Simultaneous Interpretation Corpus Construction by Large Language Models in Distant Language Pair

Yusuke Sakai*, **Mana Makinae***, **Hidetaka Kamigaito**, **Taro Watanabe**
Nara Institute of Science and Technology
{sakai.yusuke.sr9, makinae.mana.mh2, kamigaito.h, taro}@is.naist.jp

Abstract

In Simultaneous Machine Translation (SiMT), training with a simultaneous interpretation (SI) corpus is an effective method for achieving high-quality yet low-latency systems. However, constructing such a corpus is challenging due to high costs, and limitations in annotator capabilities, and as a result, existing SI corpora are limited. Therefore, we propose a method to convert existing speech translation (ST) corpora into interpretation-style corpora, maintaining the original word order and preserving the entire source content using Large Language Models (LLM-SI-Corpus). We demonstrated that fine-tuning SiMT models using the LLM-SI-Corpus reduces latencies while achieving better quality compared to models fine-tuned with other corpora in both speech-to-text and text-to-text settings. The LLM-SI-Corpus is available at https://github.com/yusuke1997/LLM-SI-Corpus.

1 Introduction

Simultaneous machine translation (SiMT)[1] (Luong and Manning, 2015; Gu et al., 2017; Ma et al., 2019; Arivazhagan et al., 2019) translates input in real-time by incrementally processing partial segments rather than waiting the whole sentence completion. While offline machine translation (MT) works without time restrictions, SiMT begins translating at certain points due to time limitations; therefore, balancing its latency and quality is crucial. This challenge is especially difficult in language pairs with drastically different word orders, such as English and Japanese (SVO vs. SOV) (He et al., 2015; Chen et al., 2021; Deng et al., 2023). To manage word order differences in simultaneous settings, one strategy is to maintain the source language word order as much as possible to keep up with the input, minimizing latency while maintaining quality (Cai et al., 2020; Han et al., 2021; Guo et al., 2023). To address the balance between quality and latency, the one of the best ways to learn this interpretation strategy for SiMT systems is to utilize simultaneous interpretation (SI) data to train the model (Ko et al., 2023). While several SI datasets have been proposed for English and Japanese, they remain relatively limited in size compared to MT corpora. Furthermore, acquiring this data is costly and resource-intensive, making manual dataset construction impractical for scaling.

Moreover, even if such issues were resolved, it remains uncertain whether professional SI transcripts are optimal for SiMT. The specialized na-

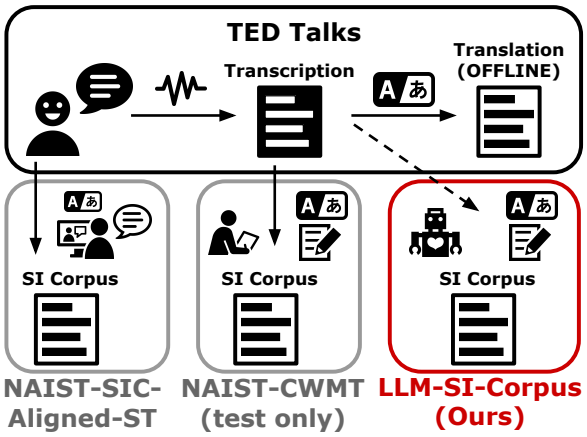

Figure 1: The corpora used in this study, each created from the same TED Talks data. TED Talks are accompanied by English-Japanese offline MT data. NAIST-SIC-Aligned-ST (Ko et al., 2023) is an SI dataset created by transcribing audio data of these talks by human interpreters. NAIST English-to-Japanese Chunk-wise Monotonic Translation Evaluation Dataset 2024 (NAIST-CWMT) (Fukuda et al., 2024) is manually created based on offline MT data from TED Talks, following the CWMT guideline (Okamura and Yamada, 2023), and used only for testing purposes. Our LLM-SI-Corpus was created by LLMs based on the CWMT guideline and comprises training, development, and test sets.

*These authors contributed equally to this work.
[1]Also, we called Simultaneous Speech Translation. We simplify the notation to SiMT in this paper for brevity.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 22375–22398
November 12-16, 2024

ture of SI causes translation quality to vary among interpreters due to differences in skills and experiences. Time constraints and cognitive overload in SI contribute to these variations, influenced by factors such as summarization, repetition, and omissions. Consequently, the quality of existing SI corpora is inconsistent, making them less faithful to the source and not ideal for training SiMT.

To address these challenges, Fukuda et al. (2024) manually created test data (chunk-wise) following Chunk-Wise Monotonic Translation (CWMT) guideline (Okamura and Yamada, 2023), with fluency and adequacy verified by professional interpreters. A key feature of chunk-wise is its monotonic alignment with the source, maintaining the entire source content, making it well-suited for the goals of SiMT. CWMT is designed for English-to-Japanese SI to reduce latency by segmenting sentences into grammatical chunks and translate sequentially. However, despite its potential, the reliance on human labor for dataset creation remains a significant barrier for scaling.

Therefore, we propose a method to convert existing speech translation (ST) corpora into SI-style data (LLM-SI-Corpus), closely maintaining the original word order and preserving the entire source content based on the CWMT guideline using Large language odels (LLMs) as shown in Figure 1. We demonstrated that fine-tuning SiMT models with the LLM-SI-Corpus, in both text-to-text and speech-to-text settings, achieves better translation quality with minimal latency compared to models fine-tuned with other corpora and the pretrained model.

To summarize, our contributions are as follows:

- We proposed a method for automatically constructing a training dataset for SiMT systems using LLMs following the CWMT guideline
- We constructed the LLM-SI-Corpus, a large-scale training dataset for SiMT.
- We confirmed that the LLM-SI-Corpus is effective in improving both translation quality and latency in SiMT systems.

2 Background and Related Work

2.1 Simultaneous Machine Translation

In SiMT, the model processes partial source sentences of length J to incrementally generate partial target sentences of length I, guided by its policy. Various policies have been proposed, primarily categorized as fixed and adaptive. Fixed policies (Dalvi et al., 2018; Ma et al., 2019; Elbayad et al., 2020; Zhang and Feng, 2021) decide READ/WRITE operations based on predefined rules, such as the wait-k policy (Ma et al., 2019), which reads k source tokens initially and then alternates between writing and reading one token. Conversely, adaptive policies (Zheng et al., 2020; Liu et al., 2020; Papi et al., 2023a,b) predict READ/WRITE operations based on the current source and target prefix, achieving a better balance between latency and translation quality.

2.2 SI Corpora

Existing SI corpora are constructed from real-time human interpretation. In English to Japanese, several SI corpora are constructed (Toyama et al., 2004; Shimizu et al., 2014; Doi et al., 2021). Doi et al. (2021) developed a large-scale SI corpus (NAIST-SIC) supporting both English to/from Japanese[2]. However, in the NAIST-SIC, most of the data lack sentence alignment, making them difficult to use for model training. To address this limitation, Zhao et al. (2024) proposed NAIST-SIC-Aligned for text-to-text alignment, and Ko et al. (2023) introduced NAIST-SIC-Aligned-ST for speech-to-text alignment, resulting in a parallel English-Japanese SI corpus available for use. Fukuda et al. (2024) constructed a test dataset from NAIST-SIC-Aligned-ST based on CWMT (described in Section 2.3). For the other language pairs, Pan (2019); Zhang et al. (2021) (English-Chinese), Kunz et al. (2021); Zhao et al. (2021); Macháček et al. (2021) (English-German), Paulik and Waibel (2009); Bernardini et al. (2016); Wang et al. (2021); Przybyl et al. (2022) (the other language pairs include English) have been established.

However, SI corpus construction requires considerable time, money, and effort, resulting in a small corpus size. To address this challenge, He et al. (2015) proposed a sentence rewriting method to automatically generate more monotonic translations for Japanese-to-English SiMT by defining syntactic transformation rules. However, spoken language presents challenges for syntactic parsing, and the rule-based approach often reduces fluency and is limited to specific language pairs, making it difficult to apply this method broadly.

[2]They provide only a part of English-to-Japanese data.

2.3 Chunk-Wise Monotonic Translation

Chunk-wise monotonic translation (CWMT) is strategy used by simultaneous interpreters, particularly for distant language pairs such as English and Japanese (Mizuno, 2016; Okamura and Yamada, 2023; Fukuda et al., 2024). This guideline addresses grammatical differences, as directly preserving the source word order could lead to unnatural translations in the target. To balance translation latency and quality when translating from English to Japanese, interpreters aim to maintain the sequential order of information chunks from the source as much as possible (Doi et al., 2021; Camayd-Freixas, 2011). Interpreters divide sentences into manageable chunks based on grammatical characteristics and translate them sequentially, preserving chunk order. Fukuda et al. (2024) defines these chunk boundaries and the chunking workflow using rule-based methods based on CWMT. The details of the guideline and workflow are described in Appendix A.

2.4 Style differences among SI, Offline Translation, and CWMT

There are significant style gaps among SI, offline translation, and CWMT as described in Fukuda et al. (2024); Ko et al. (2023). The examples are shown in Appendix B. The findings include:

- The SI translates the first half of the input earlier than the latter half with some unnaturalness and omission, whereas the offline translation preserves naturalness in Japanese through long-distance reordering from the input English (See Table 6 in Appendix B).
- The offline translation and CWMT both include all content words from the source; however, their distinction lies in the order. In offline translation, long-distance reordering occurs to preserve naturalness, whereas, in CWMT, the order of source language chunks is maintained with some unnaturalness (See Table 7 in Appendix B).

From this observation, both SI and CWMT prioritize aligning source inputs as closely as possible, whereas offline allows for long-distance reordering. The significant difference in word order between English and Japanese poses a substantial challenge in SI, as highlighted in a prior study (Mizuno, 2016). Under the real SI scenario, interpreters prioritize delivering interpretation simultaneously to convey content promptly and preserve their working memory, which may involve some omission and summarization. The current limitation in CWMT lies in their approach to maintaining fluency. Thus, it is challenging to do automatically, and it takes a high cost when annotating manually.

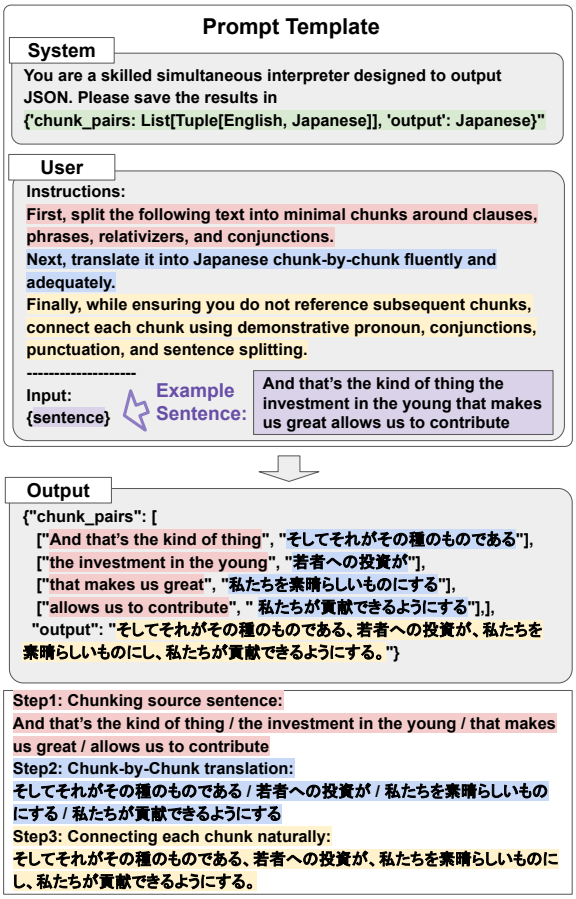

Figure 2: The prompt template used for constructing the LLM-SI-Corpus based on the CWMT workflow. Each color indicates a prompt and its corresponding outputs.

