SIMULTANEOUS SPEECH-TO-SPEECH TRANSLATION SYSTEM WITH TRANSFORMER-BASED INCREMENTAL ASR, MT, AND TTS

Ryo Fukuda, Sashi Novitasari, Yui Oka, Yasumasa Kano, Yuki Yano, Yuka Ko, Hirotaka Tokuyama,
Kosuke Doi, Tomoya Yanagita, Sakriani Sakti, Katsuhito Sudoh, Satoshi Nakamura

Nara Institute of Science and Technology, Japan

ABSTRACT

In this paper, we present an English-to-Japanese simultaneous speech-to-speech translation (S2ST) system. It has three Transformer-based incremental processing modules for S2ST: automatic speech recognition (ASR), machine translation (MT), and text-to-speech synthesis (TTS). We also evaluated its system-level latency in addition to the module-level latency and accuracy.

***Index Terms*—** Simultaneous translation, Speech translation, English-to-Japanese translation

1. INTRODUCTION

Speech-to-speech translation (S2ST) is a promising and challenging technology for assisting cross-lingual human conversation [1, 2, 3]. Recent deep learning technologies advanced speech and language processing, and many studies addressed the problem of real-time automatic S2ST. However, we face a crucial problem of the delay of the S2ST processes. Since an S2ST system usually handle speech inputs at the utterance or sentence level, it has to wait for the end of an utterance so that the delay becomes proportional to the length of the input. It is not useful for long monologues such as lectures. Simultaneous interpretaion is often used in such situations, but it is also a very challenging task that requires experienced interpreters.

In this paper, we focus on the problem of simultaneous S2ST from English to Japanese and present a system based on neural networks called Transformer. Note that we differentiate simultaneous *translation* from simultaneous *interpretation*, because the current simultaneous translation does not include interpretation efforts such as summarization. This problem requires real-time and incremental processing that works simultaneously with the input. Most previous attempts for simultaneous speech translation focused on speech-to-text translation between English and Europearn languages [4, 5, 6]. Our work aims for S2ST from English to Japanese. English and Japanese are very different in their syntax and difficult to translate each other. Our system cascades three modules: incremental speech recognition (ISR), incremental machine translation (IMT), and text-to-speech synthesis (ITTS). End-to-end approaches are used in recent studies on speech translation [7], but it is still difficult to apply them for English-to-Japanese S2ST [8].

We evaluate our system in system-level latency in addition to module-level performance on S2ST from English TED Talks to Japanese. We have two system-level latency metrics: (1) the system-level Ear-Voice Span consisting of computation time and cascading delay, and (2) the cumulative speaking latency derived from the overlap of TTS outputs. The ISR, IMT, and ITTS modules are evaluated by their standard metrics. This is the first attempt of a system-level evaluation for a simultaneous S2ST system and will benefit future studies.

2. SIMULTANEOUS SPEECH-TO-SPEECH TRANSLATION

Given input sequence $X = x_1, x_2, ..., x_{|X|}$, we predict the corresponding output sequence $Y = y_1, y_2, ..., y_{|Y|}$ by sequence-to-sequence transduction. In simultaneous translation, we make the prediction one-by-one on subsequences. Suppose we have predicted output subsequence $Y_1^j = y_1, y_2, ..., y_j$ from partial input observations $X_1^i = x_1, x_2, ..., x_i$. When we observe the next partial input $X_{i+1}^{i'} = x_{i+1}, ..., x_{i'}$, we predict the corresponding output subsequence $Y_{j+1}^{j'} = y_{j+1}, ..., k_{j'}$ based on the following formula:

$$Y_{j+1}^{j'} = \underset{\hat{Y}}{\operatorname{argmax}} P(\hat{Y}|X_1^i, X_{i+1}^{i'}, Y_1^j), \tag{1}$$

where $\hat{Y}$ is a subsequence-level partial prediction.

