NAIST Simultaneous Speech Translation System for IWSLT 2024

Yuka Ko   Ryo Fukuda   Yuta Nishikawa   Yasumasa Kano   Tomoya Yanagita
Kosuke Doi   Mana Makinae   Haotian Tan   Makoto Sakai
Sakriani Sakti   Katsuhito Sudoh   Satoshi Nakamura
Nara Institute of Science and Technology, Japan
`ko.yuka.kp2@is.naist.jp`

Abstract

This paper describes NAIST's submission to the simultaneous track of the IWSLT 2024 Evaluation Campaign: English-to-{German, Japanese, Chinese} speech-to-text translation and English-to-Japanese speech-to-speech translation. We develop a multilingual end-to-end speech-to-text translation model combining two pretrained language models, HuBERT and mBART. We trained this model with two decoding policies, Local Agreement (LA) and AlignAtt. The submitted models employ the LA policy because it outperformed the AlignAtt policy in previous models. Our speech-to-speech translation method is a cascade of the above speech-to-text model and an incremental text-to-speech (TTS) module that incorporates a phoneme estimation model, a parallel acoustic model, and a parallel WaveGAN vocoder. We improved our incremental TTS by applying the Transformer architecture with the AlignAtt policy for the estimation model. The results show that our upgraded TTS module contributed to improving the system performance.

1   Introduction

This paper presents NAIST's simultaneous speech translation (SimulST) systems for the English-to-{German, Japanese, Chinese} speech-to-text track and the English-to-Japanese speech-to-speech track within the simultaneous track of the IWSLT 2024 Evaluation Campaign.

Simultaneous translation involves generating translations incrementally based on partial input, and it requires interpreters who can provide accurate and fluent translations while minimizing delay.

Early SimulST systems are based on a cascade of automatic speech recognition (ASR) and machine translation modules (*e.g.,* Fügen et al., 2007; Bangalore et al., 2012; Yarmohammadi et al., 2013; Oda et al., 2014; Arivazhagan et al., 2020), but they suffer from error propagation and added latency imposed by the ASR module. Recently, an end-to-end approach has become popular (Agarwal et al., 2023), and this approach has been demonstrated to achieve a better quality-latency trade-off.

Conventional end-to-end SimulST models have employed training strategies and architectures designed for a simultaneous setting. However, that approach not only requires additional effort in system development but also results in high computational costs. To alleviate such problems, Papi et al. (2022a) proposed a single model trained on offline translation data for the simultaneous scenario. Applying a simultaneous decoding policy to an offline speech translation (ST) model in SimulST inference enables the model to generate outputs similar to simultaneous translation. Furthermore, a decoding policy determines whether to generate partial output or wait for more input.

Using an offline ST model with a simultaneous decoding policy has become popular because no specific task adaptation is required for a SimulST task. Among several simultaneous decoding policies (Cho and Esipova, 2016; Dalvi et al., 2018; Ma et al., 2019, 2020b; Nguyen et al., 2021), Local Agreement (LA) (Liu et al., 2020) is widely used and won the SimulST task at the IWSLT 2022 Evaluation Campaign (Anastasopoulos et al., 2022). The LA policy extracts the longest common prefixes from the n consecutive chunks as stable hypotheses. However, it requires a long computation time to obtain the longest common prefix.

Since simultaneous translation requires *real-time* translation, a policy that runs fast is desirable. Papi et al. (2023) proposed a decoding policy called AlignAtt, which takes the alignments of the source and target tokens using cross attention information. Under computation-aware settings, Papi et al. (2023) have shown that AlignAtt can generate translations with lower latency compared to the LA policy, and it is capable of reaching a latency of 2 sec or less.

For the IWSLT 2024 Evaluation Campaign, we

218

developed two types of speech-to-text translation models with different decoding policies and compared them. One is based on LA and the other on AlignAtt. The LA-based model demonstrates better quality than the AlignAtt-based one within the given latency constraints, while the AlignAtt policy works better in a low-latency region in computation-aware settings.

For the English-to-Japanese speech-to-speech track, we developed a cascade of the above SimulST model and an incremental text-to-speech module using a phoneme and prosodic symbol estimation model, a parallel acoustic model, and a parallel WaveGAN vocoder. In last year's submission, our speech-to-speech translation method suffered from the quality of the synthesized speech and possible ASR errors (Fukuda et al., 2023). The authors reported that the character error rate of the NAIST 2023 speech-to-speech translation output exceeded that of the SimulST text output by over 28%. Therefore, we upgraded our TTS module by incorporating Transformer architecture and AlignAtt in the estimation model.

2 System Architecture

This section describes the architecture of our SimulST systems. First, we explain the decoding policies used for our translation modules. Then, we present the details of our simultaneous speech-to-text and speech-to-speech translation methods.

2.1 Decoding Policies

2.1.1 Local Agreement

Liu et al. (2020) introduced the concept of Local Agreement to find a stable prefix translation hypothesis in simultaneous translation scenarios where inputs are processed in fixed-length chunks. This method assesses the stability of a hypothesis at step t by comparing it with the hypothesis at step $t-1$, thus determining the agreeing prefix (i.e., the longest common prefix) between them. The underlying principle is that the translation outputs with consistent agreeing prefixes, as the input prefixes increase, are likely to be reliable. Building upon this idea, Polák et al. (2022) extended it to encompass agreement among prefixes over n consecutive steps (LA-n), with their experiments showing that $n = 2$ performs effectively in the context of SimulST. Based on these findings, we employed LA-2 as a SimulST policy and adjusted the input chunk length (in milliseconds) to manage the trade-off between quality and latency.

