General Paper Peer-Reviewed

Prefix Alignment for Training Simultaneous Machine Translation

Yasumasa Kano[†], Katsuhito Sudoh[†] and Satoshi Nakamura[†]

Simultaneous translation is a task that starts translation even before the speaker has finished speaking. This study focuses on prefix-to-prefix translation and proposes a method to align prefixes in a bilingual sentence pair iteratively to train a machine translation model to work with prefix-to-prefix. In the experiments, the proposed method demonstrated higher BLEU than those of the baseline methods in low latency ranges on the IWSLT simultaneous translation benchmark. However, the proposed method degraded the performance in high latency ranges in the English-to-Japanese experiments; thus, we analyzed it in length ratios and prefix boundary prediction accuracies. The obtained results suggested that the degraded performance was due to the large word order difference between English and Japanese.

Key Words: *Neural Machine Translation, Simultaneous Translation*

1 Introduction

Simultaneous machine translation (SimulMT) is a task used to translate input text or speech incrementally even before observing its explicit boundaries, such as the end of sentences and pauses. It has to determine when to translate a partially-observed input (*input prefix*) in real-time and also when to generate an appropriate partial output (*output prefix*) corresponding to the partial input. This requirement poses a serious difficulty compared to standard full-sentence translation because the remaining *unobserved* input cannot be used as context information. The SimulMT quality should improve using a longer input at the cost of a long delay, and vice versa. This trade-off is controlled by a *policy* to select the next action between *read* (waiting for further partial input) and *write* (emitting partial output) according to the given input and generated output prefixes (Gu et al. 2017).

Wait-k (Ma et al. 2019) is one of the most commonly used baseline SimulMT models that waits for k tokens before starting translation. There are also more adaptive models that determine whether to *read* or *write* every time a new input token becomes available (Chiu and Raffel 2018; Raffel et al. 2017; Arivazhagan et al. 2019). These methods train the SimulMT model with the

[†] Nara Institute of Science and Technology

given policy to learn the prefix-to-prefix translation. The latency amount of these SimulMT models is provided as a hyperparameter of training; thus it is difficult to adjust latency at the inference step.

In contrast, there are SimulMT methods that can share one pre-trained NMT model for any latency and the hyperparameter of latency is provided at the inference step. For instance, Zhang et al. (2020a) proposed to segment the input sequence with a unit known as Meaningful Unit (MU) and translate the partial input. The translation is conducted with a standard NMT model pre-trained with full sentences in their method. This causes serious over-translation because the NMT model translates prefixes that is shorter than full sentences at the inference step of SimulMT. (Kano et al. 2021).

In this study, we propose to make the NMT model learn prefix-to-prefix translation by fine-tuning a standard NMT model with data augmentation called *Prefix Alignment*. We use a pre-trained full-sentence NMT model to find bilingual prefix pairs from a bilingual training corpus through the iteration of prefix translation and the matching of common prefixes between the prefix translation result and the corresponding reference in the training corpus. The NMT model fine-tuned with the collected bilingual prefix pairs can be used with any segmentation methods. The proposed fine-tuning method improved the SimulMT models with several segmentation methods in low latency ranges in our SimulMT experiments using IWSLT English-to-Japanese (En-Ja) and English-to-German (En-De) datasets. We also analyzed the effect of word order difference to explain the degradation by the proposed method in high latency ranges in En-Ja.

2 Related Work

The problem of SimulMT has been tackled for a decade. In previous 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 employed 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 were 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 translations than before. Cho and Esipova (2016) proposed greedy decoding with policies conditioned by the prediction of the decoder, known as *Wait-If-Worse* and *Wait-If-Diff*. Ma et al. (2019) proposed a simple policy known as test-time wait-k. This is a very simple fixed

policy that waits for k input tokens, and thereafter writes one target token and reads one source token alternately. The latency hyperparameter k of test-time wait-k is provided at the inference step. To learn the policy of READ/WRITE decision 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. Therefore, Zheng et al. (2019) proposed a simpler method to find oracle read and write actions using a pre-trained NMT model. These previous NMT-based SimulMT models used a standard NMT model pre-trained for full-sentence translation. However, at the inference step of SimulMT, the model needed to translate partial input; thus there was a gap between training and inference.

Therefore, Ma et al. (2019) proposed a wait-k that uses prefix-to-prefix training of a SimulMT model with full-sentence parallel corpus. The policy of wait-k is the same as that of test-time wait-k, but the latency hyperparameter k of wait-k is provided in the training step. Zheng et al. (2020) proposed an ensemble of different wait-k-based models for adaptive SimulMT. There are other approaches for training a SimulMT model that constrain the attention to the last part of the partial input (Chiu and Raffel 2018; Raffel et al. 2017). The attentions that are extended to all the tokens in the partial input and latency-augmented loss functions are proposed to jointly optimize translation quality and latency (Arivazhagan et al. 2019; Ma et al. 2020b). The latency amount of these SImulMT models is provided as a training hyperparameter similar to wait-k. These approaches make the conditions of training and inference closer, but we cannot adjust the latency amount at the inference step and need to train the SimulMT model for each latency.

Zhang et al. (2020a) proposed a method to segment the input sequence with a unit known as MU. In their SimulMT model, the standard pre-trained full-sentence NMT model translates the partial input every time a boundary of MU is detected. Their method can adjust latency at the inference step by a hyperparameter and outperformed wait-k and MILk (Arivazhagan et al. 2019). Kano et al. (2021) proposed incremental prediction of the syntactic constituents to improve the segmentation. These methods (Zhang et al. 2020a; Kano et al. 2021) use the NMT model pre-trained with full sentences; thus, there is a gap between training and inference.

There are some approaches to mitigate the gap between training and inference by fine-tuning the NMT model. Zhang et al. (2020a) proposed a refined version of MU in which they fine-tuned the pre-trained NMT model with synthetic monotonic translation sentences. However, even though the word order of the source and target becomes similar by their method, the synthetic data are sentence-length which is longer than partial inputs. Dalvi et al. (2018) attempted to fine-tune the NMT model with bilingual prefix pairs. However, they reported that the fine-tuning

largely degraded the performance in the simultaneous translation experiment. To generate the bilingual prefix pairs, they extracted the first N words, N + M words, N + 2M words,$\cdots$ from each source sentence as prefixes. Thereafter they found the corresponding target prefixes from each target sentence.

Our proposed method uses a more sophisticated technique to generate bilingual prefix pairs utilizing a pre-trained NMT model and improves the NMT model with any segmentation methods.

3 Simultaneous Machine Translation

Before explaining the proposed method, we first introduce the basics of simultaneous translation.