Here we face a problem in deciding the length of a partial observation to make a partial prediction. Non-simultaneous translation waits until the end of the sentence, but we must make partial predictions incrementally for *simultaneous* translation. This work uses simple fixed-length criteria for incremental processing, as described in Section 3.

3. INCREMENTAL PROCESSING MODULES

3.1. Incremental speech recognition

We tackle the latency problem in ASR by using our ISR method [9] on the Transformer-based encoder-decoder model

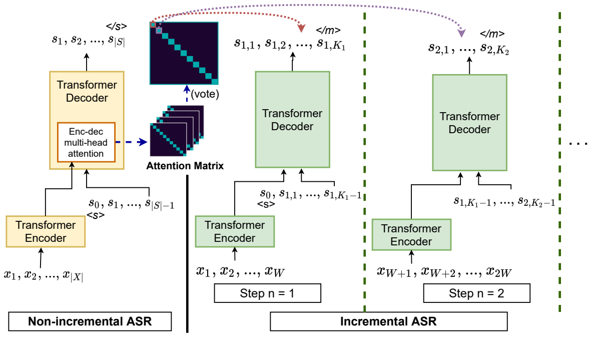

Fig. 1. Transformer-based ISR construction with attention transfer [9] from a standard Transformer-based ASR

[10]. We use a teacher-student framework to train the ISR model. We firstly train a non-incremental ASR as the teacher model and then train an ISR as a student model to do block-wise input processing by learning the knowledge of the teacher model. The student ISR model has the same model architecture as the teacher ASR model and learns to mimic the speech-text alignment based on the encoder-decoder attention by the teacher model.

Suppose the teacher ASR model transcribes speech utterance $X = x_1, x_2, ..., x_{|X|}$ with length $|X|$ into token sequence $S = s_1, s_2, ..., s_{|S|}$ with length $|S|$. The teacher's attention-based alignment is extracted from the attention between the hidden states of the encoder and decoder. From the attention sequences in all the layers and heads, the speech-text alignment for the ISR training was generated through teacher-forcing decoding to decide the final alignment A by a majority vote. We split X into M sub-segments $\bar{X} = [\bar{x}_1, \dots, \bar{x}_M]$ and S into $\bar{S} = [\bar{s}_0, \dots, \bar{s}_M]$, using A. Here, an end-of-block symbol </m> is added at the end of each sub-segment.

ISR learns the incremental steps by learning a pair of $\bar{X}$ (with the length of the W frames for each speech segment) and $\bar{S}$ (Fig. 1). By inference, ISR performs an incremental recognition step when the speech buffer achieved W frames to predict the corresponding transcription and then moves to the next incremental step when it predicts a </m> symbol. In our ISR, we allowed the model to take look-back and look-ahead input sequences as the contextual input to provide more detailed speech information.

3.2. Incremental machine translation

MT suffers from a reordering problem, caused by the syntactic differences between the source and target languages. Although it is not serious in current neural MT [11, 12], it remains very problematic when we consider IMT because we cannot observe a complete sentence. Previous studies took inputs and predicted the corresponding outputs incrementally in an adaptive manner [13, 14]. In this work, we use a method called *wait-k*, waits for the first k input tokens before starting the decoding process [15]. The *wait-k* IMT model translates a token sequence of source language $S = s_1, \dots, s_{|S|}$ into target language $T = t_1, \dots, t_{|T|}$. The decoder predicts an output token using a partial input. k is a hyperparameter for the fixed number of input tokens for the initial wait; setting k larger leads larger delays, and a smaller k worsens the output predictions due to poor context information.

