TRANSFORMER-BASED DIRECT SPEECH-TO-SPEECH TRANSLATION WITH TRANSCODER

Takatomo Kano[1], *Sakriani Sakti*[1,2], *Satoshi Nakamura*[1,2]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project AIP, Japan

ABSTRACT

Traditional speech translation systems use a cascade manner that concatenates speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) synthesis to translate speech from one language to another language in a step-by-step manner. Unfortunately, since those components are trained separately, MT often struggles to handle ASR errors, resulting in unnatural translation results. Recently, one work attempted to construct direct speech translation in a single model. The model used a multi-task scheme that learns to predict not only the target speech spectrograms directly but also the source and target phoneme transcription as auxiliary tasks. However, that work was only evaluated Spanish-English language pairs with similar syntax and word order. With syntactically distant language pairs, speech translation requires distant word order, and thus direct speech frame-to-frame alignments become difficult. Another direction was to construct a single deep-learning framework while keeping the step-by-step translation process. However, such studies focused only on speech-to-text translation. Furthermore, all of these works were based on a recurrent neural network (RNN) model. In this work, we propose a step-by-step scheme to a complete end-to-end speech-to-speech translation and propose a Transformer-based speech translation using Transcoder. We compare our proposed and multi-task model using syntactically similar and distant language pairs.

Index Terms— speech-to-speech translation, Transcoder, Transformer, sequence-to-sequence model, multitask learning

1. INTRODUCTION

Speech-to-speech translation is challenging. Humans translate speech-to-speech in several steps: comprehension, transfer of meaning, and language production. First, we understand a spoken speech; then, we turn the content into target language; finally, we pronounce the translated content using target language. The traditional speech translation system follows this step-by-step process that connect ASR, MT, and TTS to translate speech to another language's speech [1]. Such step-by-step training and inference process are essential since they reduce complexity. Unfortunately, since these components are trained and tuned separately and no total optimization is performed, MT often struggles to handle ASR errors, resulting in unnatural translation results.

Few works have constructed direct speech translation in a single model using deep learning to avoid a problem in traditional speech translation. Duong et al. introduced the first study that considered speech-to-text translation by alignment and translation reranking [2], and Berard et al. built a fullfledged, end-to-end attention-based speech-to-text translation system [3]. But their results failed to outperform the traditional cascade approach. Jia et al. constructed a complete end-to-end speech-to-speech translation model [4] that used a multi-task scheme with three decoders that learned to predict not only the target speech spectrograms as its main task but also the source and target phoneme transcriptions as auxiliary tasks. During inferences, no auxiliary tasks were used, and the model performed only with a single encoder to receive the source language's speech and a single decoder to generate the target language's speech. Therefore, the model had to directly align source speech frames into target speech frames. However, ultimately, all of these previous researches on direct speech translation just focused on syntactically similar language pairs. With syntactically distant language pairs, speech translation requires distant word order and complicating direct speech frame-to-frame alignments.

Another research direction constructed a single deep-learning framework and kept the step-by-step translation process. Kano et al. [5] proposed end-to-end speech translation that directly passes the attention's weighted contextual information from one component to another. Their model transfers the attention results to the next process. Since the attention module works as a filter function for the source speech encoder states, it generates a target text from source speech encoder states, not the ASR text. Sperber et al. [6] compared multi-task- and step-by-step-based end-to-end speech-to-text translations in various dataset sizes and reported that both the multi-task and step-by-step approaches outperformed the traditional cascade speech-to-text translation system. Furthermore, the step-by-step-based speech translation model outperformed the multi-task-based speech translation model, especially with a small dataset. However, these two works only covered speech-to-text translation.

Finally, all of the above cited works used an RNN model for modeling sequential data. However, we argue that using Transformer architecture [7] in speech translation might be more suitable than RNN. Inaguma et al. reported that a Transformer-based model outperformed an RNN-based model on end-to-end ASR, MT, and TTS tasks [8]. To translate speech-to-speech, the model needs to process long sequential data based on the long context information of the source and target speech. The Transformer can reduce the calculation time, especially when a data sequence is long since it lacks a recurrent function. Furthermore, since the Transformer uses a self-attention function to find related information from the whole sequence, it can learn long context information [7]. Unfortunately, no work applies a Transformer to end-to-end speech-to-speech translation tasks.