3.1 Formulation

A sentence-level NMT finds the best translation that maximizes the conditional probability 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:

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

In SimulMT, we take a prefix of the input for its incremental decoding 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 representing the number of input tokens to read until the t-th step. $\boldsymbol{x}_{\leq g(t)}$ denotes an input prefix read so far, and $\boldsymbol{y}_{<t}$ is the prefix translation written until the previous step. This implies that we can align the prefix translation $\boldsymbol{y}_{\leq t}$ with the input prefix $\boldsymbol{x}_{\leq g(t)}$ as a pair of input-output prefixes.

3.2 Inference of MU

Figure 1 shows the inference process of MU. Suppose we have a SimulMT policy based on boundary prediction to start a *write* action, and a binary boundary predictor gives the probability of a boundary at every step. In Step 1, the boundary predictor identifies a boundary for the given input of "I," and we obtain its corresponding translation "私は." In Steps 2 and 3, it does not predict a boundary; thus, we do not update the translation. Next, in Step 4, it finds a boundary

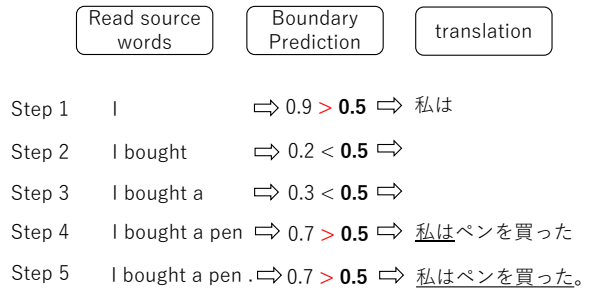


Figure 1: Example of the translation process of MU from English to Japanese. The threshold of boundary probability is 0.5 in this case. The underlined part is the output prefix used in forced decoding.

Source Prefix	Source prefix Translation	Full-sentence translation	boundary
I	私は。	私はペンを買った。	0
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。	1
I bought a pen .	私はペンを買った。	私はペンを買った。	1

Figure 2: Process of generating training dataset for MU-based boundary predictor. The underlined part is the prefix translation included in the full-sentence translation. The boundary label becomes 1 when such underlines exist.

after the observation of "I bought a pen," and we translate the input *with the output prefix constraint* "私は" and obtain "私はペンを買った." Finally, in Step 5, we translate the entire input *with the output prefix constraint* "私はペンを買った" and obtain "私はペンを買った。" We put underlines for the prefix constraints in Figure 1, which are used as the forced output prefixes in decoding. The inference processes of MU can be used with different boundary prediction methods using surface- or length-based heuristics (Kano et al. 2021), a statistical model, and so on. The translation process of our proposed method is also the same as MU, but we use the NMT model fine-tuned with bilingual prefix pairs.

3.3 MU-based boundary-predictor

Figure 2 shows the example process of generating the training dataset for an MU-based boundary predictor. First, the prefixes of source sentences in the training dataset are translated

Source Prefix	Source prefix Translation	Full-sentence translation	Extracted Target Prefix	boundary
I	私は。	私はペンを買いました。	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 3: Proposed method to extract the pairs of the source prefix and the corresponding target translation prefix. Underlines represent the longest common prefixes. We perform forced decoding with the extracted target prefix for both source prefix and full-sentence translation. Therefore, the full-sentence translation in the second row changes from that in the first row because the beam search result is different owing to forced decoding. The label of the boundary is 1 when the target prefix is extracted.

by a pre-trained full-sentence NMT model. If the entire source prefix translation is included in the full-sentence translation, it is regarded that it is a boundary of MU. In the fourth step of Figure 2, ”私はペンを買った” is the first source prefix translation included in the full-sentence translation, and the label of the boundary becomes 1. Once a boundary is detected, the prefix translation output is used for forced decoding in the next prefix translation. In Figure 2, the fourth prefix translation ”私はペンを買った” is used for forced-decoding when the fifth source prefix ”I bought a pen.” is translated.

After this process, pairs of a source prefix and boundary label such as (”I”, 0), (”I bought”, 0), ... are obtained. Thereafter, these pairs are used to train an end-to-end binary classifier. This classifier is used as described in Section 3.2

4 Proposed Method: Prefix Alignment

We fine-tune the pre-trained NMT model with bilingual prefix pairs to mitigate the gap between training and inference. To generate such prefix pairs, we propose a method called *prefix alignment*, which is shown in Figure 3. The procedure is as follows.

4.1 Generating Prefix Translation Pairs

First, we generate pairs of the source prefix and the corresponding prefix translation using the pre-trained full-sentence NMT model. Let $\boldsymbol{x}$ be a source language sentence and $\boldsymbol{y}$ be the

corresponding translation result in the target language obtained using the pre-trained NMT model.

Thereafter, we translate a prefix of $\boldsymbol{x}$ with one word[1], $\boldsymbol{x}_{|w|\leq 1}$, 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 $(\boldsymbol{x}_{|w|\leq 1}, \bar{\boldsymbol{y}}^{(1)}_{lcp})$ as a prefix translation pair. Taking the longest common prefix is motivated by the local agreement (Liu et al. 2020), which verifies the stability of a prefix translation based on the agreement between prefix translation results for growing input prefixes. We iterate this process by growing the input prefix individually; when we translate the m-word input prefix $\boldsymbol{x}_{|w|\leq m}$ into $\bar{\boldsymbol{y}}^{(m)}$, we find the longest common prefix $\bar{\boldsymbol{y}}^{(m)}_{lcp}$ between $\boldsymbol{y}$ and $\bar{\boldsymbol{y}}^{(m)}$. In the prefix translation, we constrain the decoding to follow the output prefixes found thus far for the same bilingual sentence pair by forced decoding, similar to chunk-based incremental decoding described in Section 3.2, to find consistent prefix translation pairs. The forced decoding avoids possible changes in the output prefix caused by a beam search. Furthermore, once we extract a prefix translation pair $(\boldsymbol{x}_{|w|\leq i}, \bar{\boldsymbol{y}}^{(i)}_{lcp})$, we update the full-sentence translation $\boldsymbol{y}$ by forced decoding using $\bar{\boldsymbol{y}}^{(i)}_{lcp}$ as an output prefix constraint and use the updated $\boldsymbol{y}$ for later steps $\bar{\boldsymbol{y}}^{(j)}$ $(j > i)$. If we obtain the same longest common prefix for different source language prefixes, we only take the first appearance (i.e., the pair from the smallest m) and ignore the remainder to avoid inconsistency among the prefix translation pairs. This prefix extraction strategy is different from that used for finding an MU (Zhang et al. 2020a), in which the entire prefix translation $\bar{\boldsymbol{y}}^{(i)}$ should be a prefix of the full-sentence translation $\boldsymbol{y}$ without taking the longest common prefix as in this study. Additionally, our strategy works flexibly by the possible update of the full-sentence translation by extracted output prefixes.