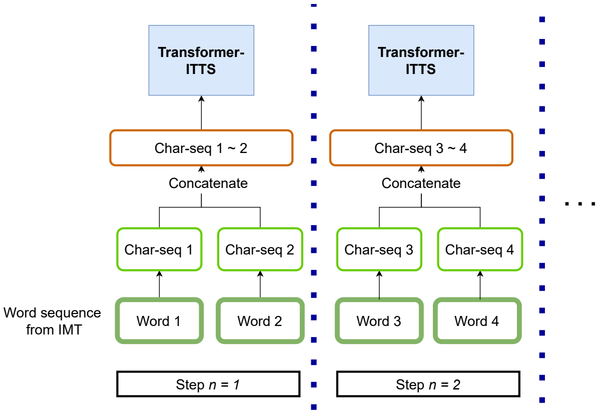

Fig. 2. Incremental text-to-speech synthesis system

In this work, we also introduce two extensions of IMT training: knowledge distillation and chunk shuffling. We distill knowledge from a non-incremental teacher MT model to a student *wait-k* model by sequence-level knowledge distillation [16]. We apply chunk shuffling, random reordering of Japanese chunks called *bunsetsu*, for data augmentation. We expect it encourages the IMT to be monotonic, based on a characteristic of Japanese in which the order of *bunsetsu* chunks can be relaxed. Suppose we have target language sentence $T = t_1, \dots, t_{|T|}$ in our training set. We segment T greedily into a sequence of chunks $\bar{T} = \mathcal{C}_1, \dots, \mathcal{C}_Q$. The length of each chunk is set to k (i.e., delay hyperparameter in *wait-k*) tokens: $\mathcal{C}_q = t_{q_1}, \dots, t_{q_k}$, except for the last chunk $\mathcal{C}_Q$ that can be shorter than k. Finally, we shuffle the chunks with a pre-defined fixed probability p_r.

3.3. Incremental text-to-speech synthesis

ITTS synthesizes speech based on a short text segment. In this work, our ITTS structure is based on the Transformer-based TTS proposed by Li et al. [17]. It consists of an encoder and a decoder, both of which have a Transformer structure. Given the input sequence of tokens, the encoder maps the input into semantic space to generate the sequence of the encoder's hidden states. This sequence is then utilized by the decoder, along with the decoder output in the previous timestep, to predict the speech's Mel-spectrogram and a stop token that marks the end-of-sentence. In this work, from the Mel-spectrogram, we generated speech signals by generating a magnitude spectrogram using a CBHG (1-D Convolution Bank + Highway + bidirectional GRU) module that resembles the Tacotron framework [18], followed by speech phase spectrogram estimation with the Griffin-Lim algorithm and an inverse short-time Fourier transform (STFT).

The process of our ITTS module in a S2ST system is illustrated in Fig. 2. For each incremental step, ITTS takes the character sequence of a fixed number of words and synthesizes the corresponding speech. Since it takes a sequence of Japanese *kana* phonograms as input, the output of the preceding IMT, including Japanese *kanji* morphograms, must be converted accordingly. We applied lattice-based tokenization and *kanji*-to-*kana* conversion[1] to IMT output strings, which were de-tokenized from subword sequences.

We trained the ITTS using word-level alignment between speech and text[2]. Given complete text $T = [t_1, t_2, ..., t_{|T|}]$ with length $|T|$ and corresponding speech utterance $Y = [y_1, y_2, ..., y_{|Y|}]$ with length $|Y|$, we split T into Q sub-segments $\bar{T} = [\bar{t}_0, \ldots, \bar{t}_Q]$ and Y into Q sub-segments $\bar{Y} = [\bar{y}_0, \ldots, \bar{s}_Q]$, based on the alignment. All $\bar{T}$ sub-segments have the same number of words, which are converted into characters when training the ITTS. Each $\bar{Y}$ sub-segment is concatenated with a blank at the end to mark the end-of-speech segment. Here, the blank tensor corresponds to a stop token label during training. By inference, ITTS stops the incremental step when the decoder predicts the stop token and proceeds to the next incremental step by taking the next step's input. We allowed the model to take look-back and look-ahead input sequences as contextual input. Similar to the main input, the number of words in the contextual input is fixed and also learned during the ITTS training.