3 SI-Corpus Construction with LLMs

To address the limitations of the current SI corpus, we leverage LLMs, which are known for their high translation performance and ability to perform purpose-specific translations based on instructions (Moslem et al., 2023; Zheng et al., 2024). For our purpose, we follow the CWMT guidelines to automatically convert ST into SI corpora using LLMs to be more monotonic while maintaining fluency, making it suitable for SiMT training.

3.1 Prompt for Creating LLM-SI-Corpus

Our prompt is based on CWMT guidelines by Okamura and Yamada (2023). CWMT has three processes as described in Section 2.3: chunking based

on grammatical characteristics, translation of each chunk, and concatenating the translated chunks into sentences. We simplify the process compared to the original to make it more suitable for LLMs[3], as described in Figure 2.

For chunking, we designed the instruction to split based on grammatical features, specifically around clauses, phrases, relativizers, and conjunctions. Next, LLMs translate each chunk while maintaining fluency and adequacy. Finally, LLMs generate the CWMT output by connecting chunks using demonstrative pronouns, conjunctions, and punctuation to maintain the original chunk sequential order while ensuring you do not reference subsequent chunks. These processes are summarized in a single prompt[4]. The outputs are formatted in JSON [5] to ensure that all operations are performed according to the instructions, without any shortcuts, and the output is generated at each step[6].

3.2 Dataset Selection

In this study, we focus on the English-Japanese direction and selected the NAIST-SIC-Aligned-ST corpus (Ko et al., 2023)[7] as the seed dataset. As shown in Figure 1, the NAIST-SIC-Aligned-ST corpus is based on TED Talks, which consist of audio, transcriptions, and sentence-by-sentence translations of the transcripts (offline translations), with the addition of interpreters' interpretations. The data size for training, development, and testing is 65,083, 165, and 511 sentences, respectively. This choice enables a comparison among models fine-tuned with the LLM-SI-Corpus, interpreter transcriptions, and offline translation to investigate which data better addresses the tradeoff between latency and quality.

[3]Although the operation of LLMs is not always stable, Section 4 shows that LLMs successfully produced CWMT-like monotonic sentences, achieving our goal of constructing the dataset to improve both latency and quality in SI models at a low cost.

[4]In the pilot study, we found similar results when we input data for each process separately as a pipeline or all at once into the LLMs. Thus, to address the cost issue, we chose to input all data at once as the prompt.

[5]https://platform.openai.com/docs/guides/text-generation/json-mode

[6]We also employ various prompt tuning techniques, such as adding specific words to the instructions and using delimiters. Most of the prompt tuning techniques used in this study are described in Bsharat et al. (2024).

[7]This type of dataset is currently only available in the NAIST-SIC dataset family (Shimizu et al., 2014; Doi et al., 2021; Zhao et al., 2024; Ko et al., 2023; Fukuda et al., 2024); therefore, the work is limited to the En-Ja direction, and we plan to explore other language pairs in future work.

	Source: OFFLINE ⇒ Target:			
Metrics (↑)	GPT-4	GPT-3.5	Chunk-wise	SIC
BLEU	13.8	15.5	**16.2**	7.9
BLEURT	55.9	56.0	**59.0**	40.8
COMET	82.3	83.2	**84.3**	71.7
COMET-QE	82.6	82.8	**82.9**	63.1

Table 1: Quality comparison between OFFLINE and each SI corpus. BLEU and ChrF indicate the similarities of textual alignment. BLEURT, COMET, and COMET-QE compare semantic similarity, as shown in Table 3.

3.3 LLM-SI-Corpus Construction by LLMs

We created two corpora using LLMs, GPT-3.5[8] (Ouyang et al., 2022) and GPT-4[9] (OpenAI et al., 2024) from the transcription of NAIST-SIC-Aligned-ST. GPT-4 is known to have a higher ability to follow instructions and generate higher-quality outputs than GPT-3.5. Therefore, we also examine the differences in LLM abilities by comparing the two corpora. The dataset size matches the numbers for NAIST-SIC-AlignST. The total cost of data creation was 20 dollars (0.0003 dollars per sentence) for GPT-3.5 and 400 dollars (0.006 dollars per sentence) for GPT-4.

4 Quality Analysis of LLM-SI Corpus

Quality Table 1 shows a quality comparison of the test data with BLEU (Post, 2018), BLEURT (Pu et al., 2021), COMET (Rei et al., 2020), and COMET-QE (Chimoto and Bassett, 2022). OFFLINE refers to the offline translation from NAIST-SIC-Aligned-ST (Ko et al., 2023). GPT-4 and GPT-3.5 are from the LLM-SI-Corpus, which was created from NAIST-SIC-Aligned-ST. SIC is the transcript of professional interpreters from NAIST-SIC-Aligned-ST. Chunk-wise comes from the NAIST English-to-Japanese Chunk-Wise Monotonic Translation Evaluation Dataset (Fukuda et al., 2024). The numbers indicate that Chunk-wise is the closest to OFFLINE across all evaluation metrics. GPT-3.5 and GPT-4 achieve comparable quality, while SIC demonstrates significantly lower quality compared to OFFLINE. Furthermore, focusing on COMET-QE, both the LLM-Corpus (GPT-3.5 and GPT-4) and Chunk-wise achieve equivalent quality, suggesting that LLMs have the capability to create data with the same quality as Chunk-wise which created manually.

[8]gpt-3.5-turbo-0125

[9]gpt-4-0125-preview

OFFLNE	Chunk-wise	GPT-3.5	GPT-4	SIC
0.478	**0.784**	**0.773**	**0.764**	0.471

Table 2: The table compares word order monotonicity across different dataset relative to the source. Chunk-wise and the LLM-Corpus (GPT-3.5 and GPT-4) demonstrate the same level of monotonicity.

Monotonicity We analyzed the word alignment and evaluated the extent to which monotonicity improved between the source and different reference for GPT-3.5, GPT-4, SIC, OFFLINE, and Chunk-wise. We used Awesome-Align (Dou and Neubig, 2021) to compare the source and reference, and evaluated the alignment consistency using Spearman's correlation coefficient. Table 2 shows that GPT-3.5/4 has improved monotonicity compared to OFFLINE and has achieved similar monotonicity to the Chunk-wise, which involved human labor. This indicates that the LLM-SI Corpus, which follows the CWMT guideline for corpus construction, contributes to the monotonicity improvement and that LLM is an effective substitute for manual work. On the other hand, the monotonicity of SIC is comparable to that of OFFLINE, suggesting that the transcription of a simultaneous interpreter does not necessarily ensure monotonicity with the source. This indicates that such data may not be ideal for training SiMT models aimed at achieving both minimal latency and high quality.

5 Experimental Setup

To evaluate the effectiveness of the LLM SI-Corpus, we conducted experiments in speech-to-text settings. We also conducted text-to-text experiments, as presented in Appendix C, which showed a similar trend to the speech-to-text results. We implemented the baseline using Fairseq (Ott et al., 2019; Wang et al., 2020) and SimulEval (Ma et al., 2020).

Speech-to-Text Settings Following the settings of Fukuda et al. (2023); Ko et al. (2023), we employ pretrained language models for both encoder and decoder using Fairseq (Ott et al., 2019; Wang et al., 2020), and integrating into the Transformer architecture (Vaswani et al., 2017). We used Hubert-Large (Hsu et al., 2021) as the encoder, and we used mBART50 (Tang et al., 2021) as the decoder. We trained the model with MuST-C v2.0 (Cattoni et al., 2021) as continuous pertaining, and then fine-tuned the models for 3K steps, evaluating their performance every 200 steps, and terminated the fine-tuning if there was no improvement in the loss score for eight consecutive evaluations. For decoding policy, we applied test-time wait-k (Ma et al., 2019)[10] to determine whether the tradeoff between latency and quality is solely a result of differences in the dataset. The value of wait-k ranges from 1 to 17 at two intervals. One unit for k was set to 160 frames and when k = 3, after reading 3 $\times$ 160 frames, the model would WRITE and READ alternately. The detailed settings are described in Appendix C.

Training Datasets We used MuST-C v2.0 for En-Ja (Di Gangi et al., 2019) for pre-training and it is as the baseline (Pretrain). We then fine-tuned the pre-trained model using different types of data: offline ST translation data (OFFLINE), NAIST-SIC-Aligned-ST (SIC), which consists of human interpretation transcriptions, and two versions of the LLM-SI-Corpus (GPT-4 and GPT-3.5). All fine-tuning datasets come from the same audio sources, allowing for a comparison of the impact of different translation styles from each dataset.

Evaluation Datasets We choose three evaluation dataset: tst-COMMON from the MuST-C v2.0 (tst-COMMON) (Di Gangi et al., 2019), the test dataset from NAIST-SIC-Aligned-ST[11] (SIC-test), and NAIST English-to-Japanese Chunk-wise Monotonic Translation Evaluation Dataset 2024[12] (Chunk-wise). These choices are based on differences in translation styles, which could influence evaluations using reference-dependent metrics. tst-COMMON represents an offline translation style, where frequent word order reordering occurs, but the source content is preserved in the target. SIC-test consists of interpreter transcriptions, where some source content is omitted due to time constraints and high cognitive load. Chunk-wise aligns the target word order with the source as much as possible while preserving the source content.

Evaluation Metrics Table 3 shows a list of translation quality evaluation used in our experiments[13], highlighting the characteristics of each metric. BLEU (Post, 2018) focuses on textual n-

[10] We followed examples in GitHub repository: https://github.com/ahclab/naist-simulst

[11] https://dsc-nlp.naist.jp/data/NAIST-SIC/Aligned-ST, (Ko et al., 2023)

[12] https://dsc-nlp.naist.jp/data/NAIST-SIC/Aligned-Chunk_Mono-EJ, (Fukuda et al., 2024)

[13] We also evaluated with BERTScore (Zhang et al., 2020), but the trend is very similar to BLEURT.

Quality Metrics	Textual	Meaning	Reference	Source
BLEU	✓		✓	
BLEURT		✓	✓	
COMET		✓	✓	✓
COMET-QE		✓		✓

Table 3: Quality metrics used in our experiments

gram matching between the generated sentences and their reference sentences.BLEURT (Pu et al., 2021), COMET (Rei et al., 2020), and COMET-QE (Chimoto and Bassett, 2022) utilize embeddings from language models to focus on semantic meanings.BLEURT evaluates the generated sentences against reference sentences, while COMET also considers both source sentences and reference sentences.In contrast, COMET-QE directly assesses the similarity between the source and generated sentences, thus avoiding the ambiguity that may arise from using references. For latency evaluation, we choose Average Lagging (AL) (Ma et al., 2019), Length Adaptive Average Lagging (LAAL) (Papi et al., 2022), and Average Token Delay (ATD) (Kano et al., 2023)[14].

6 Experimental Results on Speech-to-Text

Evaluation 1: tst-COMMON Figure 3 shows the results of speech-to-text experiments. When we focused on BLEU-AL in Figure 3 for $k = 1$, $k = 3$, and $k = 5$, the LLM-SI-Corpus (GPT-3.5 and GPT-4) achieved higher BLEU scores than OFFLINE, indicating improvements in both latency and quality. However, as the value of k increases, the BLEU score in Pretrain starts to surpass that of LLM-SI-Corpus and OFFLINE when exceeds around $k = 9$. This pattern persists across LAAL and ATD as well. This is attributed to the alignment of training and evaluation data, leading to enhanced BLEU scores. Next, in {BLEURT, COMET}–{AL, LAAL}, both quality and latency in LLM-SI-Corpus (GPT-3.5 and GPT-4) surpasses OFFLINE and Pretrain. Also, in COMET-QE, the LLM-SI-Corpus demonstrates superior quality and latency performance at all latencies in AL, LAAL, and ATD, indicating that the model trained on the LLM-SI-Corpus can perform high-quality translations with low latency. Despite the trends observed in text-to-text settings, the quality gap remains evident in speech-to-text settings even as k increases.