Figure 3 shows an example of the proposed method. The first prefix translation ends with a punctuation mark "。" Thus, MU (Zhang et al. 2020a) cannot extract the first prefix as the pair because the punctuation mark does not match the end of the prefix of full-sentence translation as shown in Figure 2. In contrast, the proposed method can extract the common output prefix by ignoring the latter part of the prefix translation. Therefore, the proposed method identifies more boundaries than the MU.

The extracted target prefixes are a part of source prefix translations. Therefore, we can fine-tune the NMT model with the extracted prefix pairs to prevent outputting unnecessary end of sentence expressions such as "。". In contrast, the main purpose of the MU is only to find the

[1] We use the word-based prefix length even though we use subwords for simplicity, and $\boldsymbol{x}_{|w|\leq 1}$ can comprise one or more subwords.

boundaries. The target prefixes with boundaries are already entirely included in the full-sentence translation; thus, they do not need to use the target prefixes for fine-tuning.

4.2 Prefix Alignment with References

The prefix translations obtained through the described process are NMT results and are different from their references ($\hat{\boldsymbol{y}}$) in general. We obtain reference-based prefix alignment by leveraging the extracted bilingual prefix pairs. We use BERTScore (Zhang et al. 2020b) to find the correspondence between an NMT-based target prefix and a reference prefix, varying the prefix window over the reference. We select the reference prefix $\hat{\boldsymbol{y}}^{(i)}$ with the largest BERTScore (in F-measure) against a given NMT-based target prefix $\bar{\boldsymbol{y}}_{lcp}^{(i)}$ and extract $(\boldsymbol{x}_{|w|\leq i},\ \hat{\boldsymbol{y}}^{(i)})$ as a reference-based bilingual prefix pair.

4.3 Fine-Tuning of a Simultaneous Machine Translation Model

We fine-tune the pre-trained NMT model for the SimulMT task using the extracted bilingual prefix pairs. The model works to translate an input prefix incrementally in the chunk-based decoding as presented in Section 3.2.

5 Experimental Setup

We conducted experiments on En-De and En-Ja simultaneous translation to compare the performance of the proposed method with the baselines in the quality-latency trade-off.

5.1 Compared Methods

The baseline is the NMT model trained with full-sentences (full-sentence NMT model), and we compared it with the NMT model fine-tuned with prefix pairs extracted by the proposed method. We used several boundary prediction methods for both models in the experiments. The compared boundary predictors are as follows.

- **Meaningful Unit (`MU`)**: We implemented the standard version of the MU boundary predictor (Zhang et al. 2020a) with the NMT model trained with full-sentences[2]. We

[2] We attempted to implement the refined version of MU (Zhang et al. 2020a). In the refined version, monotonic translations with pre-trained NMT were first generated using the training dataset. Thereafter, the NMT model was fine-tuned with the monotonic translations. However, in our experiment, the BLEU score of the NMT model fine-tuned with monotonic translations was extremely low because it was difficult to generate high-quality monotonic translations based on attention following a previous study. Therefore, we used the standard version in our experiments.

can adjust the SimulMT latency by the detection threshold for boundary probabilities given by the boundary predictor. We used 0.5 as its default value but also attempted 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 used three ranges because it is not clear how much the difference of hyperparameters changes the latency in the experiment.

- **Incremental Constituent Label Prediction (`ICLP`)**: Following our previous study (Kano et al. 2021), we used a one-look-ahead label predictor. We segmented the input sequence based on the rules with predicted labels `VP` and `S`. We varied the minimum segment length to adjust latency among $\{1, 2, 3, ..., 29\}$.
- **Fixed-length segmentation (`Fixed`)**: We also attempted simple fixed-length segmentation with a length hyperparameter L, implying that the boundary comes to every L words. We varied L among $\{2, 4, 6, ..., 30\}$.

The translation process is the same with all the boundary predictors as described in Section 3.2.

We additionally compared wait-k as another baseline that has a hyperparameter of latency provided at the training step different from the described three methods. We trained the wait-k model for each latency hyperparameter k among $\{2, 4, 6, ..., 30\}$.

5.2 Dataset and Preprocessing

	En-De	**En-Ja**
Pre-train	4.5M (WMT2014 train)	17.9M (WMT2020 train)
Fine-tune	206K (IWSLT2017 train)	223K (IWSLT2017 train)
Fine-tune by PA	206K (IWSLT2017 train)	223K (IWSLT2017 train)
Dev	5,589 (Mixed)	5,312 (Mixed)
Test	1,080 (IWSLT tst2015)	1,442 (IWSLT dev2021)

Table 1: Number of sentence pairs in the dataset. For the dev dataset, we used IWSLT dev2010, tst2010, tst2011 and tst2012 for En-De, and IWSLT dev2010, tst2011, tst2012, and tst2013 for En-Ja. PA is short for prefix alignment.

Table 1 lists the dataset that we used in the experiment. First all the translation models were trained with pre-training dataset. Next we fine-tuned these models with the IWSLT training dataset. Simultaneous translation is basically used for spoken language. The IWSLT training dataset is a spoken language corpus, but it is relatively small. Therefore, we first pre-trained

the NMT model with a large training dataset from WMT, and subsequently fine-tuned it with the IWSLT training dataset. This NMT model is the baseline. For the proposed method, we generated bilingual prefix pairs only from the fine-tuning dataset, because extracting prefix pairs from a large training corpus requires a significant amount of time. Thereafter, we fine-tuned the baseline NMT model with the prefix pairs generated after the process of prefix alignment with references explained in Section 4.2.

We tokenized Japanese sentences using MeCab (Kudo 2005) and English and German sentences using `tokenizer.perl` in Moses (Koehn et al. 2007), followed by the subword tokenization based on Byte Pair Encoding (BPE) (Sennrich et al. 2016) with a shared subword vocabulary of 16K entries.

5.3 NMT Models

All the NMT models were based on the transformer-base (Vaswani et al. 2017) implementation in fairseq (Ott et al. 2019). The 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 patience for four checkpoints, based on the loss on the development set. We set the learning rate to 0.0007, and the minibatch size to 4,096 with the parameter update frequency of 4. We applied a chunk-based beam search described in Section 3.2 to the methods except wait-k. We used a greedy decoding for wait-k, because of the nature of its model.

5.4 Evaluation Metrics

We used BLEU (Papineni et al. 2002) to evaluate translation quality and evaluated the latency of simultaneous translation with Average Lagging (AL) (Ma et al. 2019) and Average Token Delay (ATD) (Kano et al. 2023).