2.1.2 AlignAtt

Papi et al. (Papi et al., 2023) proposed AlignAtt, a method that leverages encoder-decoder attention information in Transformer to establish alignment between source and target tokens during inference. According to the AlignAtt policy, if a target token aligns with tokens beyond the last f tokens of the source speech, it implies that adequate information has been provided to generate that token. Consequently, if a target token aligns solely with the last f tokens from the source, generation is paused to await additional speech input. In our implementation, we use cross attention from the decoder to the length adapter for AlignAtt.

2.2 Simultaneous Speech-to-Text Translation

Our speech-to-text SimulST system uses multilingual offline speech translation models for the prefix-to-prefix translation required for SimulST. These models are based on large-scale pre-trained speech and text models adopting Hidden-Unit BERT (HuBERT) (Hsu et al., 2021) and mBART50 (Tang et al., 2020), following Polák et al. (2022). We initialized our ST models with the HuBERT speech encoder and the mBART50 text decoder, which were fine-tuned using English ASR data and multilingual MT data, respectively. In addition, we applied Inter-connection (Nishikawa and Nakamura, 2023) for the concatenated ST model. Inter-connection is a method that aggregates the information from each layer of a pre-trained speech model with weighted sums and then passes it into the decoder by connecting the intermediate layer of the speech encoder and the text decoder. We also fine-tuned the multilingual ST model using bilingual prefix pairs in English-to-{German, Japanese, Chinese} extracted using *Bilingual Prefix Alignment* (Kano et al., 2022). Bilingual Prefix Alignment is a method used to generate augmented prefix-to-prefix data based on a pre-trained offline model, and the SimulST model fine-tuned on those data will generate high quality output in a low-latency range compared to a model trained solely on offline data. After training these models, we applied the decoding policies in Section 2.1 to the ST model for controlling latency ranges.

2.3 Simultaneous Speech-to-Speech Translation

Our English-to-Japanese speech-to-speech simultaneous translation is a cascade of the speech-to-text

translation model (Section 2.2) and the incremental TTS module. In decoding steps, prefixes generated from the translation model are passed to the TTS module incrementally. Then, the TTS module judges whether to wait for more inputs or generate a partial hypothesis.

2.3.1 Incremental Text-to-Speech Synthesis

Incremental TTS consists of three modules: a phoneme estimator with a prosodic symbol for the Japanese language, an acoustic feature predictor, and a neural vocoder.

The phoneme estimator predicts the phonemes of SimulST outputs and prosodic symbols in the Japanese language in parallel using a Transformer model. This module uses three prosodic symbols to represent rising and falling pitches, and phrase boundary. It works simultaneously with the input based on the AlignAtt policy using models trained in full-sentence conditions. In TTS, it is assumed that there is monotonicity between input and output sequences, so there is little need to make delayed decisions and reorder, as is the case in LA. Therefore, we applied AlignAtt to TTS in this study. We modified the original Transformer architecture by adding two embedding input layers and two linear output layers to its decoder. The self-attention mask was applied to both the encoder and the decoder sides because the subsequent sequences should not be used in the inference time in the incremental condition.

The acoustic feature predictor predicts acoustic features from the phonemes and prosodic symbols mentioned above, and then the neural vocoder synthesizes speech in parallel. Its acoustic model is based on FastPitch (Łańcucki, 2021) with an additional adapter as an average phoneme power predictor. Its encoder uses two independent embedding layers for phoneme and prosodic sequences and concatenates their embedding vectors into a single sequence as the input to the Transformer model. Fastpitch estimates an acoustic feature sequence with predicted duration, pitch, and power in parallel. Parallel WaveGAN synthesizes a speech waveform for the given acoustic features and noise sequences.

3 Experimental Setting

3.1 Data

3.1.1 Simultaneous Speech-to-Text Translation

We trained our multilingual ST model on MuST-C v2.0 (Di Gangi et al., 2019) and CoVoST-2 (Wang et al., 2020) for all language pairs: English-to-German (En-De), English-to-Japanese (En-Ja), and English-to-Chinese (En-Zh). For the En-De setting, we also used MuST-C v1.0, Europarl-ST (Iranzo-Sánchez et al., 2020), and TED-LIUM (Rousseau et al., 2012). In our training data, the development and test portions of CoVoST-2 and Europarl-ST were also included. We used the MuST-C v2.0 tst-COMMON data as the evaluation data. We tokenized all of the text data in the corpora using a multilingual SentencePiece tokenizer with 250,000 subword units, distributed with the mBART50 model.

For the En-Ja setting, we trained a model that applied a data filtering approach on the prefix translation pairs for the Bilingual Prefix Alignment data. We empirically set the ratio of the number of samples in the input speech to the number of tokens in the output at 4000. Any utterance exceeding the maximum ratio was excluded from the training data. In order to prevent discrepancies in sentence structure and word order between the source and target languages in fine-tuned models and thus avoid favoring shorter output.