[14]We cover all evaluation metrics used in the shared task of IWSLT 2024: https://iwslt.org/2024/simultaneous.

Evaluation 2: SIC-test Figure 4 shows the result of SIC-test. Focus on BLEU-AL, the result indicates that the LLM-SI-Corpus exhibits higher quality than OFFLINE up to around $k = 5$. However, OFFLINE and SIC perform better as k increases because these align with the training and evaluation data, thereby improving the BLEU score. The same trends are observed in LAAL and ATD. Next, in {BLEURT, COMET}–{AL, LAAL, ATD}, both quality and latency in LLM-SI-Corpus (GPT-3.5 and GPT-4) surpasses OFFLINE and Pretrain. The same as in COMET-QE, the LLM-SI-Corpus outperforms OFFLINE and Pretrain at all latencies in AL, LAAL, and ATD, indicating that the model trained on the LLM-SI-Corpus can perform high-quality translations with low latency.

Evaluation 3: Chunk-wise Figure 5 shows that the LLM-SI-Corpus consistently exhibits superior quality and latency performance across all quality evaluation metrics. The quality gap among models is noticeable, particularly when wait-k is small, and remains significant even as wait-k values increase. GPT-4 achieves a better balance between quality and latency than GPT-3.5, likely due to its higher model capabilities. OFFLINE achieved comparable results on both tst-COMMON and SIC-test, however, in this test set, the results were weaker, indicating that OFFLINE has difficulty achieving more monotonic translation.

Summary The results indicate that the LLM-SI-Corpus delivers better translation quality with minimal latencies across all semantic similarity-focused evaluation metrics. Even in BLEU, the LLM-SI-Corpus achieves equivalent translation quality, especially when k is small. In the SIC fine-tuned model on the ATD evaluation setting, we observed significantly longer lags compared to other fine-tuned models. This trend is also observed in Ko et al. (2023). This observation may be attributed to the fact that some transcripts in SIC are extremely short relative to the source length. Fine-tuning with such data may lead to undesired generation results, such as excessive repetition (Table 12 in Appendix E), leading to longer lags. While achieving a shorter output length is advantageous in the ATD setting, this evaluation metric may overemphasize a shorter output, which could be unfair, as shorter outputs may omit important content from the source.Outputs that are excessively shortened or lengthened should be penalized, and we leave this for future work.

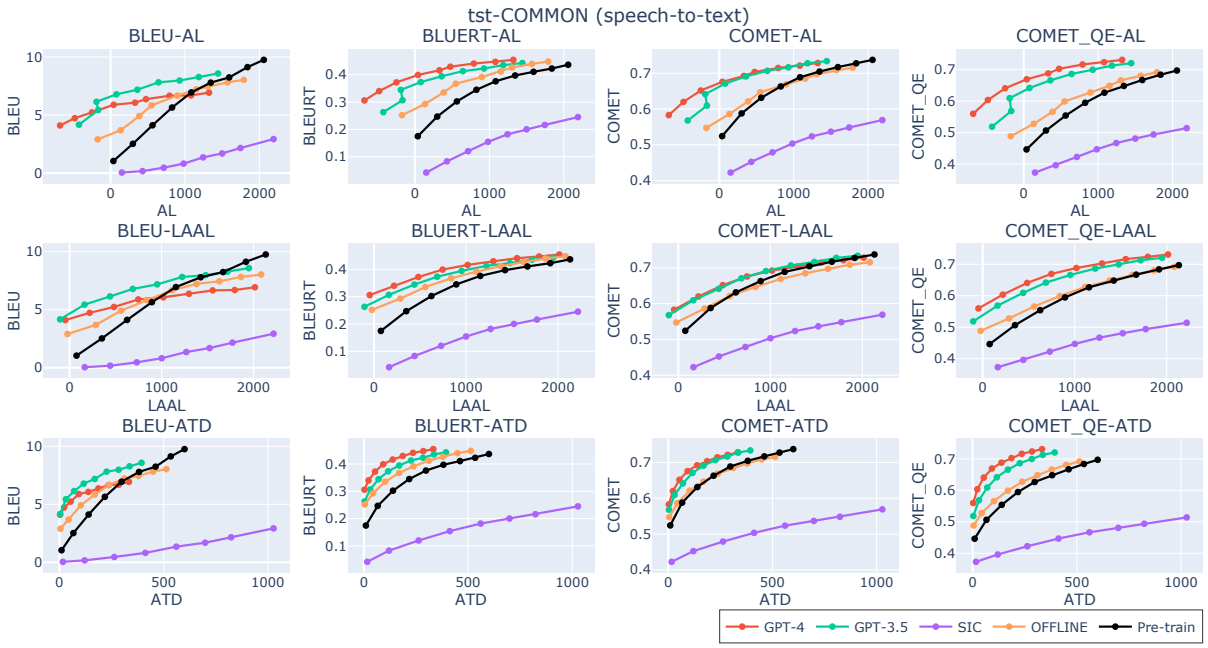

Figure 3: The results of tst-COMMON on speech-to-text settings. Each plot, from left to right, represents wait-k values ranging from $1, 3, 5, 7, 9, 11, 13, 15, 17$.

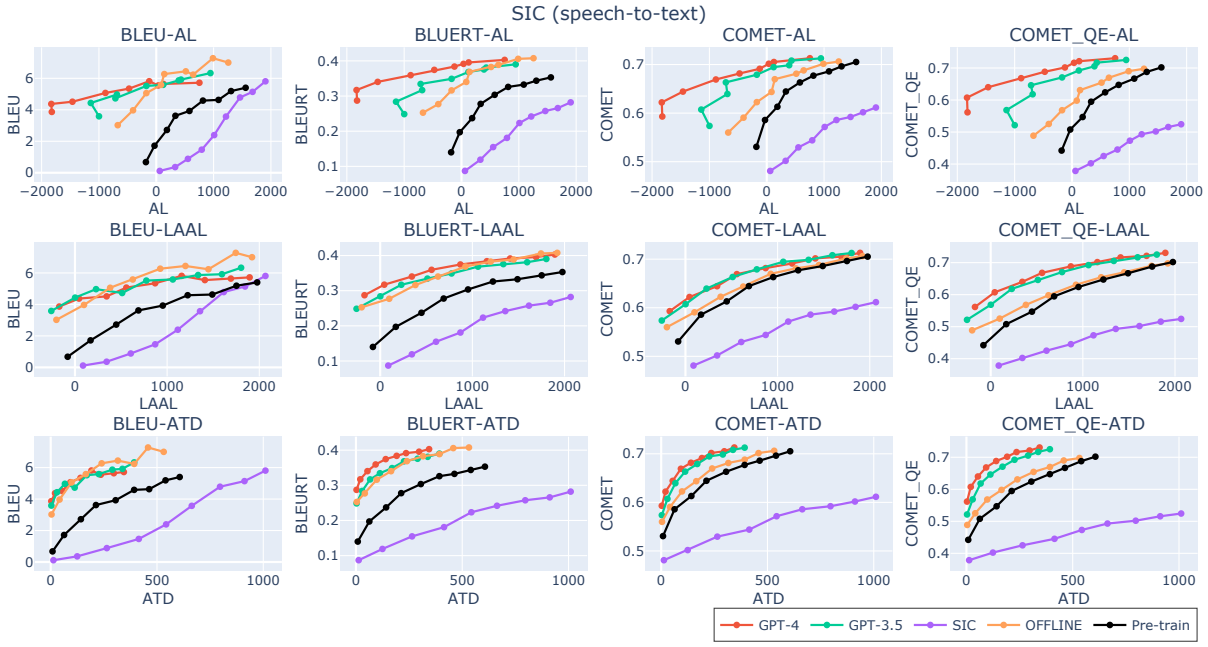

Figure 4: The results of SIC-test on speech-to-text settings. Each plot, from left to right, represents wait-k values ranging from $1, 3, 5, 7, 9, 11, 13, 15, 17$.

Qualitative Analysis Table 4 shows the quality gap among different models when evaluating tst-COMMON with $k = 7$. GPT-4 produces the longest output, retaining most of the information from the source while preserving the original word order, whereas GPT-3.5 translates only (1) and (2), omitting the rest. Other models, fine-tuned with OFFLINE, SIC, and Pretrain, performed significantly worse, translating only *(1) 'Here was some lawyer or money manager who,* while the rest was omitted. In such cases when the output length is short, ATD, which is a latency metrics that account for both the start and end timing of the translation, may favor shorter outputs. However, outputs that are too short compared to the source often result in missing information. While it is important to con-

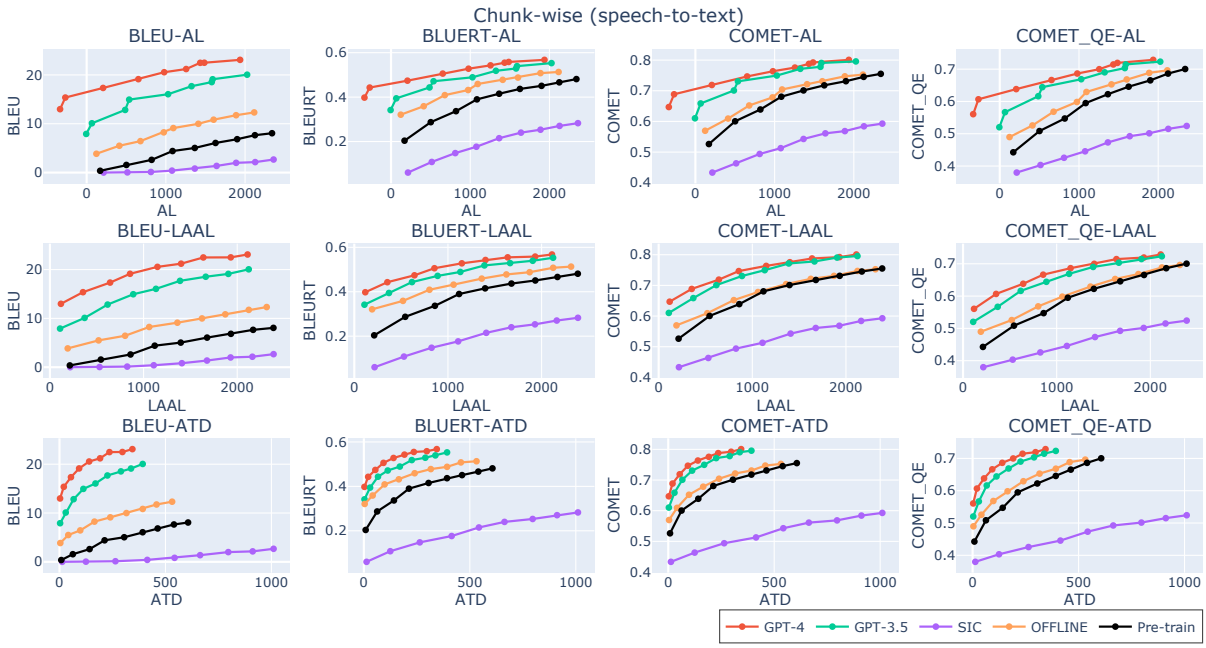

Figure 5: The results of Chunk-wise on speech-to-text settings. Each plot, from left to right, represents wait-k values ranging from $1, 3, 5, 7, 9, 11, 13, 15, 17$.