5.4.1 Average Lagging (AL)

Ma et al. (2019) proposed Average Lagging (AL). Suppose we have an input $\boldsymbol{x} = x_1, ..., x_{|\boldsymbol{x}|}$ and an output $\boldsymbol{y} = y_1, ..., y_{|\boldsymbol{y}|}$. AL is denoted as follows:

$$AL_g(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{\tau_g(|\boldsymbol{x}|)} \sum_{t=1}^{\tau_g(|\boldsymbol{x}|)} \left(g(t) - \frac{t-1}{r} \right), \tag{3}$$

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

where r denotes the length ratio defined as $|\boldsymbol{y}|/|\boldsymbol{x}|$ and $g(t)$ represents the number of source tokens already read to output the t th target token. $\tau_g(|\boldsymbol{x}|)$ denotes the cut-off step defined as follows:

$$\tau_g(|\boldsymbol{x}|) = \min\{t \mid g(t) = |\boldsymbol{x}|\}, \tag{4}$$

implying the index of the output token predicted right after the observation of the entire source sentence.

AL is the most commonly used latency metric that focuses on when to start the translation. However, AL has the problem that the longer the segment translation becomes, the smaller the latency becomes although long segment translation should delay the latter translation in actual situations. Therefore, AL is compatible with wait-k that outputs translation tokens individually, but not with chunk-based models that output multiple tokens instantly.

5.4.2 Average Token Delay (ATD)

ATD is a metric that adequately considers when to finish the translation. It includes the delay at the end of segment translation for that of simultaneous translation. Therefore, we also used ATD to appropriately evaluate chunk-based translation models.

Suppose $\boldsymbol{x}$ and $\boldsymbol{y}$ are segmented into C chunks $\boldsymbol{x} = \boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^C$ and $\boldsymbol{y} = \boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^C$, respectively. An input chunk $\boldsymbol{x}^c$ is observed after the previous prefix translation $\boldsymbol{y}^{c-1}$ and used to predict $\boldsymbol{y}^c$.

ATD is defined as follows:

$$\mathrm{ATD}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{|\boldsymbol{y}|} \sum_{t=1}^{|\boldsymbol{y}|} \big(T(y_t) - T(x_{a(t)})\big) \tag{5}$$

where

$$a(t) = \begin{cases} s(t) & \text{if } s(t) \leq L_{acc}(\boldsymbol{x}^{c(t)}) \\ L_{acc}(\boldsymbol{x}^{c(t)}) & \text{otherwise} \end{cases} \tag{6}$$

$$s(t) = t - \max(L_{acc}(\boldsymbol{y}^{c(t)-1}) - L_{acc}(\boldsymbol{x}^{c(t)-1}), 0) \tag{7}$$

$T(\cdot)$ in Eq. (5) denotes the ending time of each token and $a(t)$ denotes the index of the input token corresponding to y_t. $L_{acc}(\boldsymbol{x}^c) = \sum_{j=1}^{c} |\boldsymbol{x}^j|$ is the cumulative length up to the c-th chunk, and $L_{acc}(\boldsymbol{x}^0) = 0$. $L_{acc}(\boldsymbol{y}^c)$ is defined similarly. $c(t)$ denotes the chunk number c to which y_t

belongs.

AL and ATD were calculated using SimulEval (Ma et al. 2020a) and drawn in scatterplots to show the quality-latency trade-off.

6 Result

Figures 4a to 4d compare the baselines with the PA (prefix alignment)-based methods. Here, `full` implies that the NMT model was trained only with full sentences. `PA` implies that the NMT model was fine-tuned with prefix pairs of the source and reference in the fine-tuning dataset.

The AL results in En-De shown in Figure 4a indicate that the PA-based models demonstrated higher performance than the others in all latency ranges. In contrast, those in En-Ja exhibited the advantage only in an extremely low latency range (AL < 3), as shown in Figure 4b.

The ATD results in Figure 4c show that PA-based models also worked well in most latency ranges for En-De, except for wait-k in the lowest latency range ($k = 2, 4$). ATD is a latency metric considering the delay caused by long translation outputs in previous steps; thus, so this result reflects a possible advantage of wait-k in ATD; Its alternating actions of reading and writing one token at a step are advantageous in the ATD measurement. In En-Ja, the PA-based model worked well in the low-middle latency ranges in Figure 4d. The gap between PA-based models and full-sentence-based models, particularly in low latency is larger in ATD than in AL.

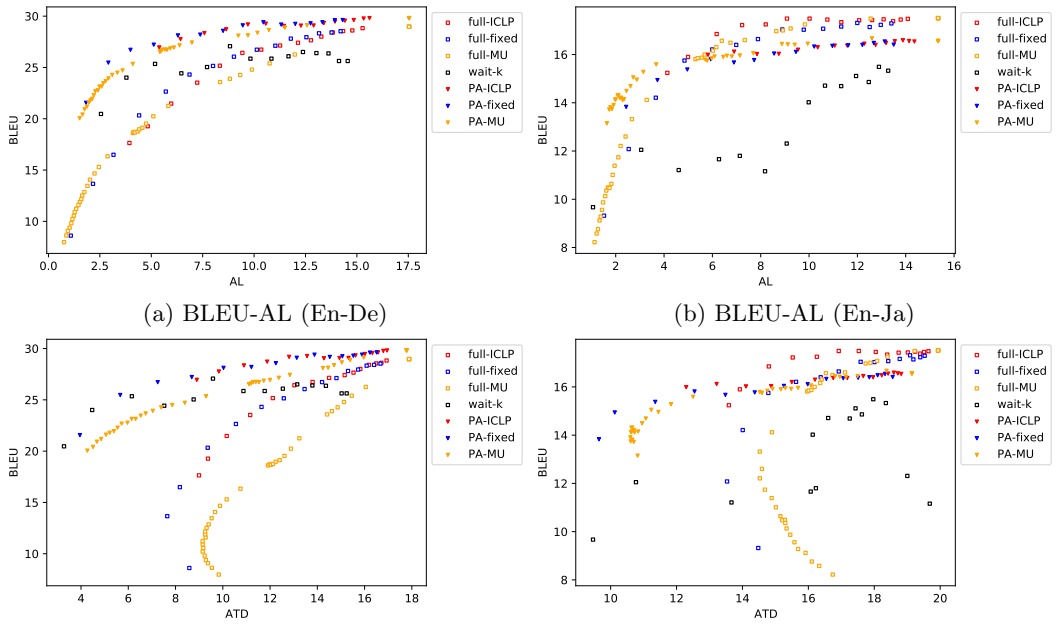
7 Analyses

We performed the following analyses to investigate the characteristics of the proposed method in detail.