3.1.2 Incremental Text-to-Speech Synthesis

We used the JSUT corpus (Sonobe et al., 2017) for training our FastPitch and Parallel WaveGAN. The numbers of sentences in the training, development, and test data were 7196, 250, and 250, respectively. For JSUT labels, we used the open-source repository . We used the Balanced Corpus of Contemporary Written Japanese (Maekawa et al., 2014) for training the phoneme and prosodic symbol estimation model. These symbols were obtained from the text using Open Jtalk for training the estimation system. The same algorithm converted these symbols (Kurihara et al., 2021), and symbols were separated into two sequences by adding blank tokens in prosodic symbols. The training, development, and test data were approximately 1.4 M, 10 K, and 10 K sentences, respectively. We also used the training portion of MuST-C as additional training data.

---

https://github.com/r9y9/jsut-lab
https://open-jtalk.sourceforge.net

3.2 Simultaneous Speech-to-Text Translation

We developed an end-to-end speech-to-text model by initializing it with two pre-trained models: Hu-BERT for the speech encoder and mBART50 for the text decoder. Furthermore, the encoder and decoder are interconnected via Inter-connection (Nishikawa and Nakamura, 2023) and a length adapter (Tsiamas et al., 2022). Speech input is provided as waveforms sampled at a rate of 16 kHz, which are then normalized to have zero mean and unit variance.

We applied checkpoint averaging to the offline SimulST model. During checkpoint averaging, model checkpoints were saved every 1000 training steps, and the averaged parameter values from the five best models, based on loss in the development data, were selected for the final model.

Subsequently, one epoch of fine-tuning was conducted on the training data, focusing solely on prefix alignment pairs in MuST-C v2. For this fine-tuning stage, the learning rate was reduced to 2.5×10^{-5}, using translation pairs obtained via Bilingual Prefix Alignment.

For our SimulST strategies, we implemented both Local Agreement and AlignAtt policies. Specifically, we used Local Agreement with $n = 2$ (LA-2). To adaptively control the quality-latency trade-off, we varied the chunk size from 200 to 1000 ms. During hypothesis generation for input chunks, a beam search with a beam size of five was employed. For the AlignAtt policy, we set the chunk size to 800 ms. In AlignAtt, the parameter f directly governs the model's latency: smaller values of f imply that fewer frames are considered inaccessible by the model, thereby reducing the likelihood of the stopping condition being met and the resulting lower latency occurring. To adjust the quality-latency trade-off, we varied the parameter f from 1 to 12. See Appendix A for the detailed parameters of the speech-to-text model.

3.3 Simulaneous Speech-to-Speech Translation

Our simultaneous speech-to-speech system was a cascade of the speech-to-text translation module and the incremental TTS module. The parameter settings for the translation module were the same as those for the speech-to-text model, as described in Section 3.2

3.3.1 Incremental Text-to-Speech Synthesis

The incremental TTS is composed of three modules: a phoneme estimator with a prosodic symbol for the Japanese language, an acoustic feature predictor, and a neural vocoder.

For the phoneme estimator, the input vocabulary size was set to 21001. The output vocabulary was set to 40 for phoneme and 4 for prosodic symbols. The parameter of the AlignAtt policy f was set to 1 in the phoneme and prosodic symbol estimation modules. See Appendix B for the detailed parameters of the TTS model.

Speech was downsampled from 48 kHz to 22.05 kHz, and an 80-dimensional Mel spectrum was used for the acoustic features. The size of the Fourier transform, frameshift length, window length, and window function were 2048, 10 ms, 50 ms, and Hann window, respectively.

Our acoustic feature predictor mostly followed FastPitch structures, and the power predictor was added behind the pitch predictor.

For the neural vocoder, experimental conditions for Parallel WaveGAN were the same as in the original paper, except for the parameters related to acoustic features and speech.

3.4 Evaluation

We assessed our systems using the SimulEval (Ma et al., 2020a) toolkit and evaluated the translation quality of the SimulST systems using BLEU with sacreBLEU . We also measured translation latency by the following metrics:

- Average Lagging (AL) (Ma et al., 2019)
- Length Adaptive Average Lagging (LAAL) (Papi et al., 2022b)
- Average Token Delay (ATD) (Kano et al., 2024)
- Average Proportion (AP) (Cho and Esipova, 2016)
- Differentiable Average Lagging (DAL) (Cherry and Foster, 2019)

For the SimulS2S system, translation quality was evaluated using BLEU scores obtained after transcribing the output speech with Whisper (Radford et al., 2022) (ASR_BLEU). Translation latency was evaluated using ATD along with Start_Offset and End_Offset (Agarwal et al., 2023).

AL is a widely used latency metric for both text-to-text and speech-to-text simultaneous translation. However, while AL focuses on the time translation

Table 1: Results of submitted speech-to-text systems on MuST-C v2 tst-COMMON

Language pair	Chunk size	BLEU	LAAL	AL	AP	DAL	ATD
En-De	960 ms	29.978	2193.352	1973.799	0.846	2863.481	1887.436
En-Ja	835 ms	15.329	2269.591	1868.759	0.893	2878.447	541.729
En-Zh	910 ms	22.300	2245.997	1959.588	0.839	2811.262	897.994