Source	(1) Here was some lawyer or money manager who, / (2) for the rest of his life, / (3) gets to tell people / (4) that he went into a burning building / (5) to save a living creature, / (6) just because he beat me by five seconds.
Reference	(1) 弁護士だったか資産運用者だったか (*some lawyer or money manager*) / (2) 彼は後々まで (*for the rest of his life*) / (3) 言い続けるでしょう (*gets to tell people*) / (4) 自分は燃え盛る建物の中に入り (*he went into a burning building*) / (5) 生き物を救ったのだと (*to save a living creature*) / (6) 私より５秒前に着いただけなのに (*just because he beat me by five seconds*)。
Pretrain	(1) ここには弁護士やお金持ちの誰かがいました (*here was some lawyer or money manager*)
SIC	(1) 弁護士やマネーマンが (*some lawyer or money manager*)。
OFFLINE	(1) ここには弁護士やマネージャーがいます (*here was some lawyer or money manager*)。
GPT-3.5	(1) ここには弁護士やマネージャーがいました (*here was some lawyer or money manager*) / (2) 残りの人生を過ごした (*spend for the rest of his life*)。
GPT-4	(1) ここには、いくつかの弁護士またはマネージャーがいました (*here was some lawyer or money manager*)。/ (2) 彼は彼の生涯の残りの間 (*for the rest of his life*)、/ (3) 人々に伝え続けました (*kept telling people*)。/ (4) 彼が燃える建物に入ったと (*he went into a burning building*)、/ (5) 生きている生き物を救うために (*to save a living creature*)。

Table 4: Example of output sentences in Pretrain, SIC, OFFLINE, GPT-3.5, and GPT-4 on tst-COMMON in wait$k$=7 on Speech-to-Text setting. From (3) to (6) is omitted in GPT-3.5, while most information is maintained in GPT-4.

sider both the start and end timing of translations in simultaneous settings, since overly long translations can delay the timing of the next sentence, it's equally important to maintain a balance between the source and target lengths because too short target outputs compared to the source may not necessarily reflect good translation quality. Additional examples and their analysis for both speech-to-text and text-to-text settings are in in Appendix D.

7 Discussions

We picked several important discussion themes, with further discussions provided in Appendix E.

7.1 Is the CWMT guideline effective for SI?

Based on our observations of chunk-wise, the test data following the CWMT achieves chunk order synchronization without omissions. This alignment, free of omissions, fits well with existing

Source	(1) A few weeks later, / (2) the department / (3) received a letter / (4) from the homeowner / (5) thanking us / (6) for the valiant effort displayed in saving her home.
Reference	(1) 数週間後 （*several weeks later*） / (2) 消防団は (*the fire brigade is*) / (4) 家主から (*from the landlord*) / (6) 火事の際の勇敢な活動に対する (*for bravery in the event of a fire*) / (5) お礼の (*thank you*) / (3) 手紙をもらいました (*I got a letter*)。
Pretrain	(1) 数週間後 (*several weeks later*) / (2) 政府は (*the government*) / (3) 手紙を送りました (*I sent a letter*))。
SIC	(1) 数週間後 (*several weeks later*)、
Offline	(1) 数週間後 (*several weeks later*)、 / (2) 政府は、 (*the government*) / (3) 手紙を送りました (*I sent a letter*)。
GPT-3.5	(1) 数週間後、 (*several weeks later*) / (2) その部門は (*the department*) / (3) 手紙を受け取った(*I got a letter*)。/ (4) 自宅のオーナーから (*from the home owner*)、/ (5) 私たちに感謝 (*thank us*) / (3) の手紙を(*letter*、 / (6) 安全を確保するために彼女の家を救うために示された勇敢な努力に感謝する。(*thanking her for the valiant efforts shown to save her home to ensure its safety*)
GPT-4	(1) 数週間後(several weeks later)、/ (2) その部門が(*the department*) / (3) 手紙を(*a letter*) / (4) 自宅から所有者から(*from home to owner*) / (3) 受け取った(*received*)。/ (5) それは、私たちに感謝の意を表すもので、(*it is our way of saying thank you*) / (6) 彼女の家を救うために勇敢な努力がなされた(*a valiant effort was made to save her home*)。

Table 5: Examples of the generated texts for k = 7 in speech-to-text settings on tst-COMMON. The bracketed numbers indicate the corresponding phrases in the source text.

machine translation evaluation metrics, which prioritize precise content correspondence between the source and target texts. However, such test data does not account for other SI strategies, such as summarization or deletion, a key technique for reducing latency in SI. Additionally, the strict focus on chunk order alignment can result in unnatural or redundant translations. Therefore, creating an SI corpus that incorporates strategies like summarization remains a critical challenge for future work.

7.2 Which is better GPT-4 vs. GPT-3.5?

Both GPT-3.5 and GPT-4 demonstrate equivalent proficiency in preserving word order, indicating a similar ability to understand prompts. If the primary goal is to maintain word order simply, GPT-3.5 is sufficient. However, for those prioritizing output quality, GPT-4 may offer better performance, as shown in Table 5. While both GPT-3.5 and GPT-4 generally maintain the source word order, GPT-4 occasionally reorders words for improved naturalness, which is acceptable. In contrast, GPT-3.5 is more consistent with maintaining the original word order but lacks fluency. Further details are provided in Appendix D. Additionally, the results in Section 6 show that GPT-3.5 surpasses GPT-4 in some BLEU scores, indicating that metrics focused solely on textual similarity cannot capture the trade-off between naturalness and word order. This highlights the need for new evaluation metrics.

Overall, the models fine-tuned with LLM-SI-Corpus outperform those fine-tuned with the other kinds of data. These results suggest that LLMs with a sufficient level of instruction-following capability are effective for constructing corpora to train models better suited for simultaneous settings. Additional discussions are provided in the Appendix. E.

8 Conclusion and Future Directions

In this study, we proposed a method for converting ST corpora to SI corpora using LLMs to improve the monotonicity yet maintain the quality. This corpus creation method follows the CWMT guidelines, focusing on the English-to-Japanese direction.

To evaluate the effectiveness of our LLM-SI-Corpus, we conducted experiments in three scenarios: a general offline ST corpus (tstCOMMON), an SI corpus (SIC-test), and a CWMT test corpus (Chunk-wise), in both speech-to-text and text-to-text settings. In all cases, the SiMT models fine-tuning with the LLM-SI-Corpus outperformed others, achieving lower latency and higher quality. Moreover, while manually constructing SI corpora is costly, the LLM-SI-Corpus can be produced for only 20 dollars. Therefore, it can be easily applied to other ST corpora or adapted to other languages since it utilizes LLMs.

For future work, we plan to explore the application of other SI techniques, such as summarization, extend these methods to larger-scale ST corpora, and expand their use to speech-to-speech settings.

9 Limitations

Lack of SiMT evaluation data, methods, and definitions The existing metrics for evaluating SiMT systems present challenges in reducing latency due to their reliance on ST test data, such as tst-COMMON, despite the diverse techniques involved in SI. This reliance on ST data for evaluation is a major limitation of this work. Therefore, there is an urgent need to establish evaluation metrics and data tailored to SiMT. Furthermore, although various SI techniques are available, there has been no thorough discussion from an engineering perspective on which techniques are essential for SiMT. Addressing this gap will be a key focus of our future work. These issues were highlighted through our comprehensive experiments and analysis.

Expanding SI Corpora In this study, we constructed the LLM-SI-Corpus based on the NAIST-SI-Aligned-ST corpus for comparison with existing SI corpora. Our method is cost-effective and applicable to various other ST corpora. Additionally, we demonstrated that LLM outputs are effective for developing SiMT corpora, and we plan to explore their applicability to other SiMT methods, such as handling omissions, in future work. We hope that expanding into multiple languages and enhancing data augmentation will contribute to further advancements in the SiMT field.

Dataset Quality In this study, we used GPT-3.5 and GPT-4 with a simple prompt for data creation. Therefore, there is room for improvement in the selection of LLMs and the refinement of prompts. Thus, it may become possible to create higher quality datasets at a lower cost when the API prices decrease or by switching to other strong LMs such as Gemini (Team et al., 2024), Claude 3 and Qwen (Bai et al., 2023). Additionally, employing prompt strategies that leverage the capabilities of LMs, such as Chain of Thought (CoT) (Wei et al., 2022), Tree of Thought (ToT) (Yao et al., 2023a) and ReAct (Yao et al., 2023b), could potentially lead to the production of higher quality datasets.

Other SI techniques In this study, we addressed CWMT, focusing on chunking within SI techniques. However, there are many other SI techniques (Camayd-Freixas, 2011; Okamura and Yamada, 2023), such as omission and summarization, and addressing these is also necessary to achieve better SI. Furthermore, the evaluation methods for these techniques are still in development and have not yet been fully established, making them a critical focus for SiMT research. While LLMs demonstrate prompt understanding based on CWMT by making translations more monotonic, the next step is to investigate whether they can identify less important words that can be omitted from a technical SI standpoint. Additionally, assessing their ability to perform balanced omission and summarization based on syllable counts to achieve low latency and high quality will be an important challenge to explore in future work.

10 Ethical Considerations

License of Source Dataset The NAIST-SIC-Aligned-ST corpus used in this study is available only for research purposes. We have used this corpus for research, so there are no license violations. Moreover, the LLM-SI-Corpus was created from the NAIST-SIC-Aligned-ST corpus and thus inherits its terms of use[15]. In terms of distribution, redistribution of interpretation transcripts is prohibited; therefore, we release only our transcripts and the corresponding audio segment information and do not contain any audio data or the original transcripts. Furthermore, the README file of the LLM-SI-Corpus clearly states the source of the data, the license, and acknowledgments, and properly documents the original data information. Note that, it is permitted to cite example sentences from the NAIST-SIC-Aligned-ST corpus.

Ownership rights about outputs of the LLMs The LLM-SI-Corpus was created using GPT-3.5 and GPT-4 and is therefore subject to OpenAI's license terms[16]. OpenAI assigns to us all rights, titles, and interests in and to the output. As a result, we retain the ownership rights. There are no restrictions on distributing the datasets, but in line with NAIST-SIC-Aligned-ST, we distribute only for research purposes. However, these terms may change, and there may be a need to impose distribution restrictions depending on the terms.

Moderations Since the LLM-SI-Corpus fundamentally originates from TED Talks, it does not contain any potentially harmful information. Furthermore, we checked using OpenAI Moderation APIs[17] and found no examples of harmful content.

[15] https://dsc-nlp.naist.jp/data/NAIST-SIC/Aligned-ST/

[16] https://openai.com/policies/terms-of-use

[17] https://platform.openai.com/docs/guides/moderation

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

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

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: a framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

Silvia Bernardini, Adriano Ferraresi, and Maja Miličević. 2016. From EPIC to EPTIC – exploring simplification in interpreting and translation from an intermodal perspective. *Target. International Journal of Translation Studies*, 28(1):61–86.

Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2024. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4.

Zhongxi Cai, Koichiro Ryu, and Shigeki Matsubara. 2020. What affects the word order of target language in simultaneous interpretation. In *2020 International Conference on Asian Language Processing (IALP)*, pages 135–140.

Erik Camayd-Freixas. 2011. Cognitive theory of simultaneous interpreting and training. In *Proceedings of AMTA*.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Must-c: A multilingual corpus for end-to-end speech translation. *Computer Speech & Language*, 66:101155.

Junkun Chen, Renjie Zheng, Atsuhito Kita, Mingbo Ma, and Liang Huang. 2021. Improving simultaneous translation by incorporating pseudo-references with fewer reorderings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5857–5864, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Everlyn Chimoto and Bruce Bassett. 2022. COMET-QE and active learning for low-resource machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4735–4740, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Hexuan Deng, Liang Ding, Xuebo Liu, Meishan Zhang, Dacheng Tao, and Min Zhang. 2023. Improving simultaneous machine translation with monolingual data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):12728–12736.