4. EVALUATION

We evaluated our system by an English-to-Japanese simultaneous S2ST experiment. We investigated the detailed system performance by focusing on: (1) *system-level latency* and (2) *module-level quality*.

4.1. Evaluation setup

The ISR, IMT, and ITTS models were trained with a few different latency parameters for comparison, motivated by IWSLT evaluation campaign [20].

We constructed ISR with Transformer-big-based model configuration proposed in Speech-Transformer [10]. The ISR training was done using TED-LIUM release 1 [21] with a total of 774 talks (56.8k cut utterances, representing about 118 hours of speech). We extracted 80 dimensions of Mel-spectrogram features with a 50-ms frame window and a 12.5-ms window shift as the ISR input. The English speech transcriptions in the ISR training were segmented into subwords using a byte-pair encoding[3] model, which is identical as the one in the IMT part. We trained two ISR systems with different input delay configurations: 64 frames and 96 frames. The ISR with 64 frames of input required 32 frames as the main input and the next 32 frames as the look-ahead input for each incremental step. In the ISR with 96 frames, it takes 64 frames as the main input and 32 look-ahead frames for each incremental step. Both models are also allowed to see the look-back frames with a range of 256 frames. The module-level ISR systems were evaluated on the TED-LIUM release 1 test set ("TED-LIUM test") that consisted of 1155 cut short speech utterances (average 7.88 sec) of long TED talks.

The IMT model configuration was based on Transformer-base [12]. We first trained the model using the WMT 2020 news task data (17.9 million sentence pairs) and fine-tuned using IWSLT 2017 data (223 thousand sentence pairs). In the fine-tuning, we examined different configurations for knowledge distillation and chunk shuffling and chose the configuration based on the results on the validation data. We tokenized the sentences into subwords based on Byte Pair Encoding [22]. The vocabulary was shared over the source and target languages using 16,000 entries.

The ITTS model configuration was based on Transformer-based TTS [17]. We utilized the JSUT dataset [23] for model training, which consists of one Japanese woman's speech with 5.2k pairs of speech utterances and their corresponding transcriptions. We used 5k utterances for training, 100 for development, and 100 for test. We extracted 80 Mel-spectrogram dimensions with a 50-ms frame window and a 12.5-ms window shift as the speech feature targets. We constructed two ITTSs with different input delays in an incremental step: 5 words and 7 words. The ITTS with input of 5 words took 3 main words and 2 look-ahead words for each incremental step. The ITTS with input of 7 words took 5 main words and 2 look-ahead words as the input for each incremental step. Both models were also allowed to see the look-back input in the range of 10 words.

The system-level test set consisted of eight TED talks (1.89 hours in total) with English and Japanese subtitles.

4.2. Latency

We measured the latency of our system with two different metrics: *Ear-Voice Span (EVS)* and *cumulative speaking latency*.

4.2.1. Ear-Voice Span (EVS)

EVS is a common measure of simultaneous interpretation latency, calculated as the delay between the start of input and interpretation speech. We used it for the evaluation of the system-level delay, which we call *system-level EVS*, caused by the module-level processing and inter-module communications. For this purpose, we aligned the module-level outputs with the output timestamps (Fig. 3). The delays are shown in Table 1.

[1] We used PyKakasi for the tokenization and morphogram conversion (https://github.com/miurahr/pykakasi)

[2] We generated the alignment using Montreal Forced Aligner [19]

[3] subword-nmt toolkit (https://github.com/rsennrich/subword-nmt)