(a) BLEU and AL in En-De   (b) BLEU and AL in En-Ja   (c) BLEU and AL in En-Zh
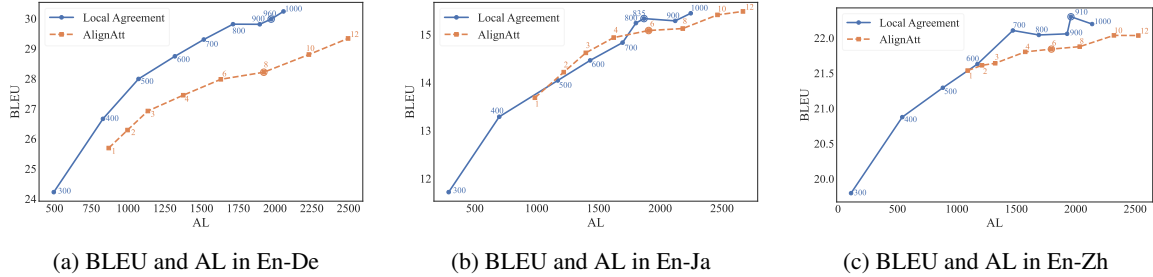
Figure 1: Results of **Local Agreement** and **AlignAtt** policies with AL on the speech-to-text systems. Circled dot in LA graph indicates our submitted system. Circled dot in AlignAtt graph indicates the best model satisfying the task requirement of IWSLT 2024 Shared Task.

(a) BLEU and AL_CA in En-De   (b) BLEU and AL_CA in En-Ja   (c) BLEU and AL_CA in En-Zh
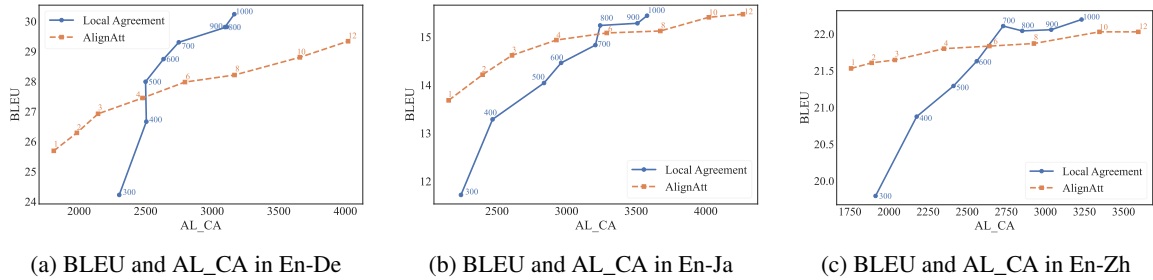
Figure 2: Results of **Local Agreement** and **AlignAtt** policies with AL_CA on the the speech-to-text systems

Table 2: Results of offline ST in submitted speech-to-text systems on MuST-C v2 tst-COMMON

Language pair	BLEU
En-De	31.00
En-Ja	15.98
En-Zh	24.98

begins, it does not adequately consider the time each input chunk's translation ends. In scenarios where speech segments are generated sequentially, as in speech-to-speech translation, the translation output may be delayed if the preceding outputs occupy the speech output channel. Consequently, AL may not be suitable for evaluating the latency of speech-to-speech simultaneous translation. Instead, we employ ATD, which includes delays caused by output in the latency calculation. ATD computes

delays by calculating the average time difference between each source token and its corresponding target token. In the SimulEval setup, assuming each word requires 300 ms to be spoken, both the input and output speech are segmented into 300-ms intervals, treating these segments as tokens for ATD calculations.

4 Experiment Results

4.1 Simultaneous Speech-to-Text System

We chose one submission for each language direction, ensuring that the settings met the task requirement of $AL \leq 2$ sec. The submission model is based on the LA policy, since it outperformed the AlignAtt policy used in earlier models.

4.1.1 NAIST 2023 model vs. 2024 model

Table 1 shows the results of the submitted speech-to-text systems evaluated on MuST-C v2 tst-COMMON. Although the system architecture of our submitted models was the same as that of last year's models, the chunk size settings were different in every language pair. Using different chunk size settings slightly improved the BLEU scores in every language pair (see Appendix C for the scores for our 2023 submission). We also show the results of the offline ST in submitted speech-to-text systems on MuST-C v2 tst-COMMON in Table 2.

4.1.2 Local Agreement vs. AlignAtt Policies

Figure 1 shows BLEU and AL trade-offs in non-computation-aware conditions. When comparing the results of the LA and AlignAtt policies, there was little difference observed in En-Ja (Figure 1 (b)), while there were relatively large gaps in BLEU in En-De and En-Zh, especially in the high latency region (Figures 1 (a) and (c)).

Figure 2 shows the BLEU and AL trade-offs in computation-aware conditions. In all language pairs, the AlignAtt policy was better in the low-latency region, while the LA policy was better in the high-latency regions.

4.1.3 Non-Computation-Aware vs. Computation-Aware Latency

The quality-latency trade-off results differed significantly between the non-computation-aware and the computation-aware conditions. The LA policy requires a relatively long computation time to obtain the longest common prefixes. This is especially true when the source speech is divided into many small segments. Therefore, the latency increases significantly when a small chunk size is set (see Figure 2).

The main constraint of the IWSLT 2024 Shared Task (*i.e.*, latency is measured in a non-computation-aware setting) may have been advantageous for the LA policy. In fact our LA-based system outperformed our AlignAtt-based one. However, in reality, the LA policy is time-consuming, and thus the AlignAtt policy may be better suited to practical applications.

4.2 Simultaneous Speech-to-Speech System

We submitted a model with the LA policy for the En-Ja speech-to-speech track. We selected a model configured with a chunk size of 950 ms, which satisfies the task requirement Start_Offset ≤ 2.5 sec. Table 3 shows the results of our speech-to-speech model (LA (NAIST 2024)). We also developed a model with the AlignAtt policy, but the LA model achieved higher ASR_BLEU than the AlignAtt model. The quality-latency trade-offs in non-computation-aware and computation-aware conditions are shown in Figures 3 and 4.