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.

Kosuke Doi, Yuka Ko, Mana Makinae, Katsuhito Sudoh, and Satoshi Nakamura. 2024. Word order in English-Japanese simultaneous interpretation: Analyses and evaluation using chunk-wise monotonic translation. In *Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024)*, pages 254–264, Bangkok, Thailand (in-person and online). Association for Computational Linguistics.

Kosuke Doi, Katsuhito Sudoh, and Satoshi Nakamura. 2021. Large-scale English-Japanese simultaneous interpretation corpus: Construction and analyses with sentence-aligned data. In *Proceedings of the 18th*

International Conference on Spoken Language Translation (IWSLT 2021), pages 226–235, Bangkok, Thailand (online). Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Maha Elbayad, Laurent Besacier, and Jakob Verbeek. 2020. Efficient Wait-k Models for Simultaneous Machine Translation. In *Proc. Interspeech 2020*, pages 1461–1465.

Ryo Fukuda, Kosuke Doi, Katsuhito Sudoh, and Satoshi Nakamura. 2024. Test data creation in simultaneous machine translation in english to japanese pair: Insights from simultaneous interpretation tactics. *IPSJ SIG Technical Report*. (In Japanese).

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.

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

Shoutao Guo, Shaolei Zhang, and Yang Feng. 2023. Simultaneous machine translation with tailored reference. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3070–3084, Singapore. Association for Computational Linguistics.

HyoJung Han, Seokchan Ahn, Yoonjung Choi, Insoo Chung, Sangha Kim, and Kyunghyun Cho. 2021. Monotonic simultaneous translation with chunk-wise reordering and refinement. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1110–1123, Online. Association for Computational Linguistics.

He He, Alvin Grissom II, John Morgan, Jordan Boyd-Graber, and Hal Daumé III. 2015. Syntax-based rewriting for simultaneous machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 55–64, Lisbon, Portugal. 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. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:3451–3460.

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. Libri-light: 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. 2023. Average Token Delay: A Latency Metric for Simultaneous Translation. In *Proc. INTERSPEECH 2023*, pages 4469–4473.

Eugene Kharitonov, Morgane Rivière, Gabriel Synnaeve, Lior Wolf, Pierre-Emmanuel Mazaré, Matthijs Douze, and Emmanuel Dupoux. 2021. Data augmenting contrastive learning of speech representations in the time domain. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 215–222.

Yuka Ko, Ryo Fukuda, Yuta Nishikawa, Yasumasa Kano, Katsuhito Sudoh, and Satoshi Nakamura. 2023. Tagged end-to-end simultaneous speech translation training using simultaneous interpretation data. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 363–375, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Kerstin Kunz, Christoph Stoll, and Eva Klüber. 2021. HeiCiC: A simultaneous interpreting corpus combining product and pre-process data. In *Proceedings for the First Workshop on Modelling Translation: Translatology in the Digital Age*, pages 8–14, online. Association for Computational Linguistics.

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.

Minh-Thang Luong and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 76–79, Da Nang, Vietnam.

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. 2020. 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.

Dominik Macháček, Matúš Žilinec, and Ondřej Bojar. 2021. Lost in Interpreting: Speech Translation from Source or Interpreter? In *Proc. Interspeech 2021*, pages 2376–2380.

Akira Mizuno. 2016. Simultaneous interpreting and cognitive constraints.

Makoto Morishita, Katsuki Chousa, Jun Suzuki, and Masaaki Nagata. 2022. JParaCrawl v3.0: A large-scale English-Japanese parallel corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6704–6710, Marseille, France. European Language Resources Association.

Yasmin Moslem, Gianfranco Romani, Mahdi Molaei, John D. Kelleher, Rejwanul Haque, and Andy Way. 2023. Domain terminology integration into machine translation: Leveraging large language models. In *Proceedings of the Eighth Conference on Machine Translation*, pages 902–911, Singapore. Association for Computational Linguistics.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Yuki Okamura and Masaru Yamada. 2023. Jyun okuri yaku" no kihan to mohan doji tsuyaku wo mohan tosita kyoikuron no shiron (). In Hiroyuki Ishizuka, editor, *Word Order in English-Japanese Interpreting and Translation: The History, Theory and Practice of Progressive Translation*, pages 217–250. Hitsuji Syobo.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav

Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Jun Pan. 2019. The Chinese/English political interpreting corpus (CEPIC): A new electronic resource for translators and interpreters. In *Proceedings of the Human-Informed Translation and Interpreting Technology Workshop (HiT-IT 2019)*, pages 82–88, Varna, Bulgaria. Incoma Ltd., Shoumen, Bulgaria.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. 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, Matteo Negri, and Marco Turchi. 2023a. Attention as a guide for simultaneous speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13340–13356, Toronto, Canada. Association for Computational Linguistics.

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

Matthias Paulik and Alex Waibel. 2009. Automatic translation from parallel speech: Simultaneous interpretation as mt training data. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 496–501.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. 2018. JESC: Japanese-English subtitle corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Heike Przybyl, Ekaterina Lapshinova-Koltunski, Katrin Menzel, Stefan Fischer, and Elke Teich. 2022. EPIC UdS - creation and applications of a simultaneous interpreting corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1193–1200, Marseille, France. European Language Resources Association.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Collection of a simultaneous translation corpus for comparative analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 670–673, Reykjavik, Iceland. European Language Resources Association (ELRA).

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association*

for Computational Linguistics: ACL-IJCNLP 2021, pages 3450–3466, Online. Association for Computational Linguistics.

Gemini Team et al. 2024. Gemini: A family of highly capable multimodal models.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Hitomi Toyama, Shigeki Matsubara, Koichiro Ryu, Nobuo Kawaguchi, and Yasuyoshi Inagaki. 2004. Ciair simultaneous interpretation corpus. In *Proceedings of Oriental COCOSDA*.

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.

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

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online. Association for Computational Linguistics.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020. Fairseq S2T: Fast speech-to-text modeling with fairseq. 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: System Demonstrations*, pages 33–39, Suzhou, China. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Ruiqing Zhang, Xiyang Wang, Chuanqiang Zhang, Zhongjun He, Hua Wu, Zhi Li, Haifeng Wang, Ying Chen, and Qinfei Li. 2021. BSTC: A large-scale Chinese-English speech translation dataset. In *Proceedings of the Second Workshop on Automatic Simultaneous Translation*, pages 28–35, Online. Association for Computational Linguistics.

Shaolei Zhang and Yang Feng. 2021. Universal simultaneous machine translation with mixture-of-experts wait-k policy. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7306–7317, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Jinming Zhao, Philip Arthur, Gholamreza Haffari, Trevor Cohn, and Ehsan Shareghi. 2021. It is not as good as you think! evaluating simultaneous machine translation on interpretation data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6707–6715, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jinming Zhao, Yuka Ko, Kosuke Doi, Ryo Fukuda, Katsuhito Sudoh, and Satoshi Nakamura. 2024. Naist-sic-aligned: an aligned english-japanese simultaneous interpretation corpus.

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

Jiawei Zheng, Hanghai Hong, Xiaoli Wang, Jingsong Su, Yonggui Liang, and Shikai Wu. 2024. Fine-tuning large language models for domain-specific machine translation.

A Detail of the CWMT Guideline and Workflow

Okamura and Yamada (2023) defines these chunk boundaries using the following rules (rule 1, 2, 3, and 4), then Fukuda et al. (2024) added the fifth rule as follows:

1. Before conjunctions or relative pronouns that introduce clauses (excluding when they modify the subject).

2. After infinitives, prepositions, or gerunds when followed by three or more words.

3. When the subject consists of three or more words.

4. Before and after punctuation marks such as commas (excluding lists of individual words), semicolons, hyphens, etc.

5. After prepositional phrases or adverbial phrases at the beginning of a sentence (or directly after conjunctions or relative pronouns that introduce clauses).

Based on these guidelines, Fukuda et al. (2024) defines its chunking workflow. First, rules 1, 3, 4, and 5 are applied to each source sentence chunk, and then the translated chunks are concatenated while preserving boundaries. Rule 2 is optionally applied in the last step to avoid the influence of the prior steps causing extremely small chunk translations. This chunk-wise approach enables interpreters to navigate the challenges posed by grammatical differences between the source and target languages while managing the demands for translation speed and accuracy.

Based on this chunking workflow and CWMT guideline, Fukuda et al. (2024) constructed a test dataset, and its fluency and adequacy were evaluated by a professional interpreter. The procedure is as follows:

1. Translate each chunk from the beginning of the sentence.

2. Translate in a way that the connection between chunks is natural when considering the entire sentence.

3. Translate without including information from the following chunks.

4. Additionally, for the sake of maintaining the fluency of the sentence, the following operations are permitted, but applied carefully:

   (a) Repeating the information from the previous chunk.

   (b) Deferring the information to be translated to the following chunk.

   (c) Omitting unnecessary information.

The CWMT-like test dataset proposed by Fukuda et al. (2024) has been validated and analyzed by Doi et al. (2024) confirming its effectiveness.

B Style differences among SI, Offline Translation and CWMT (Details)

There are significant style gaps among SI, offline translation, and CWMT as described in Fukuda et al. (2024); Ko et al. (2023). Table 6 and Table 7 are examples describing their differences.

C Experiments (Details)

Speech-to-Text Settings Following the settings of Fukuda et al. (2023); Ko et al. (2023), we employ pretrained language models for both encoder and decoder[18] by integrating them into the Transformer architecture (Vaswani et al., 2017).We used Hubert-Large (Hsu et al., 2021) as the encoder, which includes a feature extractor and transformer encoder layers. The feature extractor, trained on 60k hours of unlabeled speech data from Libri-Light (Kahn et al., 2020), consists of a 7-layer convolutional network with kernel sizes of (10,3,3,3,3,2,2), strides of (5,2,2,2,2,2,2), and 512 channels. For the decoder side, we use the decoder parts of mBART50 (Tang et al., 2021), an encoder-decoder model pretrained with 50 language pairs. The decoder consists of 12 layers of transformer decoders, and the embedding layer and linear projection weights are shared, with a vocabulary size of 250K. The inputs are waveforms with a 16kHz sampling rate that are normalized to zero mean and unit variance. During training, each source audio is augmented (Kharitonov et al., 2021) with a probability of 0.8. We train the model with MuST-C v2.0 (Cattoni et al., 2021) as continuous pretraining. We fine-tuned the models for 3K steps, evaluating their performance every 200 steps, and terminated the fine-tuning if there was no improvement in the loss score for eight consecutive evaluations. To avoid overfitting to the small SI data, the following parameters are fixed (Tsiamas et al., 2022): the feature extractor and feed-forward layers of the encoder and the em-

[18]Our baselines are almost the same as the baseline of IWSLT2023 Speech-to-Text settings (https://github.com/facebookresearch/fairseq/tree/iwslt2023/examples/simultaneous_translation), but, due to an implementation issue, we have switched the encoder from wav2vec 2.0 (Baevski et al., 2020) to HuBERT (Hsu et al., 2021).

Source	And (1) I'm / (2) not here to / (3) say that / (4) men are to / (5) blame for the / (6) crisis and what / (7) happened in my / (8) country.
OFFLINE	しかしこの経済 (*but this economy*) / (6) 危機や私の (*crisis and what*) / (8) 国での (*country*) / (7) 出来事について (*happened in my*) / (1) 私は (*I'm*) / (4) 男性に (*men are to*) / (5) 非があると (*blame for the*) / (3) 言うつもりは (*say that*) / (2) ありません (*not here to*)。
SI	(4)男性の (*men are to*)、 / (5) せいだけでは (*blame for the*) / (2) ありません、私どもの (*not here to*) / (8) 国の、金融 (*country*) / (6) 崩壊の (*crisis and what*)、 / (5) 責任は (*blame for the*)。

Table 6: Translation style difference between offline and SI. The number indicates the corresponding words in the source. The example is coming from (Ko et al., 2023).