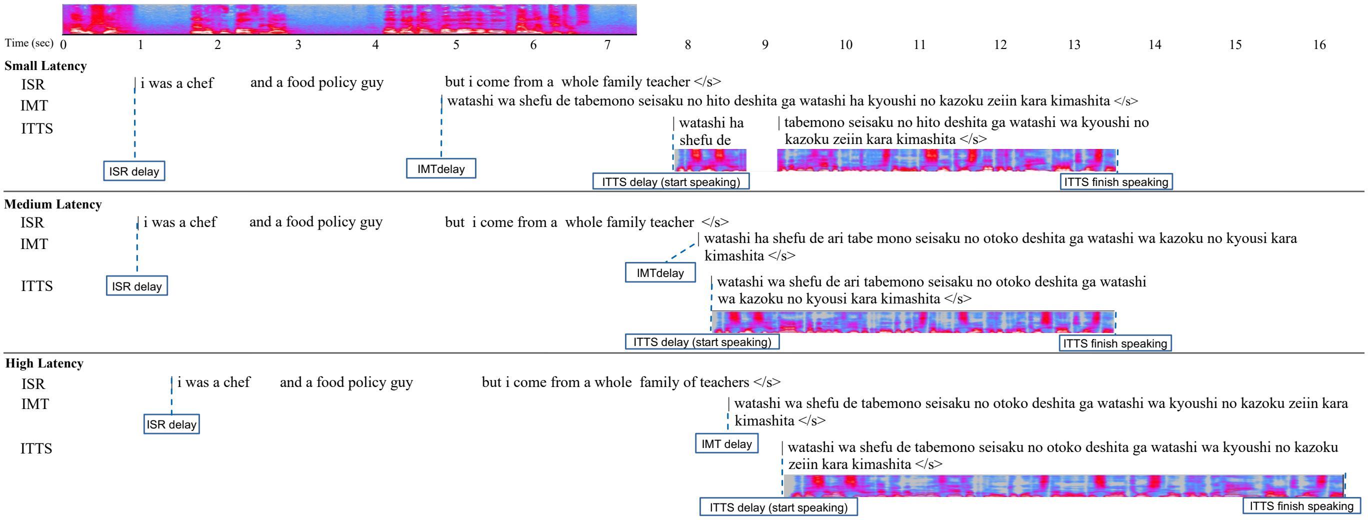
Time (sec) 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Small Latency
ISR
IMT
ITTS
i was a chef
and a food policy guy
but i come from a whole family teacher </s>
watashi wa shefu de tabemono seisaku no hito deshita ga watashi ha kyoushi no kazoku zeiin kara kimashita </s>
watashi ha shefu de
tabemono seisaku no hito deshita ga watashi wa kyoushi no kazoku zeiin kara kimashita </s>
ISR delay
IMTdelay
ITTS delay (start speaking)
ITTS finish speaking
Medium Latency
ISR
IMT
ITTS
i was a chef
and a food policy guy
but i come from a whole family teacher </s>
watashi ha shefu de ari tabe mono seisaku no otoko deshita ga watashi wa kazoku no kyousi kara kimashita </s>
IMTdelay
ISR delay
watashi wa shefu de ari tabemono seisaku no otoko deshita ga watashi wa kazoku no kyousi kara kimashita </s>
ITTS delay (start speaking)
ITTS finish speaking
High Latency
ISR
IMT
ITTS
i was a chef
and a food policy guy
but i come from a whole family of teachers </s>
watashi wa shefu de tabemono seisaku no otoko deshita ga watashi wa kyoushi no kazoku zeiin kara kimashita </s>
ISR delay
IMT delay
watashi wa shefu de tabemono seisaku no otoko deshita ga watashi wa kyoushi no kazoku zeiin kara kimashita </s>
ITTS delay (start speaking)
ITTS finish speaking

Fig. 3. ISR, IMT, and ITTS output and alignment example

System	ISR delay	IMT delay	ITTS delay
Low latency	0.93 (64 frames)	4.69 (*k* 10)	8.81 (5 words)
Medium latency	0.93 (64 frames)	8.43 (*k* 20)	11.87 (5 words)
High latency	1.30 (96 frames)	11.47 (*k* 30)	16.91 (7 words)

Table 1. Module-level processing delays (sec) from beginning of speech inputs by ISR, IMT, and ITTS: Experiment was conducted on long talk source speech (average length 14.15 min). Note: later modules have to work after their preceding module, so they include delays in preceding module.