4.2.1 NAIST 2023 model vs. 2024 model

Our submitted model outperformed our last year's submission (LA (NAIST 2023)). We compared our 2024 submission with the 2023 one to clarify what contributed to improving the score. The significant difference between the two systems lies in the upgraded TTS, which has an estimation model based on Transformer architecture with the AlignAtt policy (see Section 3.3).

When comparing the output from the speech translation modules, there was little difference in BLEU scores between the two systems (2023 system: 14.93; 2024 system: 15.44) . However, the performance of our 2024 system, which was measured by ASR_BLEU, was more than 2 points higher than that of our 2023 system. The results suggest that our new TTS contributed to the improved score. We listened to samples of synthesized speech and observed that the outputs from the 2024 system tended to be more natural in accent and intonation compared to those from the 2023 system.

4.2.2 Local Agreement vs. AlignAtt Policies

We further compared the model with the LA policy (our 2024 submission) with the model with the AlignAtt policy. Comparing the translation modules of the two systems, the difference in translation quality measured by BLEU was about 0.4 points (15.44 and 15.09 for the LA and the AlignAtt models, respectively). This gap is almost the same as the gap in the evaluation scores of the speech-to-speech systems (ASR_BLEU, see Table 3).

Although no difference was observed in BLEU scores, a comparison between the output from the speech-to-speech system (*i.e.*, transcribed speech) with the output from the translation module suggests that the policy difference affects the TTS performance. We extracted sentences that satisfy the following criteria: (1) translations from the translation modules were identical between the two policies, but (2) transcribed speeches were different between the two systems.

The chunk size setting for the 2024 speech-to-speech system was different from that for the 2024 speech-to-text system.

Table 3: Results of submitted SimulS2S system on the MuST-C v2 tst-COMMON

System	Chunk size	ASR_BLEU	Start_Offset	End_Offset	ATD
LA (NAIST 2023)	650 ms	9.873	2495.010	4134.752	3278.809
LA (NAIST 2024)	950 ms	12.082	2425.485	3745.743	3792.405
AlignAtt	800 ms (f=6)	11.650	2493.908	3505.377	3682.920

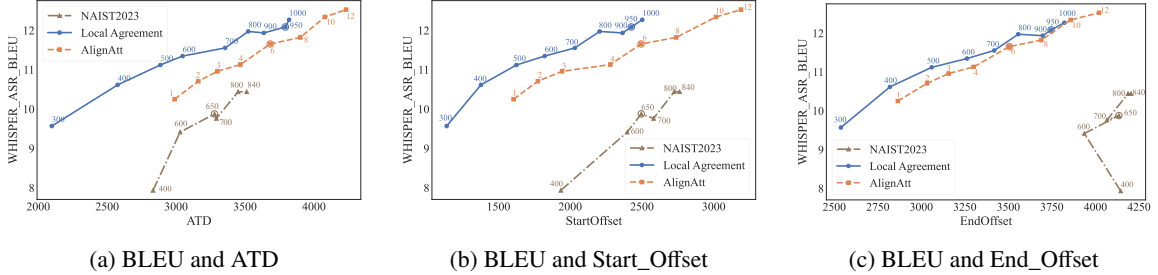(a) BLEU and ATD   (b) BLEU and Start_Offset   (c) BLEU and End_Offset

Figure 3: Results of **Local Agreement** and **AlignAtt** policies with ATD, Start_Offset, and End_Offset on speech-to-speech systems. Circled dot in LA graph indicates submitted system. Circled dot in AlignAtt graph indicates the best model satisfying the task requirement of IWSLT 2024 Shared Task.

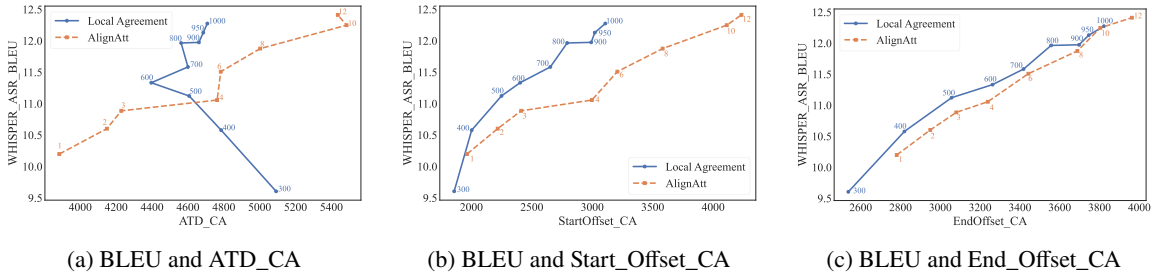(a) BLEU and ATD_CA   (b) BLEU and Start_Offset_CA   (c) BLEU and End_Offset_CA

Figure 4: Results of **Local Agreement** and **AlignAtt** policies with ATD_CA, Start_Offset_CA and End_Offset_CA on speech-to-speech translation systems.