Source	(1) Groups like Anonymous / (2) have risen up / (3) over the last 12 months / (4) and have become a major player / (5) in the field of online attacks.
OFFLINE	(1) Anonymous というグループは (*Groups like Anonymous*) / (3) この12ヶ月ほど (*over the last 12 months*) / (2) 活気づいていて (*have risen up*) / (5) オンライン攻撃において (*in the field of online attacks*) / (4) 大きな存在になってます (*and have become a major player*)。
CWMT	(1) アノニマスのようなグループが (*Groups like Anonymous*) / (2) 台頭してきています (*have risen up*)、 / (3) 過去12ヶ月にわたって (*over the last 12 months*)、 / (4) そして主要なプレイヤーになっています (*and have become a major player*)、 / (5) オンライン攻撃の分野において (*in the field of online attacks*)。

Table 7: Translation style difference between offline and CWMT. The number indicates the corresponding words in the source. The example is coming from (Fukuda et al., 2024).

bedding, self-attention, and feed-forward layers of the decoder.

Text-to-Text Settings We train an NMT model through pretraining[19], then fine-tuned it using SI data. For pretraining, we used WMT21 En-Ja datasets (Akhbardeh et al., 2021) (JParaCrawl v3 (Morishita et al., 2022), News Commentary v16 (Tiedemann, 2012), WikiTitles v3 (Tiedemann, 2012), WikiMatrix v1 (Schwenk et al., 2021), JESC (Pryzant et al., 2018), KFTT (Neubig, 2011)) and MuST-C v2.0 (Cattoni et al., 2021). We use SentencePiece (Kudo and Richardson, 2018) for subword tokenization with a Unigram Language Model (Kudo, 2018). The vocabulary size is 32K tokens with a character coverage of 0.99995 on a shared dictionary. The tokenizer was trained on the pretraining data. We use a Transformer-big model (Vaswani et al., 2017), warmup update at 4000, dropout at 0.3, and the learning rate at 0.0005. The model is trained for 100K steps, with evaluation conducted every 2K steps. Training is terminated if there is no improvement in the best loss after eight consecutive evaluations. During fine-tuning, we trained for 3K steps, with evaluations conducted every 200 steps. Fine-tuning is also finished if there are no updates after eight consecutive evaluations. The evaluation metrics and test datasets are the same as those described in Section 5.

[19]Our baselines are based on the English-to-Japanese Text-to-Text translation at IWSLT2022 settings: https://github.com/ksudoh/IWSLT2022_simul_t2t_baseline_enja

C.1 Results on Text-to-Text Setting

Evaluation 1: tst-COMMON Figure 6 shows the result of tst-COMMON in text-to-text settings. Focusing on k=1 and k=3 in BLEU, the LLM-SI-Corpus (GPT-3.5 and GPT-4) achieves higher BLEU scores with lower latency than OFFLINE. However, as the value of k increases, the BLEU scores for GPT-3.5 and GPT-4 begin to stagnate compared to the Pretrained and OFFLINE models. In {BLEURT, COMET}, the quality of the LLM-Corpus surpasses that of OFFLINE when k is less than 5, after which the quality of all three models becomes similar. Additionally, compared to the Pretrained model, the translation quality of the LLM-Corpus remains superior at all latency levels. In COMET-QE, which focuses on semantic similarity between the source and generated text directly, the LLM-SI-Corpus outperforms OFFLINE when k is up to around 9, indicating that models fine-tuned with the LLM-SI-Corpus can achieve high-quality translations with relatively low latency.

On the other hand, the results from SIC show lower quality at all k values, despite demonstrating an advantage in latency, particularly achieving the lowest latency in ATD. The reason SIC achieves the lowest latency may be due to its shorter outputs, as shown in Table 8. This could be attributed to omis-

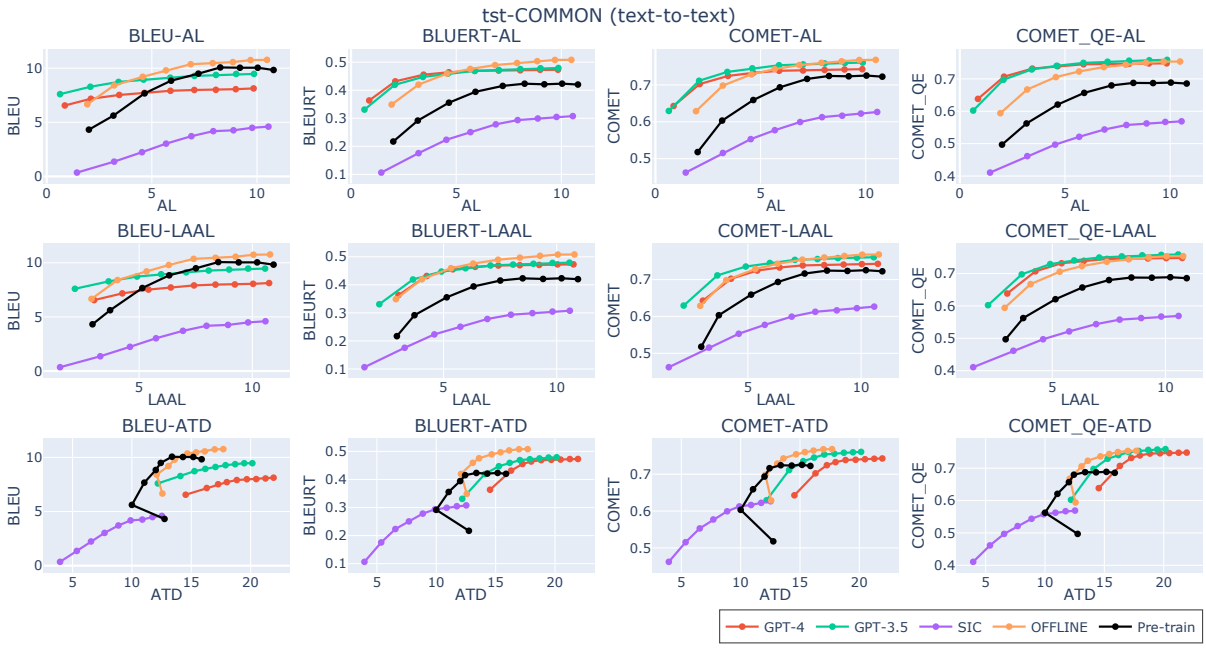

Figure 6: The results of tst-COMMON on text-to-text settings. Each plot, from left to right, represents wait-k values ranging from $1, 3, 5, 7, 9, 11, 13, 15, 17$.

sions and other factors in the SIC corpus[20], which lead to shorter outputs compared to the source length, resulting in the lowest quality but the smallest latency among the models.

Evaluation 2: SIC-test Figure 7 shows the result of SIC-test in text-to-text settings, in which we highlight BLEU-AL, where the LLM SI-Corpus exhibits higher quality than OFFLINE up to about k=5. The same trend is observed in LAAL. However, SIC performs better at high latency because it aligns the training and evaluation data at the sentence level, thereby improving the BLEU score. In contrast, the LLM-SI-Corpus demonstrates higher quality than SIC at low latencies. Conversely, when focusing on ATD, SIC shows the best results in both latency and quality, suggesting that the shorter output sentences are attributed to omissions and truncations. Meanwhile, when focusing on {BLEURT, COMET, COMET-QE}, SIC exhibits the worst translation quality. This is likely due to the effects of omissions, where missing information from the source text leads to decreased semantic similarity. Conversely, the LLM-SI-Corpus outperforms OFFLINE up to a moderate level of latency, and in terms of COMET-QE, it achieves comparable or better results at all latencies.

[20]This trend has also been reported by Ko et al. (2023).

Evaluation 3: Chunk-wise Additionally, when focusing on {AL, LAAL}, SIC tends to translate slightly faster than any other corpus, but the quality is the lowset, and this was also seen in tst-COMMON. Figure 8 shows the test results of Chunk-wise in text-to-text settings. The LLM-SI-Corpus consistently delivers better translation quality than other models. For latency measuring with ATD, although SIC has a latency advantage, its translation quality is significantly lower. Additionally, when focusing on {AL, LAAL}, SIC tends to translate slightly faster than any other corpus, but the quality is the lowset, and this was also seen in tst-COMMON.

Summary We evaluated the models using three different test datasets. When measuring quality with BLEU, the results vary depending on the characteristics of the test data. If measured using tst-COMMON and SIC-test, the model fine-tuned with OFFLINE performs slightly better than the LLM-SI-Corpus, but the LLM-Corpus outperforms when evaluated with chunk-wise. These variations suggest that BLEU scores are significantly influenced by the translation characteristics of the reference. Moreover, in semantic evaluation metrics using references, such as BLEURT and COMET, the LLM-SI-Corpus achieves comparable or superior translation quality at all latencies. In the reference-free metric COMET-QE, the LLM-SI-Corpus consis-

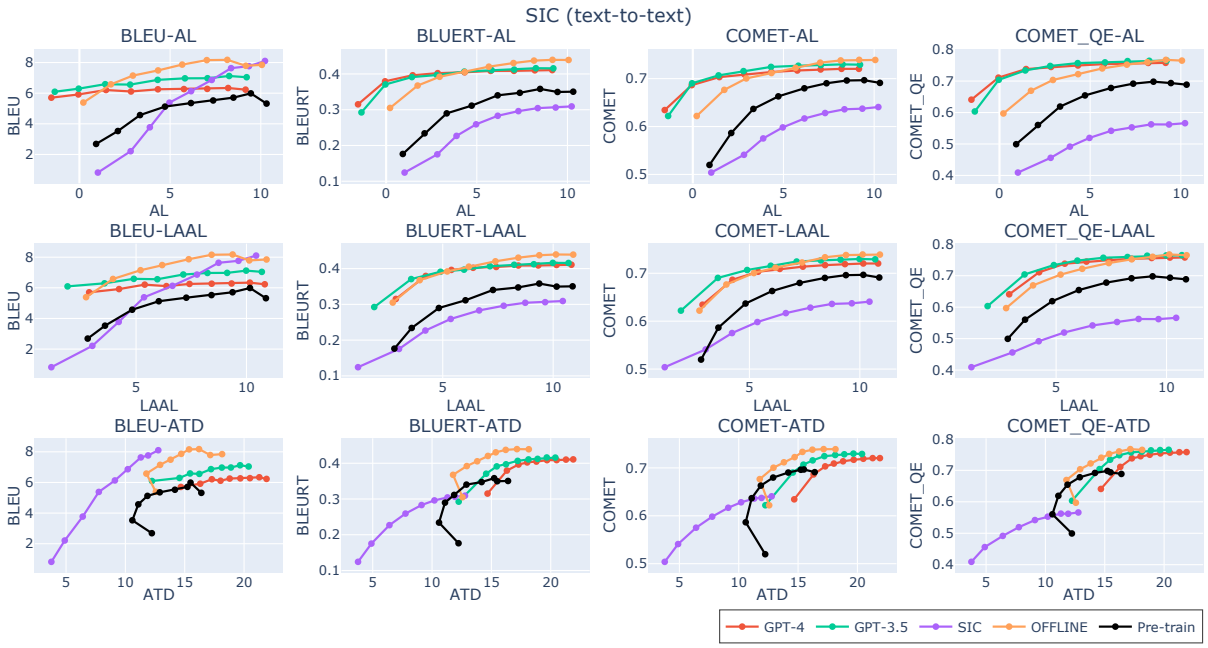

Figure 7: The results of SIC-test on text-to-text settings. Each plot, from left to right, represents wait-k values ranging from $1, 3, 5, 7, 9, 11, 13, 15, 17$.

