda et al., 2022).

Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Numbers JP21H05054 and JP21H03500.

References

Ashkan Alinejad, Maryam Siahbani, and Anoop Sarkar. 2018. Prediction improves simultaneous neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3022–3027, Brussels, Belgium. Association for Computational Linguistics.

Antonios Anastasopoulos, Luisa Bentivogli, Marcely Z. Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Marcello Federico, Christian Federmann, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel M. Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, Juan Pino, Elizabeth Salesky, Jiatong Shi, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alex Waibel, Changhan Wang, and Shinji Watanabe. 2022. FINDINGS OF THE IWSLT 2022 EVALUATION CAMPAIGN. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, Dublin, Ireland. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, I Te, Wolfgang Macherey, Pallavi Baljekar, and George Foster. 2020. Re-Translation Strategies for Long Form, Simultaneous, Spoken Language Translation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7919–7923.

Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan, Ladan Golipour, and Aura Jimenez. 2012. Real-time incremental speech-to-speech translation of dialogs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 437–445, Montréal, Canada. Association for Computational Linguistics.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *arXiv preprint arXiv:1606.02012*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation in neural machine translation. 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 493–499, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2013. Simple, Lexicalized Choice of Translation Timing for Simultaneous Speech Translation. In *Proc. Interspeech 2013*, pages 3487–3491.

Ryo Fukuda, Yuka Ko, Yasumasa Kano, Kosuke Doi, Hirotaka Tokuyama, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura. 2022. NAIST Simultaneous Speech-to-Text Translation System for IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, Dublin, Ireland. Association for Computational Linguistics.

Alvin Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. 2014. Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1342–1352, Doha, Qatar. Association for Computational Linguistics.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2021. Simultaneous neural machine translation with constituent label prediction. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1124–1134, Online. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej 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 Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2005. Mecab : Yet another part-of-speech and morphological analyzer. `http://mecab.sourceforge.net/`.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 144–150, Online. Association for Computational Linguistics.

Xutai Ma, Juan Miguel Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020b. Monotonic Multihead Attention. In *International Conference on Learning Representations*.

Jan Niehues, Thai Son Nguyen, Eunah Cho, Thanh-Le Ha, Kevin Kilgour, Markus Müller, Matthias Sperber, Sebastian Stüker, and Alex Waibel. 2016. Dynamic Transcription for Low-Latency Speech Translation. In *Interspeech 2016*, pages 2513–2517.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Optimizing segmentation strategies for simultaneous speech translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 551–556, Baltimore, Maryland. Association for Computational Linguistics.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Syntax-based simultaneous translation through prediction of unseen syntactic constituents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 198–207, Beijing, China. Association for Computational Linguistics.

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.

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.

Colin Raffel, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. 2017. Online and Linear-Time Attention by Enforcing Monotonic Alignments. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *JMLR Workshop and Conference Proceedings*, pages 2837–2846. JMLR.org.

Vivek Kumar Rangarajan Sridhar, John Chen, Srinivas Bangalore, Andrej Ljolje, and Rathinavelu Chengalvarayan. 2013. Segmentation strategies for streaming speech translation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Lan-*

guage Technologies, pages 230–238, Atlanta, Georgia. Association for Computational Linguistics.

Baskaran Sankaran, Ajeet Grewal, and Anoop Sarkar. 2010. Incremental decoding for phrase-based statistical machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 216–223, Uppsala, Sweden. Association for Computational Linguistics.

Harsh Satija and Joelle Pineau. 2016. Simultaneous machine translation using deep reinforcement learning. In *Workshops of International Conference on Machine Learning*, page 110–119.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2020. Learning adaptive segmentation policy for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2280–2289, Online. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. 2020. Simultaneous translation policies: From fixed to adaptive. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2847–2853, Online. Association for Computational Linguistics.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019. Simpler and faster learning of adaptive policies for simultaneous translation. 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 1349–1354, Hong Kong, China. Association for Computational Linguistics.