We computed BLEU scores using extracted sentences (N=414) while regarding the outputs from the translation modules as references. The score for the LA policy was more than 2 points higher than that for the AlignAtt policy (67.79 and 65.46, respectively). In addition, the transcribed speech for AlignAtt was shorter than that for LA (sys_len: 6364 and 6464, respectively). These results suggest that the manner of passing the translations to the TTS was different between the two decoding policies (*e.g.*, timing) and affected the TTS performance.

Our analysis suggests that the output from the LA policy was more suitable for our TTS than that from the AlignAtt policy because the LA policy generated longer partial output with more confident agreement. On the other hand, the AlignAtt policy tended to generate prefixes whose boundaries did

not correspond to meaningful units and sometimes divided a word in the middle of it. Figure 5 shows an example of the timing difference in passing the translations to the TTS. This figure compares the output prefixes generated from the translation modules with different decoding policies and the output prefixes generated from the TTS module along with the timing information. In this example, the translations generated from the translation modules are identical between the LA and AlignAtt policies. However, the prefixes (see Tgt text in Figure 5) and the timing when they were passed to the TTS module were different between the two decoding policies. In this example, the LA policy tended to generate semantically coherent prefixes, which resulted in more successful output from the TTS module (see Tgt speech). On the other hand, the AlignAtt policy divided the word "フォーミュ
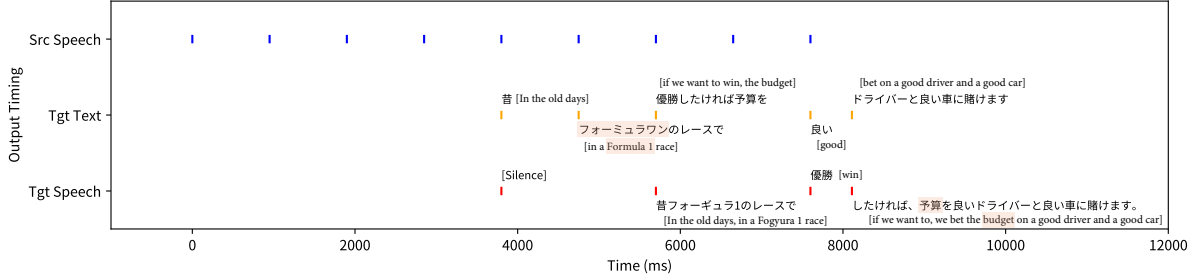
(a) LA (chunk size = 950 ms)
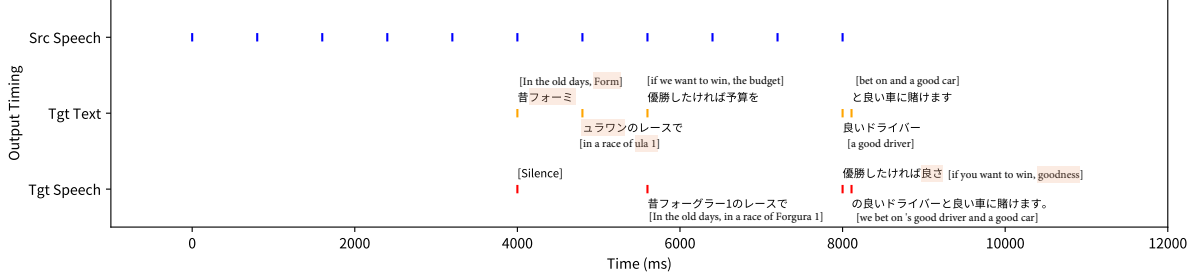
(b) AlignAtt (chunk size = 800 ms, $f = 6$)

Figure 5: Example of timing difference in passing translations to the TTS between the LA and AlignAtt policies. Translations generated by speech-to-text models were identical between the two policies, but outputs from the TTS module were different.

ラ ワ ン [Formula 1]" into two prefixes, "フォーミ [Form]" and "ュ ラ ワ ン [ula 1]." When the boundaries of the prefixes do not correspond to the meaning units or words are divided into prefixes, it might be difficult to capture the context of a sentence, which results in poor performance of the TTS module. In this example, the word "予算 [budget]" (pronounced as *yosan*) was wrongly recognized as "良さ [goodness]" (pronounced as *yosa*) in the system with the AlignAtt policy. The results suggest that feeding stable prefixes to the TTS module is important in our speech-to-speech system. Future study will involve making the AlignAtt policy generate more stable prefixes.

4.2.3 Non-Computation-Aware vs. Computation-Aware Latency

Figures 3 and 4 show the results in non-computation-aware and computation-aware settings, respectively. When the latency was measured by the Start_Offset and the End_Offset, there were no large differences between the results in non-computation-aware and computation-aware settings. However, when latency was measured by ATD, the quality-latency trade-offs exhibited different trends in non-computation-aware and computation-aware settings.

Start_Offset does not include computation time

as a delay because Start_Offset is measured only at the start of translation. Therefore, Start_Offset is not appropriate as the latency metric in computation-aware settings. Moreover, Start_Offset and End_Offset measure the delay at a single point in the translation and does not consider the delays in the middle section of the translation.

In contrast, ATD measures the delay at multiple points and has a higher correlation with Ear-Voice Span, which is often used as a reference latency metric in human interpretation research (Kano et al., 2024). As the segments become smaller, the number of segments increases. This increases the number of comparison processes at the inference of LA. Therefore, the computation time becomes larger as the segment size becomes smaller and BLEU becomes lower in the low-latency range of LA, which is only shown in Figure 4a.