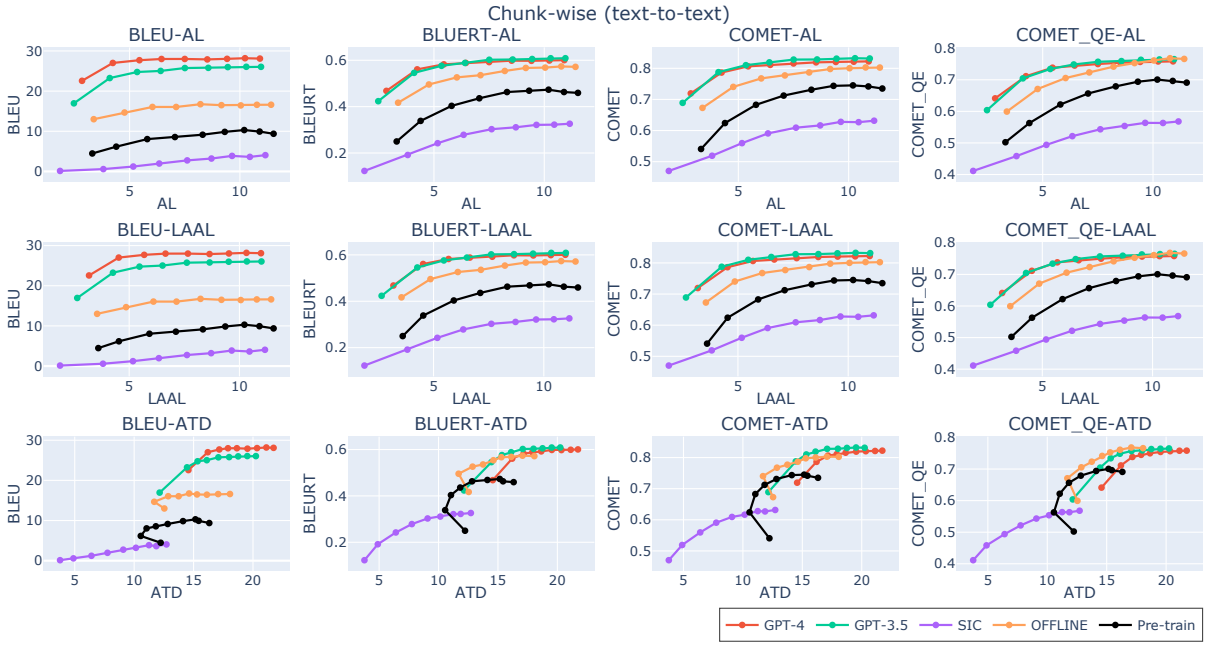

Figure 8: The results of Chunk-wise on text-to-text settings. Each plot, from left to right, represents wait-k values ranging from $1, 3, 5, 7, 9, 11, 13, 15, 17$.

tently demonstrates better quality across all test datasets.

When focusing on ATD to measure latency, the LLM-SI-Corpus tends to produce longer outputs, leading to slightly higher latency. However, this increased latency is necessary to balance quality and latency, as output examples show that the model fine-tuned with LLM-SI-Corpus achieves higher quality compared to other models with lower latency, however such latency is necessary to balance latency and quality, as output examples show that model fine-tuned with LLM-SI-Corpus achieves good quality compared to other models, which achieves small latency. These findings indicate that while achieving low latency is considered preferable in simultaneous settings, excessively small

latency in ATD increases the risk of producing outputs that are too short to fully translate the source content, thereby reducing translation quality.

D Qualitative Analysis

D.1 Text-to-Text setting on tst-COMMON when k=7

Table 13 demonstrates the equivalent quality of GPT-3.5 and GPT-4, with a small reordering between (4) and (5) observed in both models. Table 8 shows that GPT-4, with a small reordering, demonstrates better fluency than GPT-3.5, while both models successfully translate all the content from the source. A small reordering between (2) and (3) appears in GPT-4, whereas GPT-3.5 maintains the exact word order from the source, sacrificing fluency at each chunk boundary Although our motivation in this work is to keep word order in the source, we also consider small reorderings necessary to maintain its fluency. Our focus is on long-distance reordering, such as the complete switch between (1) and (3) observed in the reference, which should be avoided. Such long-distance reordering leads to increased latency because translating (3) in the reference is only possible once (3) in the source becomes available, and the rest can only be translated after (3). Table 9 shows that GPT-4 achieves both fluency and word order, though the output becomes longer. In contrast, GPT-3.5 omits (5), the latter part of the source, indicating that GPT-4 produces better quality compared to GPT-3.5.

D.2 Speech-to-Text setting on tst-COMMON when k=7

In Table 10, both GPT-3.5 and GPT-4 could translate all information in the source but GPT-4 is better at quality and maintains its fluency.

D.3 Summary

From these analyses, we report that while both GPT-3.5 and GPT-4 have the ability to follow the prompt to maintain the word order in the source, GPT-4 could manage the prompt and fluency at the same time better than GPT-3.5 (Table 13, Table 8, Table 10). We also note that the severity of omitting information from the source is more serious in GPT-3.5 than GPT-4 (Table 9, Table 4). We leave the investigation of whether the omission is attributed to the ability gap between GPT-3.5 and GPT-4 for future work.

E Discussions (Details)

E.1 Word Order

We investigate the extent to which the source word order is preserved in the target, focusing on examples generated with a wait-k value of 7 in the text-to-text setting as shown in Table 11. In the source, the phrase order is structured as (1), (2), (3), and (4), whereas in the reference, which comes from the TED Talk subtitles, the order is (1), (4), and (2), with (3) omitted. Both GPT-3.5 and GPT-4 fine-tuned models maintain the original word order of the source, yielding (1), (2), (3), and (4) sequentially. Conversely, the OFFLINE fine-tuned model retains all the content from the source but reorders it as (1), (4), (3), and (2). In contrast, the SIC fine-tuned model translates only (1), omitting the rest. This example demonstrates that both GPT-3.5 and GPT-4 achieved maintaining phrase order in the source. These results suggest that while GPT-4 is considered superior to GPT-3.5 in terms of model ability, however for this task, the source language phrase order preservation, GPT-3.5 satisfies to fulfill the task.

E.2 Quality

We focus on the quality using reference-free metrics to avoid biases inherent in references. Despite increasing wait-k values, SIC exhibits low output quality as observed in the outputs (Figure 3, Figure 4, Figure 5, Figure 6, Figure 7, Figure 6). Although training SiMT and SiST with real SI data is assumed to be beneficial for learning real-SI tactics, relying solely on SI transcripts proves inadequate for effective model training. Similarly, pretrained models trained exclusively on MuST-C v2.0, which consists of offline translation data with frequent reordering, do not perform as well as the OFFLINE model, which is fine-tuned with NAIST-SIC-Aligned offline translation. This suggests that even though the translation style is offline, fine-tuning with additional offline translation data is effective for this task. Although OFFLINE demonstrates competitive performance on tst-COMMON, even at small wait-k values such as $k = 3$ or higher, these models result in lower quality at smaller wait-k values on chunk-wise test datasets, suggesting potential overfitting to the translation style in tst-COMMON. Conversely, GPT-3.5 and GPT-4 consistently deliver competitive results across both test sets.

Source	(1) I just came back from a community that / (2) holds the secret / (3) to human survival.
Reference	(3) 私は人類の生存に関わる (*to human survival*) / (2) 秘密を握る (*holds the secret*) / (1) あるコミュニティから戻ってきたばかりです (*I just came back from a community*)。
Pretrain	(1) ちょうどコミュニティから戻って (*I just came back from a community*) / (2) シークレットを (*the secret*) / (3) 人間に持つようになりました (*holds to human*)。
SIC	(1) コモンティから戻って来たんです (*I came back from a community*)。
OFFLINE	(1) ちょうど、コミュニティから戻り (*I just came back from a community*)、/ (2) シカゴに秘密を隠しました (*hid a secret in Chicago*)。
GPT-3.5	(1) ちょうどコミュニティから戻ってきた (*I just came back from a community*)。 / (2) それはシナリオに秘密を保持している (*holds the scenario secret*)。 / (3) 人間の生存に (*to human survival*)。
GPT-4	(1) ちょうど戻ってきたのは、コミュニティからで (*I just came back from a community*)、/ (3) それは人類に (*human*) / (2) 秘密を秘めている (*holds the secret*)。

Table 8: Example of output sentences in Pretrain, SIC, OFFLINE, GPT-3.5, and GPT-4 on tst-COMMON in wait$k$=7 on Text-to-Text setting. GPT-3.5 maintains source word order completely, while GPT-4 allows small reordering in (2) and (3), which increases fluency.

Source	(1) I came to realize, / (2) as a physician, / (3) that I was working toward a goal / (4) which was different from the goal of evolution / (5) not necessarily contradictory, just different.
Reference	(2) 私は医師として (*as a physician*) / (1) 気づきました (*I came to realize*) / (3) 私は目標に向かって働いていますが (*I was working toward a goal*) / (4) それは進化の目標とは異なっていました (*which was different from the goal of evolution*) / (5) 矛盾というわけではありませんが単に違ったのです (*not necessarily contradictory, just different*)。
Pretrain	(1) 気づきに来ました (*I came to realize*)
SIC	(1) 私が、気づいたのは (*I came to realize*)、
OFFLINE	(1) 気づいたのです (*I came to realize*)。
GPT-3.5	(1) 私は気づいた (*I came to realize*)。 / (2) 医師として (*as a physician*)、 / (3) 私が目指していた目標は (*I was working toward a goal*) / (4) 異なることに (*which was different*)。
GPT-4	(1) 私は気づきました (*I came to realize*)、 / (2) 医師として(*as a physician*)、 / (3) 私が取り組んでいた目標は (*I was working toward a goal*)、/ (4) それは異なる、その目標は進化の目標から (*it is different, its goal is different from the goal of evolution*)、/ (5) 必ずしも矛盾するわけではない、単に異なるものです (*not necessarily contradictory, just different*)。

Table 9: Example of output sentences in Pretrain, SIC, OFFLINE, GPT-3.5, and GPT-4 on tst-COMMON in wait$k$=7 on Text-to-Text setting. (5) is omitted in GPT-3.5, while the whole content in the source is kept in GPT-4.

E.3 Latency

In this section, our analysis regarding latency concentrates on Pretrained, OFFLINE, GPT-3.5, and GPT-4. We exclude SIC due to its short outputs with poor quality in Table 5, and serious repetitions in Table 12. In AL and LAAL, both GPT-3.5 and GPT-4 demonstrate smaller latency compared to Pretrain and OFFLINE across both text-to-text and speech-to-text settings (Figure 3, Figure 4, Figure 5, Figure 6, Figure 7, Figure 6). In ATD, Pretrain and OFFLINE exhibit smaller latency in text-to-text settings compared to GPT-3.5 and GPT-4, whereas LLM-SI-Corpus achieves smaller latency than OFFLINE and Pretrain in speech-to-text settings. This discrepancy arises from the tendency that Pretrained and OFFLINE produce shorter translation outputs than GPT-3.5 and GPT-4 in text-to-text settings (Table 13), serious repetitions, leading to long latency, and such tendencies are effectively captured by ATD, which accounts for both start and end timing to measure latency.