From Table 1, we can see that the ISR worked with fixed delays, while the latter IMT and ITTS modules were influenced by the delays of the preceding modules.

4.2.2. *Cumulative speaking latency*

We identified long delays by the ITTS module. A TED Talk speaker generally talks smoothly without disfuency and long pauses, so following inputs come to the ITTS module even during the play of the previous output waveform (Fig. 3). As a result, such TTS outputs were queued and played later with large delays, which we call the *cumulative speaking latency*. This overlap becomes critical in the long speech inputs. In this work, we conducted segment-based evaluation and will tackle talk-level evaluation in future studies.

4.3. Quality

We evaluated the quality of the results by our system, including the module-level one for all three modules: ISR, IMT, and ITTS. In the subjective evaluation described below, twelve graduate students evaluated the results of IMT and ITTS.

Model	**TED-LIUM test** (Cut utterances)	**TED talk test** (Long talks)
Topline (non-incremental)		
Seq2seq LSTM	23.78	25.46
Seq2seq Transformer	21.15	20.74
Baseline: Seq2seq ISR (LSTM)		
Input = 64 frames/step	29.01	31.88
Input = 96 frames/step	28.80	32.43
Proposed: Transformer ISR		
Input = 64 frames/step	28.55	32.06
Input = 96 frames/step	27.31	25.01

Table 2. ASR performance (WER%) on TED-LIUM and our TED test data.

4.3.1. *Incremental speech recognition*

We evaluated the ISR results with the word error rate (WER). We compared our ISR with the seq2seq LSTM-based ISR with attention transfer [9] as the baseline and the standard non-incremental ASR as our topline (Table 2). Our proposed Transformer-based ISR outperformed the LSTM-based ISR, especially in the long talk recognition in which the task reflects the speech condition in simultaneous S2ST.

4.3.2. *Incremental machine translation*

We evaluated the ISR+IMT results with BLEU [24] using TED Japanese subtitle as the reference, calculated by SacreBLEU. Table 3 shows the results. Unfortunately, our BLEU-4 result was very low for the following various reasons: (1) Domain mismatch: the in-domain (TED) training data was not large; (2) Style mismatch: the TED Japanese subtitles are often idiomatic translations and they may not be suitable for surface-based evaluation like BLEU; and (3) ASR error propagation.

System	ASR WER	MT BLEU	Subjective Evaluation Adequacy	Subjective Evaluation Fluency
ST Topline (non-incremental)				
Correct text + MT	0.00	15.7	3.41	3.93
Standard ASR + MT	20.74	12.8	3.20	4.01
ST with baseline IMT (incremental)				
ISR (64) + IMT (small)	32.06	4.5	2.60	2.56
ISR (64) + IMT (medium)	32.06	7.5	2.86	3.30
ISR (96) + IMT (high)	25.01	8.1	3.31	3.82
ST with proposed IMT (incremental)				
ISR (64) + IMT (small)	32.06	5.1	2.80	3.03
ISR (64) + IMT (medium)	32.06	8.4	2.98	3.54
ISR (96) + IMT (high)	25.01	9.4	3.34	3.80

Table 3. Speech translation performance on our TED talk test data.

System	Tacotron TTS	Transformer TTS
TTS (non-incremental)	0.57	0.51
ITTS (5 words/step)	0.77	0.58
ITTS (7 words/step)	0.65	0.57

Table 4. L2-norm loss between Mel-spectrogram of original and ITTS speech on JSUT test data.

We also subjectively evaluated the ISR+IMT results, based on 1-5 scale adequacy and fluency metrics used in past MT evaluation campaigns [25]. The rightmost columns in Table 3 show the results. The adequacy and fluency results showed correlation with the BLEU-1 results, which are dominated by word unigram precision. The proposed method demonstrated better results than the baseline in the small and medium latency conditions.