In a computation-aware setting, we observed that the AlignAtt policy outperformed the LA policy in the low-latency region (Figure 4). In practical situations, the LA policy might be time-consuming for a speech-to-speech system. One future direction would be improving the performance of a model with the AlignAtt policy.

5 Conclusions

In this paper, we described our SimulST systems for the IWSLT 2024 Simultaneous Speech Translation task. Experimental results demonstrated the effectiveness of AlignAtt by comparison to Local Agreement in terms of computation-aware latency, especially in the low-latency range. Our speech-to-speech translation system also showed the effectiveness of applying AlignAtt to the TTS model and resulted in better performance compared to our IWSLT 2023 system. This time, our speech-to-text method used HuBERT with the mBART model, while our TTS method only used the Parallel WaveGAN vocoder. In the future, we will investigate other methods such as WavLM (Chen et al., 2022) and Hi-Fi GAN (Kong et al., 2020).

Acknowledgements

References

Milind Agarwal, Sweta Agrawal, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Mingda Chen, William Chen, Khalid Choukri, Alexandra Chronopoulou, Anna Currey, Thierry Declerck, Qianqian Dong, Kevin Duh, Yannick Estève, Marcello Federico, Souhir Gahbiche, Barry Haddow, Benjamin Hsu, Phu Mon Htut, Hirofumi Inaguma, Dávid Javorský, John Judge, Yasumasa Kano, Tom Ko, Rishu Kumar, Pengwei Li, Xutai Ma, Prashant Mathur, Evgeny Matusov, Paul McNamee, John P. McCrae, Kenton Murray, Maria Nadejde, Satoshi Nakamura, Matteo Negri, Ha Nguyen, Jan Niehues, Xing Niu, Atul Kr. Ojha, John E. Ortega, Proyag Pal, Juan Pino, Lonneke van der Plas, Peter Polák, Elijah Rippeth, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Yun Tang, Brian Thompson, Kevin Tran, Marco Turchi, Alex Waibel, Mingxuan Wang, Shinji Watanabe, and Rodolfo Zevallos. 2023. FINDINGS OF THE IWSLT 2023 EVALUATION CAMPAIGN. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 1–61, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marcely Zanon Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanuel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gahbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde,

Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, John Ortega, Juan Pino, Elizabeth Salesky, Jiatong Shi, Matthias Sperber, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alexander 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)*, pages 98–157, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Naveen Arivazhagan, Colin Cherry, Isabelle 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. IEEE.

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.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Colin Cherry and George Foster. 2019. Thinking slow about latency evaluation for simultaneous machine translation. *arXiv preprint arXiv:1906.00048*.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *CoRR*, abs/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.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. 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 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Christian Fügen, Alex Waibel, and Muntsin Kolss. 2007. Simultaneous translation of lectures and speeches. *Machine translation*, 21:209–252.

Ryo Fukuda, Yuta Nishikawa, Yasumasa Kano, Yuka Ko, Tomoya Yanagita, Kosuke Doi, Mana Makinae, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura. 2023. NAIST simultaneous speech-to-speech translation system for IWSLT 2023. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 330–340, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233.

J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. 2020. Librilight: A benchmark for asr with limited or no supervision. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673. https://github.com/facebookresearch/libri-light.

Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2022. Simultaneous neural machine translation with prefix alignment. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 22–31, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2024. Average token delay: A duration-aware latency metric for simultaneous translation. *Journal of Natural Language Processing*, 31(3):To appear.

Eugene Kharitonov, Morgane Rivière, Gabriel Synnaeve, Lior Wolf, Pierre-Emmanuel Mazaré, Matthijs Douze, and Emmanuel Dupoux. 2020. Data augmenting contrastive learning of speech representations in the time domain. *arXiv preprint arXiv:2007.00991*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33:17022–17033.

Kiyoshi Kurihara, Nobumasa Seiyama, and Tadashi Kumano. 2021. Prosodic features control by symbols as input of sequence-to-sequence acoustic modeling for neural tts. *IEICE Transactions on Information and Systems*, 104(2):302–311.

Danni Liu, Gerasimos Spanakis, and Jan Niehues. 2020. Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection. In *Proc. Interspeech 2020*, pages 3620–3624.

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 Pino, and Philipp Koehn. 2020b. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 582–587, Suzhou, China. Association for Computational Linguistics.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. 2014. Balanced corpus of contemporary written japanese. *Language resources and evaluation*, 48:345–371.

Ha Nguyen, Yannick Estève, and Laurent Besacier. 2021. An empirical study of end-to-end simultaneous speech translation decoding strategies. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7528–7532. IEEE.

Yuta Nishikawa and Satoshi Nakamura. 2023. Interconnection: Effective Connection between Pretrained Encoder and Decoder for Speech Translation. In *Proc. INTERSPEECH 2023*, pages 2193–2197.

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.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022a. Does simultaneous speech translation need simultaneous models? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 141–153, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022b. Over-generation cannot be rewarded: Length-adaptive average lagging for simultaneous speech translation. In *Proceedings of the Third Workshop on Automatic Simultaneous Translation*, pages 12–17, Online. Association for Computational Linguistics.

Sara Papi, Marco Turchi, and Matteo Negri. 2023. AlignAtt: Using Attention-based Audio-Translation Alignments as a Guide for Simultaneous Speech Translation. In *Proc. INTERSPEECH 2023*, pages 3974–3978.