7.1 Length-Related Issues

7.1.1 Length Ratio

First, we analyzed the length ratios of the translation outputs and references to investigate over-translation phenomena in prefix-to-prefix translation. Figures 5a and 5b show the scatterplots of AL and the length ratio for En-De and En-Ja, respectively. Fine-tuning using prefix alignment pairs decreased the length ratio toward 1.0. This implies that the translation outputs had fewer unnecessary words, resulting in higher BLEU by increasing n-gram precisions. From the perspective of the latency in ATD, this length ratio reduction derives better ATD because it considers delays caused by the outputs. In contrast, the `full-MU` exhibited extremely large



(a) BLEU-AL (En-De)

(b) BLEU-AL (En-Ja)

(c) BLEU-ATD (En-De)

(d) BLEU-ATD (En-Ja)

Figure 4: Each point represents each value of latency hyperparameter. For instance, in Figure 4a, the leftmost black dot of wait-k represents k = 2. The x-axis represents delay and the delay is larger on the right side. The y-axis represents quality and the quality is better on the upper side. Therefore, the dot near the upper left corner is better in quality-latency trade-off.

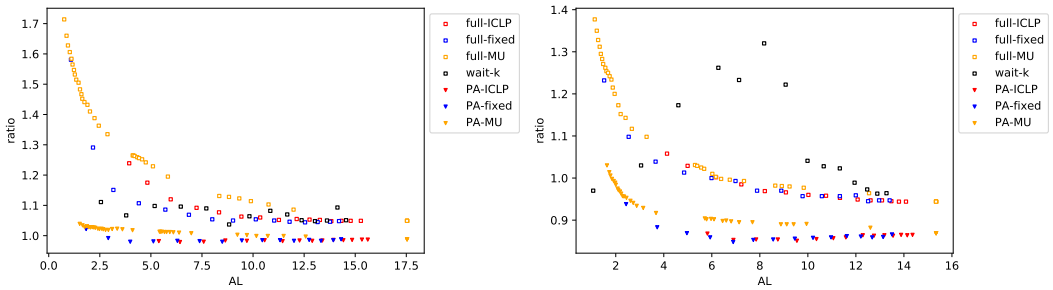
length ratios (> 1.2) with their hyperparameter settings for smaller AL. Such large length ratios increased the ATD as shown in the bottom part of Figures 4c and 4d.

	En-De	**En-Ja**
Reference sentence / Source sentence	1.03	1.06
Translation prefix / Source prefix	1.05	**0.87**

Table 2: Length ratio of source to target

7.1.2 Degradation in high latency for En-Ja

The degradation of PA-based models in high latency for En-Ja can be explained by word order difference and the quality of the full-sentence NMT model. From Figures 5a and 5b, we



(a) En-De (b) En-Ja

Figure 5: Length ratio and Average Lagging

Source Prefix	Source prefix Translation	Extracted Target Prefix
I	Ich.	Ich
I like	Ich mag.	
I like playing	Ich spiele gerne.	Ich spiele gerne
I like playing soccer	Ich spiele gerne Fußball.	Ich spiele gerne Fußball
I like playing soccer in	Ich spiele gerne Fußball im	Ich spiele gerne Fußball im
I like playing soccer in the	Ich spiele gerne Fußball im	
I like playing soccer in the park	Ich spiele gerne Fußball im Park	Ich spiele gerne Fußball im Park
I like playing soccer in the park.	Ich spiele gerne Fußball im Park.	Ich spiele gerne Fußball im Park.

Figure 6: Example of Prefix Alignment (En-De)

	En-De	En-Ja
F1	0.87	0.80

Table 3: BERTScore of target prefix and reference prefix

find that the length ratios by the PA-based methods were smaller than those by the full-sentence translation, and they became less than 1.0 in En-Ja. We investigated the difference between the language pairs on the training data. Table 2 lists the length ratios of the source to target in the IWSLT training dataset for the original sentence pairs and the prefix pairs extracted by Prefix Alignment. We found that the length ratio of the prefix pairs of En-Ja was much smaller than En-De, while it was almost the same in the sentence level. This is probably because of the larger word order difference between English and Japanese. Figures 6 and 7 show examples of the Prefix Alignment for an English sentence "I like playing soccer in the park," with a German

Source Prefix	Source prefix Translation	Extracted Target Prefix
I	私は。	私は
I like	私は好き。	
I like playing	私は遊ぶのが好き。	
I like playing soccer	私はサッカーをするのが好き。	私はサッカーを
I like playing soccer in	私はサッカーをするのが好き。	
I like playing soccer in the	私はサッカーをするのが好き。	
I like playing soccer in the park	私はサッカーを公園でするのが好き	私はサッカーを公園でするのが好き
I like playing soccer in the park.	私はサッカーを公園でするのが好き。	私はサッカーを公園でするのが好き。

Figure 7: Example of Prefix Alignment (En-Ja)

translation “Ich spiele gerne Fußball im Park,” and a Japanese translation “私はサッカーを公園でするのが好き。”. For an English prefix “I like playing soccer”, we extract a semantically equivalent German prefix “Ich spiele gerne Fußball” but obtain a Japanese prefix “私はサッカーを” that misses the counterparts of “like (好き)” and “playing (する)”. This is because Prefix Alignment does not guarantee the semantic equivalence of an extracted prefix pair. Our prefix pair extraction described in Figure 3 finds the longest common prefix between a prefix and full-sentence translations. The remaining part of the prefix translation is discarded and is missing in the extracted prefix pair, and this causes the aforementioned semantic inequivalence. For a given source language prefix, Prefix Alignment gives possible translation prefixes that *can be determined* at the time of the observation of the source language prefix and does not guarantee these prefixes cover the contents of the entire source language prefix. Such an issue happens mainly because of word order differences because some parts in a source language prefix can be translated only after the translation of later parts that are not observed thus far. For instance, in the translation from a Subject-Verb-Object language (English) to a Subject-Object-Verb language (Japanese), a verb should be translated after the translation of its object. Consequently, the PA-based models derived short translations as shown in Figure 5b and resulted in worse BLEU for En-Ja.

From the perspective of the quality of prefix pairs, Table 3 lists the similarity between the hypothesis- and reference-based prefix measured by BERTScore (Zhang et al. 2020b). We observed a lower score in En-Ja than in En-De. This is because of the difference in the translation quality by the full-sentence NMT model as shown in Figures 4b and 4a.

Therefore, the degradation of PA-based models in high latency for En-Ja is caused by the word omission in the extracted target prefix and low quality of extracted prefix pairs.

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

Table 4: Statistics of the training data

7.1.3 Length Distribution in the Training Set

Table 4 summarizes statistics from the IWSLT training set in the number of extracted prefix pairs and words in the entire dataset. Even though the number of words was almost similar, that of prefix pairs in En-De was almost two times larger than that in En-Ja. This is also due to the large word order difference discussed above because we have to wait for the later source language input to *grow* a prefix translation in the target language. Figure 8 shows the length distributions of the source language portions of the prefix pairs. The numbers of short source language prefixes in En-Ja were much smaller than those in En-De. Figures 9 and 10 show the length distributions from the entire training data with and without Prefix Alignment. Prefix alignment augmented much more prefix translation pairs for English-to-German than for English-to-Japanese. These findings suggest that many short prefix pairs should contribute to improving SimulMT in En-De.