E.4 Chunking

Figure 9 shows the differences in the number of chunks per sentence between the Chunk-wise data and the LLM-SI-Corpus (GPT-3.5 and GPT-4) in the test set. It compares how much the chunk sizes in GPT-3.5 and GPT-4 differ from the chunk-wise data, assuming the latter is considered the oracle. The findings indicate that the chunk size in GPT-4

Source	(1) So I went and met with his brother and father (2) and said, (3) "We're going to give you this money. What are you going to do with it?"
Reference	(1) お兄さんとお父さんに会い (*I met with his brother and father*) / (3)「支援金を差し上げますが何に使いますか (*We're going to give you this money. What are you going to do with it*)？」 / (2) と尋ねました (*said*)。
Pretrain	(1) それで私は彼の兄弟と父に会い (*So I met with his brother and father*) / (2) こう言いました (*said*)。
SIC	(1) 彼と会いました (*I met him*)。
OFFLINE	(1) 彼と会ったのは、兄と父親と (*I met him with brother and father*)、/ (2) こう言いました (*said*)。
GPT-3.5	(1) だから、私は彼の兄と父と会いました (*So I went and met with his brother and father*)。/ (2) そして、言いました (*and said*)、/ (3)「わかるでしょう、このお金を渡します (*You understand. I'm giving you this money*)」。
GPT-4	(1) だから、私は行きました。そして、彼の兄と父親に会いました (*So I met with his brother and father*)。/ (2) そして、言いました (*and said*)、/ (3)「このお金をあなたにあげますね、何をしますか (*I give you this money, what do you do with it*)?」。

Table 10: Example of output sentences in Pretrain, SIC, OFFLINE, GPT-3.5, and GPT-4 on tst-COMMON in wait$k$=7 on Speech-to-Text setting. GPT-4 is better than GPT-3.5 at fluency.

Source	(1) Back in New York, / (2) I am the head of development / (3) for a non-profit / (4) called Robin Hood.
Reference	(1) 私はニューヨークにある (*back in New York I am*) / (4) ロビンフッド財団で (*at the Robin Hood Foundation*) / (2) 組織開発の責任者をしています (*I'm responsible for organizational development*)。
Pretrain	(1) バック・イン・ニューヨーク (*back in New York*) / (2) 私は開発部門のトップで (*I am the head of development*) / (4) ロビン・フッドと呼ばれます (*called Robin Hood*)。
SIC	(1)ニューヨークに戻ります (*back in New York*)。
OFFLINE	(1)バック・イン・ニューヨークでは (*back in New York*)、/ (4) 私は、ロビン・フッドという (*I am a Robin Hood*) / (3) 非営利団体の (*non-profit organizations*)、/ (2) 開発部門のトップです (*head of development*)。
GPT-3.5	(1) ニューヨークに戻ると (*back in New York*) / (2) 私は開発の責任者です (*I am the head of development*)。/ (3) 非利益のために (*for non-profit organizations*)、/ (4) ロビンフッドと呼ばれる (*called Robin Hood*)。
GPT-4	(1)ニューヨークに戻って (*back in New York*)、/ (2) 私はその開発の責任者です (*I am the head of development*)。/ (3)それは、非営利のための (*for non-profit organizations*)、/ (4) ロビンフッドと呼ばれる利益のためのものです (*it's called Robin Hood for profit*)。

Table 11: Example of output sentences in Pretrain, SIC, OFFLINE, GPT-3.5, and GPT-4 on tst-COMMON in wait-k=7 on Text-to-Text setting.

Source	(1) And I spent 30 days / (2) eating nothing but this – / (3) fun in the beginning, / (4) little difficult in the middle, / (5) very dangerous in the end.
Reference	(1) そしてこればかり30日間 (*and all this for 30 days*) / (2) 食べたときは (*when I eat it*) / (3) 最初は楽しかったのが (*fun in the beginning*) / (4) 途中で困難に (*little difficult in the middle*) / (5) 最後には非常に危険となりました (*very dangerous in the end*)。
Pretrain	(2) 何でも食べない (*eating nothing*) / (1) 30日間 (*30 days*) / (3) 最初から楽しみました (*fun in the beginning*)。
SIC	(1) 三十日 (*30 days*)、 / (2) これ、これ、これ、... (*this this this ...*)。
OFFLINE	(1) 30日も (*30 days*)、/ (2) こんなものを食べて、何でも食べました、 (*eating nothing but this*) / (3) (笑)、最初から、ちょっと面白いですね ((*laughs), fun in the beginning*)。
GPT-3.5	(1) だから、私は30日間過ごしました。(*And I spent 30 days*) / (2) これ以外何も食べませんでした (*eating nothing but this*) / (3) 最初から楽しい (*fun in the beginning*)。
GPT-4	(1) そして、私は30日間を過ごしました (*And I spent 30 days*) / (2) これ以外何も食べていません (*eating nothing but this*)、/ (3) 始めに楽しいです (*fun in the beginning*)。 / (4) 少し難しいです。真ん中に。(*little difficult in the middle*)。

Table 12: SIC results in serious repetition at tst-COMMON wait-k = 17 on Speech-To-Text settings.

Source	(1) But still it was a real foot race / (2) against the other volunteers / (3) to get to the captain in charge / (4) to find out / (5) what our assignments would be.
Reference	(3) それでも団長を見つけて (*still find the captain*) / (4) 任務を割り振ってもらうのに (*to assign a mission*) / (2) 他のボランティアと (*against the other volunteers*) / (1) 激しい競走になりました (*it was a real foot race*)。
Pretrain	(2) それでも足を踏みにじる他のボランティアたちに (*still, to the other volunteers who trample their feet*) / (3) キャプテンに (*the captain*) / (1) 足を踏みにじる真のレースでした (*it was a real foot race*) / (5) 私たちの課題を (*our assignments*) / (4) 見つけるためです (*to find out*)。
SIC	(1)でも (*but*)、
OFFLINE	(1) それでも、実に、アフトレースで (*it was a real foot race*)、 / (2) 他のボランティアが (*the other volunteers*) / (3) キャプテンに (*the captain*)、 / 手紙を送り (*send a letter*)、 / (5) 課題を (*assignments*) (4) 探しました (*find out*)。
GPT-3.5	(1) それでも、それは本物の足のレースでした (*it was a real foot race*)。/ (2) 他のボランティアたちに対して (*against the other volunteers*)、 / (3) キャプテンに向かうために (*against the captain*)、/ (5) 私たちの課題が (*our assignments*) / (4) 何かを見つけるために (*to find out what would be*)。
GPT-4	(1) それでも、それは本当に足の運命でした (*it was a real foot race*)。 / (2) 他のボランティアたちに対して (*against the other volunteers*)、/ (3) キャプテンに到着するために (*to get to the captain in charge*)、/ (5) 私たちの標的が何であるかを (*what our targets would be*) / (4) 調べるために (*to find out*)。

Table 13: Example of output sentences in Pretrain, SIC, OFFLINE, GPT-3.5, and GPT-4 on tst-COMMON in wait$k$=7 on Text-to-Text setting. Both GPT-3.5 and GPT-4 achieve fluency allowing small reordering in (4) and (5).

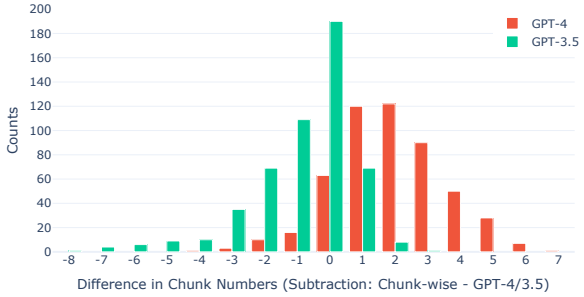

Figure 9: The difference in chunk numbers between Chunk-wise and GPT-4/GPT-3.5. The total number of sentences is 511.

is smaller than in the chunk-wise data, while GPT-3.5 tends to produce larger chunks compared to the chunk-wise data. Although we included this analysis, it is important to note that chunking is only one criterion, and matching chunk sizes does not necessarily indicate that the translation quality based on the chunk size is good.

E.5 Misalignment between Source Input and the SI data

In our corpus analysis, we found that both NAIST-SIC-Aligned and MuST-C v2.0 contain noise in the form of misalignment between the source and target sentences. This misalignment results in the shift of information, e.g., information in a sentence appearing in its neighbors, leading to imbalanced sentence correspondences. When dealing with MuST-C v2.0, difficulty arises in aligning audio input features with subtitles due to space limitations, which may lead to unbalanced correspondences. Similarly, in the case of NAIST-SIC-Aligned, which utilizes Japanese transcripts of interpreted data, aligning source text becomes challenging. This is due to the SI characteristics, involving omissions and summaries, which further complicate the alignment process due to imbalances between the source and target transcripts. Some examples are shown in Table 14, Table 15. Addressing alignment in unbalanced sentences emerges as a particularly challenging aspect of SI, representing an important area for future research.

E.6 Toward Applying to Other Language Pairs

We conducted a preliminary investigation to determine whether our proposed method could be scaled to multiple language pairs, including English-to-Chinese (en-zh), and English-to-German (en-de), using the MuST-C v2.0 tst-COMMON dataset (Di Gangi et al., 2019). We translated the source into each target language by replacing the "output:Japanese" with Chinese and German in the system, as shown in Figure 2. The same method described in Section 4 was used to measure monotonicity between the source and target languages, using Spearman's correlation coefficient based on the alignments obtained from Awesome-align (Dou and Neubig, 2021). From Table 16, we found that our method improves mono-

Source	Target
Really important. So I'm committing to potatoes; I'm committing to milk; I'm committing to leeks and broccoli all very important stuff.	これが、 問題なわけです。ポテト、そしてミルク、 そして、ネギ、ブロッコリー、こういったものに対して、
Because of our differences, we create and sustain life. So we should embrace our difference and aim for challenge.	違いがあるから 持続可能性を生み出すことができます。

Table 14: Example of misalignment sentence pairs in SIC.

Source	Target
I do the philosophy of art, aesthetics, actually,for a living. I try to figure out intellectually, philosophically, and psychologically, what the experience of beauty is, what sensibly can be said about it, and how people go off the rails in trying to understand it.;	私は美の哲学、美学を。 生業としています、美という体験は何なのか、美について確かに言えることは何か、人は美を理解しようとして、いかに道に迷うかといったことを、知的、哲学的、心理学的に解明しようとしています。
Now this is an extremely complicated subject, in part because the things that we call beautiful are so different. I mean just think of the sheer variety a baby's face, Berlioz's "Harold in Italy," movies like "The Wizard of Oz" or the plays of Chekhov, a central California landscape, a Hokusai view of Mt. Fuji, "Der Rosenkavalier," a stunning matchwinning goal in a World Cup soccer match, Van Gogh's "Starry Night," a Jane Austen novel, Fred Astaire dancing across the screen.	美というのは恐ろしく込み入ったテーマであり、私たちが美しいと呼んでいるものには、非常に大きな幅があります、いかにバラエティに富んでいることか、赤ちゃんの顔。 ベルリオーズの「イタリアのハロルド」、「オズの魔法使い」のような映画、チェーホフの戯曲、中部カリフォルニアの風景、北斎の富士山の絵、「ばらの騎士」。

Table 15: Example of misalignment sentence pairs in MuST-C v2.0.

Language	**Data**	**Monotonicity**
En-Ja	MuST-C	0.522
	Ours (GPT-3.5)	0.798
	Ours (GPT-4)	**0.815**
En-Zh	MuST-C	0.875
	Ours (GPT-3.5)	0.929
	Ours (GPT-4)	**0.952**
En-De	MuST-C	0.938
	Ours (GPT-3.5)	**0.960**
	Ours (GPT-4)	0.958

Table 16: The table compares word order monotonicity across three language pairs (en-ja, en-zh, en-de) in the Must-C v2.0 tst-COMMON, similar to Table 2.

tonicity for the other language pairs, though the improvement was not as significant as what we observed in English-to-Japanese As this study focuses on verifying the SI data creation method based on CWMT, the extension to other languages will be addressed in future work. Additionally, since the CWMT guidelines and protocols are specifically designed for English-to-Japanese, there is room for improvement, such as exploring more generalized methods for other languages.