Peter Polák, Ngoc-Quan Pham, Tuan Nam Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, Ondřej Bojar, and Alexander Waibel. 2022. CUNI-KIT system for simultaneous speech translation task at IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 277–285, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.

Anthony Rousseau, Paul Deléglise, and Y. Estève. 2012. Ted-lium: an automatic speech recognition dedicated corpus. In *International Conference on Language Resources and Evaluation*.

Ryosuke Sonobe, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2017. Jsut corpus: free large-scale japanese speech corpus for end-to-end speech synthesis. *arXiv preprint arXiv:1711.00354*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Ioannis Tsiamas, Gerard I. Gállego, Carlos Escolano, José Fonollosa, and Marta R. Costa-jussà. 2022. Pretrained speech encoders and efficient fine-tuning methods for speech translation: UPC at IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 265–276, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu. 2020. CoVoST: A diverse multilingual speech-to-text translation corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4197–4203, Marseille, France. European Language Resources Association.

Mahsa Yarmohammadi, Vivek Kumar Rangarajan Sridhar, Srinivas Bangalore, and Baskaran Sankaran. 2013. Incremental segmentation and decoding strategies for simultaneous translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1032–1036, Nagoya, Japan. Asian Federation of Natural Language Processing.

Adrian Łańcucki. 2021. Fastpitch: Parallel text-to-speech with pitch prediction. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6588–6592.

A Speech-to-Text Parameter Settings

The speech encoder was initialized with HuBERT-Large, comprising a feature extractor trained on 60 K hours of unlabeled speech data from Libri-Light (Kahn et al., 2020), along with Transformer encoder layers. The feature extractor consists of seven convolutional layers with kernel sizes of (10, 3, 3, 3, 3, 2, 2), corresponding strides of (5, 2, 2, 2, 2, 2, 2), and 512 channels. The number of Transformer encoder layers is 24. The text decoder was initialized using the decoder component of mBART50. The decoder is composed of twelve Transformer layers, sharing an embedding layer and linear projection weights sized at 250,000. Each Transformer and feed-forward layer has dimensions of 1024 and 4096, respectively, with 16 attention heads. ReLU serves as the activation function, and layer normalization is applied before attention operations. The length adapter is implemented as a three-layer convolutional network featuring 1024 channels, a stride of 2, and a Gated Linear Unit (GLU) activation function. During training, each source audio was augmented (Kharitonov et al., 2020) prior to normalization, with a probability of 0.8. Multilingual models were trained using all of the data with a maximum source length of 400,000 frames and a target length of 1024 tokens. To achieve a batch size of approximately 32 million tokens, we employ gradient accumulation and data-parallel computations. We utilize the Adam optimizer with $\beta_1 = 0.99$, $\beta_2 = 0.98$, and a base learning rate of 2.5×10^{-4}. A tri-stage scheduler controls the learning rate, with warm-up, hold, and decay phases set to 0.15, 0.15, and 0.70, respectively. The initial and final learning rates are scaled to 0.01 compared to the base rate. Sentence averaging and gradient clipping of 20 are applied, along with a dropout probability of 0.1. Time masking is used for 10-length spans with a probability of 0.2, while channel masking is applied to 20-length spans with a probability of 0.1 in the output of the encoder's feature extractor. The loss function employed is cross-entropy with label smoothing of 20% probability mass.

B Incremental Text-to-Speech Parameter Settings

For the phoneme estimator, each Transformer layer, head, dimension of head, or dimension of Transformer model was 2, 8, 64, or 512, respectively. The embedded size of the encoder was the same as the dimension of the Transformer, and the embedding sizes of the decoder were 128 dimensions for the prosodic symbols and 512 dimensions for the phonemes. The batch size for training was 256. We used Adam optimizer with a learning rate of 0.1, $\beta_1 = 0.99$, $\beta_2 = 0.99$, and $\epsilon = 1e - 8$. The warmup scheduler is the same as that of the original Transformer. The size of the Fourier transform, frameshift length, window length, and window function were 2048, 10 ms, 50 ms, and Hann window, respectively. The changed settings were as follows: We used two embedding layers with a hidden size of 256, the hidden size in Transformer was 256, the number of heads was 2, the encoder and decoder had 4 layers, the first convolution layer in each FFT block in FastPitch had a kernel size of 3 and 256/1024 input/output channels, the second convolution layer in an FFT block had 1024/256 input/output channels with the same kernel size, the first convolution layer in each predictor had a kernel size of 3 and 256/256 input/output channels, and the second convolution layer in each predictor had 256/256 input/output channels with the same kernel size. We used Adam optimizer with a learning rate of 0.1, $\beta_1 = 0.99$, $\beta_2 = 0.99$, and $\epsilon = 1e - 9$. The batch size was 48. The schedule for the warmup followed FastPitch.

C NAIST 2023 Submission for Speech-to-Text

Table 4 shows the results for all chunk size settings for the En-De, En-Ja, and En-Zh models, respectively, used in our 2023 submission (Fukuda et al., 2023).

Table 4: Results of submitted speech-to-text systems on MuST-C v2 tst-COMMON in IWSLT 2023

Language pair	Chunk size	BLEU	LAAL	AL	AP	DAL	ATD
En-De	950 ms	29.975	2172.927	1964.329	0.846	2856.738	1893.749
En-Ja	840 ms	15.316	2290.716	1973.586	0.892	2889.950	547.752
En-Zh	700 ms	22.105	1906.995	1471.287	0.821	2436.948	667.780