7.2 Comparison of Boundary Predictors

Second, we investigated the effects of boundary prediction on the performance of the PA-based models by comparing different boundary prediction methods.

7.2.1 Boundary Predictors Based on Prefix Pairs

In addition to the existing segmentation methods, we tested a boundary predictor that is suitable to the NMT model trained with prefix pairs. The predictor identifies a prefix boundary to start a *write* action. The training data for the boundary prediction model comprise pairs of a source language sentence prefix and the corresponding boundary label. The label is set to 1 for a prefix included in the extracted prefix pairs and 0 otherwise, as shown in the rightmost column in Figure 3. The boundary prediction can also be extended to use a target language prefix as an additional clue. The extended model takes a pair of input and output prefixes at the time of the boundary prediction and can be trained similarly. For instance, at the fourth step in Figure 3, it should identify a boundary with an input-output prefix pair of (“I bought a pen”, “私は”) in the form of a concatenates string “I bought a pen ||| 私は”.

We trained the following boundary prediction models for comparison:

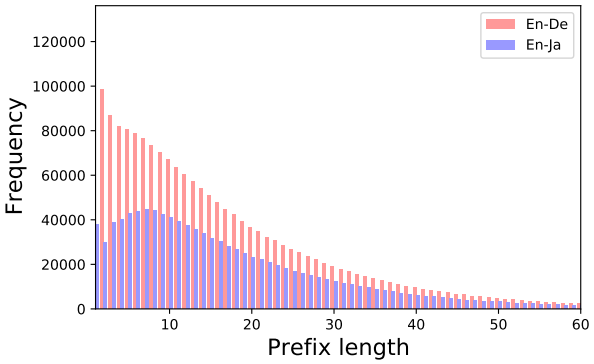


Figure 8: Source prefix length distribution in the IWSLT training data

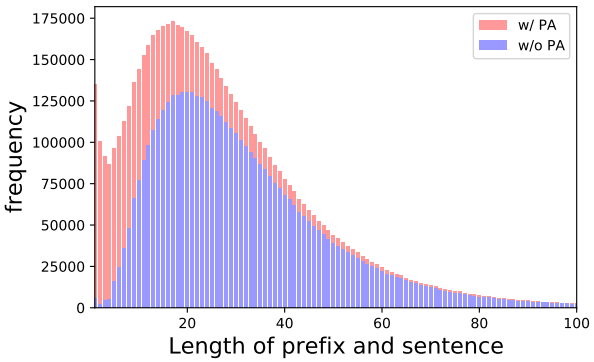


Figure 9: Source sentence length distribution in the training data (En-De)

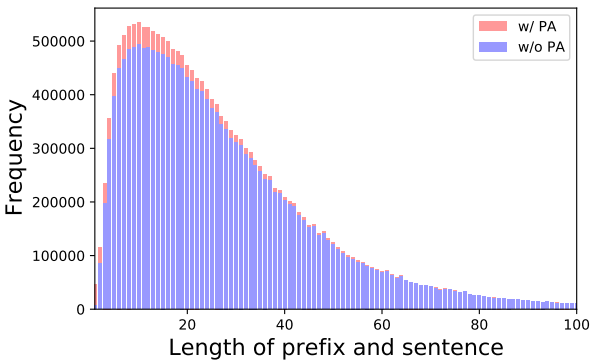


Figure 10: Source sentence length distribution in the training data (En-Ja)

Language Pair	Model	Precision	Recall	F1
En-De	`bert-src-w`	0.58	0.70	0.64
	`ro-src-w`	0.63	0.76	0.69
	`ro-src-tgt`	**0.78**	0.75	**0.77**
	`ro-src-tgt-w`	0.74	**0.81**	**0.77**
En-Ja	`bert-src-w`	0.43	0.69	0.53
	`ro-src-w`	0.43	0.77	0.56
	`ro-src-tgt`	**0.73**	0.51	0.60
	`ro-src-tgt-w`	0.48	**0.81**	**0.61**

Table 5: Result of boundary prediction

- `bert-src-w`
- `ro-src-w`
- `ro-src-tgt-w`
- `ro-src-tgt`

where `bert` means the use of a pre-trained English $\text{BERT}_{\text{base}}$ model [4] (Devlin et al. 2019), and `ro` means the use of a pretrained XLM-$\text{RoBERTa}_{\text{base}}$ model [5] (Conneau et al. 2020). `src` and `tgt` mean the use of the current prefix in the source and target languages, respectively. `w` means the use of a weighted loss to mitigate the label imbalance for training boundary prediction models.

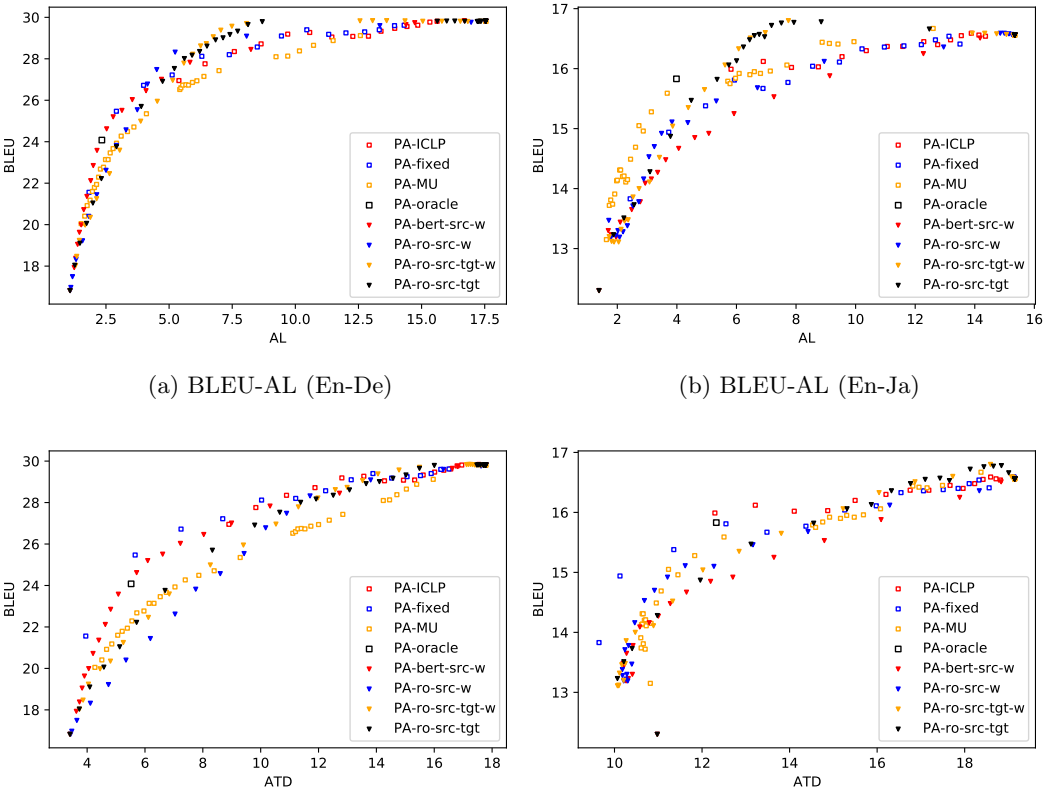
We also compared oracle segmentation to investigate the limitation of the PA-based boundary prediction. To obtain oracle segmentation boundaries, we applied Prefix Alignment to the test set and the end of source prefixes with label 1 are the oracle boundaries.