We address these remaining issues by performing the following:

1. constructing the step-by-step scheme to complete end-to-end speech-to-speech translation;
2. revamping the RNN-based speech-to-speech translation model with the Transformer;
3. analyzing the model's behavior and comparing the translation performance with the traditional cascade and the state-of-the-art multi-task frameworks on syntactically similar and distant language pairs.

Our proposed model translates in a speech-to-speech, step-by-step process to reduce complexity. Then it transfers attention results instead of its output text and performs total optimization. To the best of our knowledge, this is the first work that develops an end-to-end Transformer-based speech-to-speech translation system. It is also the first one that investigates the performance of syntactically distant language pairs. For comparison, we also reconstruct the recent RNN-based multi-task speech-to-speech model [4] as well as build the Transformer version, and compare the performances in syntactically similar and distant language pairs.

2. PROPOSED METHOD

Constructing a direct speech-to-speech translation task for single attention-based encoder-decoder architecture is difficult because the model simultaneously needs to solve three complex problems:

1. learning how to process long speech sequences to map them to the corresponding text, similar to the issues addressed in ASR [9];
2. learning how to make proper alignment rules between the source and target languages, similar to the issues addressed in MT [10, 11];
3. learning how to generate long speech sequences from the corresponding text, similar to the issues addressed in TTS [12].

Furthermore, the model requires a large amount of parallel speech that is often unavailable.

In the traditional approach, the above three problems are respectively handled by ASR, MT, and TTS, which were trained and tuned independently. This approach reduces the complexity and the need for a large amount of parallel speech. Unfortunately, since these components are trained separately, MT often struggles to handle ASR errors, resulting in unnatural translation results. In this proposed work, since we still separately handle these three problems, the model does not require a large amount of parallel speech. In contrast with the traditional approach, those problems are not addressed by completely separate components that are trained and tuned independently.

Figure 1 illustrates the overall framework of our proposed end-to-end speech-to-speech translation architecture. We trained the model step-by-step with curriculum learning from easy to complicated tasks while changing the model structures. First, we trained an attention-based encoder-decoder component for each problem task and gradually progressed to a more difficult target task (i.e., speech-to-text and speech-to-speech translation tasks) by connecting these components to the Transcoder network [13]. In this case, the learning scheme changes from single-task to multi-task learning by simultaneously training the decoders and the Transcoders. The overall architecture has a single source-language-speech encoder, three decoders that predict source-language text transcriptions, target-language-text transcriptions, target-language speech, and two Transcoders. The first Transcoder transfers the attention context information of the acoustic hidden representations to the linguistic hidden representations, and the second transfers the attention context information of the linguistic hidden representations to the acoustic hidden representations of the target language. Further details are described below.

2.1. Training process

First, we prepared pre-trained ASR, MT, and TTS models. After that, we utilized the pre-trained ASR encoder for the source-language-speech encoding, and the ASR, MT, and TTS decoders for the source-language-text, target-language-text, and target-language-speech generation, respectively. Then we fine-tuned the overall framework by connecting these components with two Transcoders. Various studies have described how to use pre-trained speech recognition and machine translation to initialize speech translation models [5, 14, 15, 3]. However, the hidden representation of a pre-trained ASR and an MT encoder is very different. First, the lengths of the ASR and MT input sequences are different. Second, ASR's hidden states represent the input speech's phonological information necessary for transcription, while the MT hidden states represent bilingual semantic information for translation. Therefore, this gap affects the MT decoder's tuning by connecting the pre-trained ASR encoder and pre-trained MT decoder. To avoid this problem, we use a