4.3.3. *Incremental speech synthesis*

Our ITTS evaluations were done with objective and subjective evaluations. We conducted an objective evaluation on JSUT using L2 loss between the correct Mel-spectrogram and the Mel-spectrogram predicted by ITTS (Table 4). Here our baselines are the ITTS with a Tacotron (LSTM) structure, and the topline is the standard non-incremental Transformer TTS. For the subjective evaluation, we performed an AB preference test with our TED talk test data to compare and evaluate the naturalness of the synthesized speech. Sixty pairs of synthetic speech samples generated by using different methods were presented to listeners in random orders. The results show that the performance of the Transformer-based system was better than the Tacotron-based system. With respect to the difference between delays in 5 and 7 words, there were no remarkable differences both in L2-norm and subjective evaluation. Thus, the ITTS worked efficiently using 5-word delay with comparable quality to that using 7-word delay.

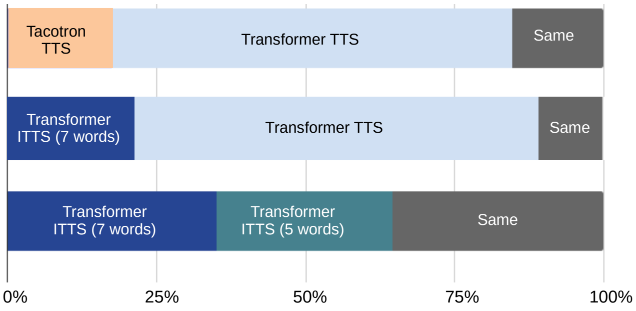

Fig. 4. AB preference test scores for ITTS on TED test data.

4.4. Discussion

From the latency viewpoint, our cascade simultaneous S2ST system worked successfully with relatively short delays. However, the problem is still challenging in quality due to various reasons including the error propagation by the cascade and data scarcity. Tight integration of the modules such as a lattice-to-sequence [26] is promising, although it is not trivial to apply such integration into simultaneous translation.

Unfortunately, we still do not have common system-level evaluation methodologies and metrics for simultaneous S2ST other than module-level ones. The two metrics used in this work focused only on latency, so we need to evaluate content delivery through objective measurement. Comparison with human interpreters in terms of content delivery and user satisfaction are very important in future.

5. CONCLUSIONS

We presented our English-to-Japanese simultaneous S2ST system and evaluated it using a TED talks dataset. The system works incrementally by the cascaded incremental processing of ASR, MT, and TTS, implemented based on Transformer. A latency evaluation revealed that module-level delay remains problematic in incremental MT and TTS, even though it can be controlled by delay hyperparameters at the cost of a drop in accuracy. Our speech-to-speech simultaneous translation system also suffers from speaking latency.

In future work, we aim to improve the accuracy and efficiency of the modules based on aggressive anticipation using large-scale pre-trained models and to decrease the ITTS latency by controlling the speaking duration.

6. ACKNOWLEDGMENTS

Part of this work was supported by JSPS KAKENHI Grant Number JP21H05054.

7. REFERENCES

[1] Satoshi Nakamura, Konstantin Markov, Hiromi Nakaiwa, Genichiro Kikui, Hisashi Kawai, Takatoshi Jitsuhiro, Jin-Song Zhang, Hirofumi Yamamoto, Eiichiro Sumita, and Seiichi Yamamoto, "The ATR Multilingual Speech-to-Speech Translation System," *IEEE Trans. ASLP*, vol. 14, no. 2, pp. 365–376, 2006.

[2] Eiichiro Sumita, Tohru Shimizu, and Satoshi Nakamura, "NICT-ATR speech-to-speech translation system," in *Proc. ACL: Demo*, 2007, pp. 25–28.