7.2.2 Boundary Prediction Performance

We evaluated their performance using the development set listed in Table 1 against the boundaries obtained through Prefix Alignment.

Table 5 summarizes the results. For both language pairs, `ro-src-tgt-w` demonstrated the highest F1 scores, and those of `ro-src-w` were slightly better than those of `bert-src-w`. `ro-src-tgt` resulted in a higher precision and lower recall, particularly in En-Ja, but the F1 scores were almost the same as those of `ro-src-tgt-w`.

[4] https://huggingface.co/bert-base-uncased
[5] https://github.com/facebookresearch/fairseq/tree/main/examples/xlmr



(a) BLEU-AL (En-De)

(b) BLEU-AL (En-Ja)

(c) BLEU-ATD (En-De)

(d) BLEU-ATD (En-Ja)

Figure 11: Models fine-tuned using the proposed method with different boundary predictors

7.3 Comparison of PA-based Models

Figures 11a to 11d show the quality-latency results of PA-based models in AL and ATD on En-De and En-Ja SimulMT experiments. For PA-based boundary predictors, we used the same thresholds as MU to adjust the latency. The AL results in Figures 11a and 11b show that `PA-ro-src-tgt-w` and `PA-ro-src-tgt` were the best in the middle and high latency ranges for both En-Ja and En-De. In low latency ranges, `PA-bert-src-w` and `PA-MU` are the best for En-De and En-Ja respectively, and their curves are near the oracle segmentation. Surprisingly, `PA-fixed` was comparably effective in the low latency range.

The ATD results in Figures 11c and 11d show that `PA-ro-src-tgt-w` and `PA-ro-src-tgt` were slightly better than the other models in the high latency range. In the low latency range, `PA-fixed` worked the best. The fixed-length heuristics ran much faster than the other segmenta-

Latency	# Null	# Total
2	822	9752
6	24	3607
10	1	2364
14	1	1841
18	0	1555
22	0	1398
26	0	1290
30	0	1229

(a) En-De

Latency	# Null	# Total
2	2978	11393
6	244	4297
10	49	2852
14	16	2282
18	3	1953
22	0	1765
26	0	1638
30	0	1567

(b) En-Ja

Table 6: Total number of input prefixes and frequencies of those that resulted in null outputs by `PA-fixed` in the test set translation

tion models requiring more computations; thus, we may prefer it in practice. Although `ICLP` has a limitation on the number of segmentation because of the number of verbs, the first few points in low latency have achieved the best performance in ATD for En-Ja. When we observe oracle segmentation, it is not the best in ATD.

In conclusion, the curves by `PA-ro-src-tgt` are very similar to those by `PA-ro-src-tgt-w` despite the difference in the precision-recall trade-offs shown in Table 5. In the high latency range, `PA-ro-src-tgt` and `PA-ro-src-tgt-w` were the best with high thresholds because they are compatible with the NMT model fine-tuned with PA and use the richer input feature for prediction compared to other PA-based boundary predictors. These models also have the highest precision and recall as listed in Table 5. In the low latency range, mixed results were obtained and the PA-based boundary predictors were not always the best for low thresholds.

7.4 Analysis of Fixed-size Segmentation

Fixed-size segmentation (`PA-fixed`) worked comparably or sometimes better than model-based boundary predictions. Tables 6a and 6b list the numbers of input prefixes that yield *null outputs*, where the end-of-the-sentence (EOS) symbol was predicted at the beginning, on the test at different latency (in the number of input tokens) settings of `PA-fixed`. As summarized in these tables, the prefix-to-prefix translation often generated null outputs in the low latency settings. The null outputs do not impose any constraints on the following prefix translations; thus, `PA-fixed` worked flexibly to wait for later inputs. The tables also reveeal that it happened more frequently in En-Ja than in En-De. This indicates that En-Ja translation requires a longer input prefix than En-De because of the larger word order difference.

8 Conclusion

We proposed a method to train the neural SimulMT model by extracting bilingual prefix pairs through prefix alignment. The proposed method outperformed the baselines in a quality-latency trade-off in En-De SimulMT, but the results in En-Ja were mixed. We investigated the results in detail and found the performance degradation in En-Ja came from poor prefix pair extraction mainly because of the word order difference. The proposed method can be extended to the speech-to-text translation by extracting a pair of a source language speech and target language text fragments.[6]

In future work, we will tackle En-Ja problems and improve the method to work for more language pairs.

Acknowledgement

This paper is an extended version of our conference paper (Kano et al. 2022) with additional experiments and analyses. Part of this work was supported by JSPS KAKENHI with Grant Numbers JP21H05054 and JP21H03500.

References

Alinejad, A., Siahbani, M., and Sarkar, A. (2018). "Prediction Improves Simultaneous Neural Machine Translation." In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3022–3027, Brussels, Belgium. Association for Computational Linguistics.

Anastasopoulos, A., Bentivogli, L., Boito, M. Z., Bojar, O., Cattoni, R., Currey, A., Dinu, G., Duh, K., Elbayad, M., Federico, M., Federmann, C., Gong, H., Grundkiewicz, R., Haddow, B., Hsu, B., Javorský, D., Kloudová, V., Lakew, S. M., Ma, X., Mathur, P., McNamee, P., Murray, K., Nădejde, M., Nakamura, S., Negri, M., Niehues, J., Niu, X., Pino, J., Salesky, E., Shi, J., Stüker, S., Sudoh, K., Turchi, M., Virkar, Y., Waibel, A., Wang, C., and Watanabe, S. (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.

[6] We applied this method to our speech-to-text simultaneous machine translation system for the IWSLT 2022 Evaluation Campaign (Anastasopoulos et al. 2022; Fukuda et al. 2022).

Arivazhagan, N., Cherry, C., Macherey, W., Chiu, C.-C., Yavuz, S., Pang, R., Li, W., and Raffel, C. (2019). "Monotonic Infinite Lookback Attention for Simultaneous Machine Translation." In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1313–1323, Florence, Italy. Association for Computational Linguistics.