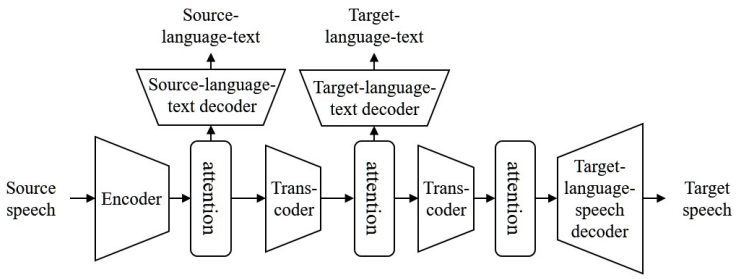

Fig. 1. Proposed framework of the end-to-end speech-to-speech translation architectures

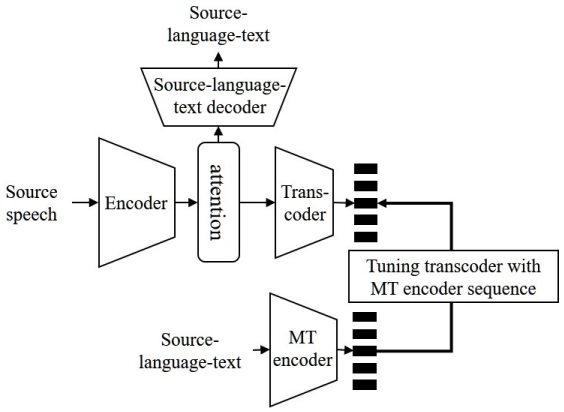

Fig. 2. Transcoder training process

Transcoder to transfer the ASR hidden representation closer to the MT hidden representation for the pre-trained MT decoder. In the first Transcoder training, we make a Transcoder with the same architecture of the pre-trained MT encoder *Encoder*$^{\text{MT}}$. However the Transcoder has a feed-forward network instead of an *Encoder*$^{\text{MT}}$. We fine-tuned the pre-trained ASR decoder and trained the Transcoder to output the source language's linguistic hidden representation by transfer learning, where the pre-trained MT encoder is treated as a teacher (Fig. 2). The Transcoder received weighted contextual information A^{ASR} from the ASR attention module of the ASR decoder and generated Transcoder output H^{TC}. In this step, we used an attention mechanism to align the ASR speech sequence to the text sequence. After that, we used MT encoder hidden states H^{s} as a target to optimize the Transcoder to bring the ASR hidden representations closer to the MT representations:

$$\begin{aligned} H^{\text{TC}} &= \textit{Transcoder}(A^{\text{ASR}}), \\ H^{\text{S}} &= \textit{Encoder}^{\text{MT}}(S). \end{aligned} \tag{1}$$

In this way, the pre-trained MT decoder can attend to the pre-trained ASR contextual information of the source-language-acoustic hidden representation. Here S is a source-language-text sentence. The length of the H^{TC} and H^{t} sequences equals the source text length. We froze other unit parameters during this Transcoder training and only updated the Transcoder and ASR decoder parameters. We thoroughly optimized the Transcoder to minimize the smooth L1 loss between H^{TC} and H^{S}:

$$loss(H^{\text{TC}}, H^{\text{S}}) = \begin{cases} 0.5 * (H^{\text{TC}} - H^{\text{S}})^2, \\ \quad if|H^{\text{TC}} - H^{\text{S}}| < 1, \\ |H_l^{\text{TC}} - H^{\text{S}}| - 0.5, \\ \quad otherwise. \end{cases} \tag{2}$$

Here l denotes the source-language text sequence's index. When the transcoding loss is below a threshold value (e.g., 0.05), we connect the Transcoder with a pre-trained MT decoder and start training for speech-to-text translation.

In the second Transcoder training, we train the second Transcoder using the pre-trained TTS encoder as a teacher model, with the same process as the first translation training process (Fig. 2). Using the same training mechanism as for the first Transcoder, we trained the second Transcoder until the smooth L1 loss fell below a threshold. After that, we connected the trained Transcoder with the pre-trained TTS decoder and performed a total optimization for all the parameters of all the components: one encoder, three decoders, and two Transcoders.