[3] Sakriani Sakti, Michael Paul, Andrew Finch, Xinhui Hu, Jinfu Ni, Noriyuki Kimura, Shigeki Matsuda, Chiori Hori, Yutaka Ashikari, Hisashi Kawai, Hideki Kashioka, Eiichiro Sumita, and Satoshi Nakamura, "Distributed speech translation technologies for multiparty multilingual communication," *ACM Trans. SLP*, vol. 9, no. 2, Aug. 2012.

[4] Jan Niehues, Ngoc-Quan Pham, Thanh-Le Ha, Matthias Sperber, and Alex Waibel, "Low-latency neural speech translation," in *Proc. Interspeech*, 2018, pp. 1293–1297.

[5] Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu, "SimulSpeech: End-to-end simultaneous speech to text translation," in *Proc. ACL*, 2020, pp. 3787–3796.

[6] Xutai Ma, Juan Pino, and Philipp Koehn, "SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation," in *Proc. AACL-IJCNLP*, 2020, pp. 582–587.

[7] Ye Jia, Ron J. Weiss, Fadi Biadsy, Wolfgang Macherey, Melvin Johnson, Zhifeng Chen, and Yonghui Wu, "Direct speech-to-speech translation with a sequence-to-sequence model," in *Proc. Interspeech*, 2019, pp. 1123–1127.

[8] Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura, "Transformer-Based Direct Speech-To-Speech Translation with Transcoder," in *Proc. IEEE SLT*, 2021, pp. 958–965.

[9] Sashi Novitasari, Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Sequence-to-Sequence Learning via Attention Transfer for Incremental Speech Recognition," in *Proc. Interspeech*, 2019, pp. 3835–3839.

[10] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. ICASSP*, 2018, pp. 5884–5888.

[11] Thang Luong, Hieu Pham, and Christopher D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, 2015, pp. 1412–1421.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is All you Need," in *Proc. NIPS*, pp. 5998–6008. 2017.

[13] Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura, "Simple, Lexicalized Choice of Translation Timing for Simultaneous Speech Translation," in *Proc. Interspeech*, 2013, pp. 3487–3491.

[14] Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li, "Learning to translate in real-time with neural machine translation," in *Proc. EACL*, 2017, pp. 1053–1062.

[15] 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, "STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework," in *Proc. ACL*, 2019, pp. 3025–3036.

[16] Yoon Kim and Alexander M Rush, "Sequence-level knowledge distillation," *arXiv preprint 1606.07947*, 2016.

[17] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu, "Neural speech synthesis with transformer network," in *Proc. AAAI*, 2019, pp. 6706–6713.

[18] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[19] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, "Montreal Forced Aligner: Trainable text-speech alignment using Kaldi," in *Proc. INTERSPEECH*, 2017, pp. 498–502.

[20] Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alexander Waibel, Changhan Wang, and Matthew Wiesner, "Findings of the IWSLT 2021 evaluation campaign," in *Proc. IWSLT*, 2021, pp. 1–29.

[21] Anthony Rousseau, Paul Deléglise, and Yannick Estève, "TED-LIUM: an automatic speech recognition dedicated corpus," in *Proc. LREC*, 2012, pp. 125–129.

[22] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural machine translation of rare words with subword units," in *Proc. ACL*, 2016, pp. 1715–1725.

[23] Ryosuke Sonobe, Shinnosuke Takamichi, and Hiroshi Saruwatari, “JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis,” *arXiv preprint 1711.00354*, 2017.

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proc. ACL*, 2002, pp. 311–318.

[25] LDC, “ Linguistic Data Annotation Specification: Assessment of Fluency and Adequacy in Translations Revision 1.5, January 25, 2005 ,” Tech. Rep., Linguistic Data Consortium, 2005.

[26] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel, “Neural lattice-to-sequence models for uncertain inputs,” in *Proc. EMNLP*, 2017, pp. 1380–1389.