Bangalore, S., Rangarajan Sridhar, V. K., Kolan, P., Golipour, L., and Jimenez, A. (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*, pp. 437–445, Montréal, Canada. Association for Computational Linguistics.

Chiu, C.-C. and Raffel, C. (2018). "Monotonic Chunkwise Attention." In *International Conference on Learning Representations*.

Cho, K. and Esipova, M. (2016). "Can neural machine translation do simultaneous translation?" *arXiv preprint arXiv:1606.02012*.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). "Unsupervised Cross-lingual Representation Learning at Scale." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Online. Association for Computational Linguistics.

Dalvi, F., Durrani, N., Sajjad, H., and Vogel, S. (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)*, pp. 493–499, New Orleans, Louisiana. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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)*, pp. 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fujita, T., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. (2013). "Simple, Lexicalized Choice of Translation Timing for Simultaneous Speech Translation." In *Proc. Interspeech 2013*, pp. 3487–3491.

Fukuda, R., Ko, Y., Kano, Y., Doi, K., Tokuyama, H., Sakti, S., Sudoh, K., and Nakamura, S. (2022). "NAIST Simultaneous Speech-to-Text Translation System for IWSLT 2022." In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT*

2022), pp. 286–292, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Grissom II, A., He, H., Boyd-Graber, J., Morgan, J., and Daumé III, H. (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)*, pp. 1342–1352, Doha, Qatar. Association for Computational Linguistics.

Gu, J., Neubig, G., Cho, K., and Li, V. O. (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*, pp. 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Kano, Y., Sudoh, K., and Nakamura, S. (2021). "Simultaneous Neural Machine Translation with Constituent Label Prediction." In *Proceedings of the Sixth Conference on Machine Translation*, pp. 1124–1134, Online. Association for Computational Linguistics.

Kano, Y., Sudoh, K., and Nakamura, S. (2022). "Simultaneous Neural Machine Translation with Prefix Alignment." In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pp. 22–31, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Kano, Y., Sudoh, K., and Nakamura, S. (2023). "Average Token Delay: A Latency Metric for Simultaneous Translation.".

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (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*, pp. 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Kudo, T. (2005). "Mecab : Yet another part-of-speech and morphological analyzer." http://mecab.sourceforge.net/.

Liu, D., Spanakis, G., and Niehues, J. (2020). "Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection." In *Proc. Interspeech 2020*, pp. 3620–3624.

Ma, M., Huang, L., Xiong, H., Zheng, R., Liu, K., Zheng, B., Zhang, C., He, Z., Liu, H., Li, X., Wu, H., and Wang, H. (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*, pp. 3025–3036,

Florence, Italy. Association for Computational Linguistics.

Ma, X., Dousti, M. J., Wang, C., Gu, J., and Pino, J. (2020a). "SIMULEVAL: An Evaluation Toolkit for Simultaneous Translation." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 144–150, Online. Association for Computational Linguistics.

Ma, X., Pino, J. M., Cross, J., Puzon, L., and Gu, J. (2020b). "Monotonic Multihead Attention." In *International Conference on Learning Representations*.

Oda, Y., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. (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)*, pp. 551–556, Baltimore, Maryland. Association for Computational Linguistics.

Oda, Y., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. (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)*, pp. 198–207, Beijing, China. Association for Computational Linguistics.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (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)*, pp. 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). "Bleu: a Method for Automatic Evaluation of Machine Translation." In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Raffel, C., Luong, M.-T., Liu, P. J., Weiss, R. J., and Eck, D. (2017). "Online and linear-time attention by enforcing monotonic alignments." In *International conference on machine learning*, pp. 2837–2846. PMLR.

Rangarajan Sridhar, V. K., Chen, J., Bangalore, S., Ljolje, A., and Chengalvarayan, R. (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 Language Technologies*, pp. 230–238, Atlanta, Georgia. Association for Computational Linguistics.

Sankaran, B., Grewal, A., and Sarkar, A. (2010). "Incremental Decoding for Phrase-Based

Statistical Machine Translation." In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pp. 216–223, Uppsala, Sweden. Association for Computational Linguistics.

Satija, H. and Pineau, J. (2016). "Simultaneous machine translation using deep reinforcement learning." In *Workshops of International Conference on Machine Learning*, p. 110–119.

Sennrich, R., Haddow, B., and Birch, A. (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)*, pp. 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). "Attention is All you Need." In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.

Zhang, R., Zhang, C., He, Z., Wu, H., and Wang, H. (2020a). "Learning Adaptive Segmentation Policy for Simultaneous Translation." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2280–2289, Online. Association for Computational Linguistics.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020b). "BERTScore: Evaluating Text Generation with BERT." In *International Conference on Learning Representations*.

Zheng, B., Liu, K., Zheng, R., Ma, M., Liu, H., and Huang, L. (2020). "Simultaneous Translation Policies: From Fixed to Adaptive." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2847–2853, Online. Association for Computational Linguistics.

Zheng, B., Zheng, R., Ma, M., and Huang, L. (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)*, pp. 1349–1354, Hong Kong, China. Association for Computational Linguistics.

Yasumasa Kano：He is a Ph.D. student at Nara Institute of Science and Technology. He received his bachelor's degree in economics from Yokohama National University in 2019 and his master's degree in engineering from Nara Institute of Science and Technology in 2021.

Katsuhito Sudoh : He is an associate professor at Nara Institute of Science and Technology. He received his bachelor's degree in engineering in 2000 and his master's and Ph.D. degrees in informatics in 2002 and 2015, respectively, from Kyoto University. He worked with the NTT Communication Science Laboratories from 2002 to 2017. He currently works on machine translation and natural language processing. He is a member of ACL, ISCA, ANLP, IPSJ, ASJ, and JSAI.

Satoshi Nakamura : He is a professor at Nara Institute of Science and Technology and an honorary professor at the Karlsruhe Institute of Technology, Germany. He received his B.S. from the Kyoto Institute of Technology in 1981 and a Ph.D. from the Kyoto University in 1992. He was the Director of ATR Spoken Language Communication Research Laboratories in the period 2000–2008 and Vice President of ATR in the period 2007–2008. He was the Director General of Keihanna Research Laboratories and Executive Director of Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology in 2009–2010. He is currently the director of the Augmented Human Communication laboratory and a full professor at the Data Science Center and Information Science Division, Graduate School of Science and Technology, Nara Institute of Science and Technology. He was an Elected Board Member of the International Speech Communication Association, ISCA, in the period June 2011-2019, and IEEE Signal Processing Magazine Editorial Board member in the period 2012-2015, and IEEE SPS Speech and Language Technical Committee Member in the period 2013-2015. He is an ATR Fellow, IPSJ Fellow, IEEE Fellow, and ISCA Fellow.