2.2. Inference process

First, we performed an ASR to generate ASR attention results $A^{\text{ASR}} = [a_1^{\text{ASR}}, \ldots, a_l^{\text{ASR}}, \ldots, a_L^{\text{ASR}}]$ for the first Transcoder. Here L and l denote the length and index of source-language text sequence S:

$$\begin{aligned} H^{\text{query}} &= \textit{Encoder}(X), \\ h_{l-1}^{key} &= \textit{Decoder}^{key}(s_{l-1}), \\ a_l &= \textit{Attention}(H^{query}, h_{l-1}^{key}), \\ s_l &= \textit{Decoder}^{out}(a_l^{\text{MT}}, h_{l-1}^{key}). \end{aligned} \tag{3}$$

ASR attention results A^{ASR}, which are used in the next speech-to-text translation step, are ASR encoder hidden states and the attention weight's dot products. The Transformer model has two decoding steps. The $Decoder^{key}$ generates key-value $H^{key} = [h_1, \ldots, h_l, \ldots, h_L]$ for the attention module from the previous target. The $Decoder^{out}$ generates a source text token from attention results $A^{\text{MT}} = [a_1, \ldots, a_l, \ldots, a_L]$ and key-value h_{l-1}^{key}. The $Attention$

function is a multi-head attention. s_l and a_l denote the source text and attention result at the $l-1$ step. In this step, we generate source text sequence $S = [s_1, \ldots, s_L]$ and attention results A^{ASR} by performing autoregressive decoding. We only use the attention results for the next translation process. The generated source text sequence is only used for this autoregressive decoding. Next we utilized a Transcoder and a target-language-decoder to translate the source-language-speech to target-language-text sequence $T = [t_1, \ldots, t_M]$:

$$\begin{aligned} H^{query} &= Transcoder^{\text{1st}}(A^{\text{ASR}}), \\ t_m, a_m &= Decoder^{MT}(t_{m-1}). \end{aligned} \tag{4}$$

Here M and m denote target-language-text sequence T's length and index. A^{ASR} is the acoustic representation of the source-language-text, and H^{query} is its linguistic representation. The first Transcoder transfers the acoustic hidden representations from the ASR attention results to the linguistic hidden representations of the MT encoder hidden states. The MT target-language-text decoder attends to linguistic hidden representations H^{query} to generate target-language-text $T = [t_1, \ldots, t_L]$ and target language attention results $A^{\text{MT}} = [a_1, \ldots, a_L]$. We only use attention results A^{MT} for the next translation process. The ASR attention results of A^{ASR} are a filtered feature of a source-language-speech encoder sequence. This step performs a direct translation from the source-language-speech to the target-language-text:

$$\begin{aligned} H^{query} &= Transcoder^{\text{2nd}}(A^{\text{MT}}), \\ t_n, a_n &= Decoder^{TTS}(y_{n-1}). \end{aligned} \tag{5}$$

Finally, we generate target-language-speech feature sequence $Y = [y_1, \ldots, y_n, \ldots, y_N]$ using the second Transcoder with target-language-text decoder attention results $A^{\text{MT}} = [a_0^{\text{MT}}, \ldots, a_m^{\text{MT}}, a_M^{\text{MT}}]$, and a target-language-speech decoder. Here M and m denote target-language-speech sequence Y's length and index. Input sequence A^{MT} is a linguistic representation of the target-language-text sequence. We used the second Transcoder to map the linguistic hidden representations from MT attention results A^{MT} to the acoustic hidden representations of the TTS encoder hidden states. Here we do not need to consider the source and target language alignments.

In summary, we solved the speech-to-speech translation task by predicting the target speech spectrograms as well as the source and target text transcriptions from the first and second decoders. This result resembles the auxiliary tasks in previous works. The main difference is that the previous multi-task system only used one attention module to align the input speech to the target speech. If the translation task becomes too difficult, then the translation performance will fall significantly. In contrast, our proposed system relies on three attention modules that focus on the input-output alignment of these specific problems: (1) speech-to-text in the source language, (2) text-to-text in the source and target languages, (3) text-to-speech in the target language.

3. EXPERIMENTS

3.1. Experimental setup

Table 1. Model setting

Transformer setting	
Encoder layers	3
Decoder layers	6
Multi-head	8
Transformer hidden size	256
Transformer FFN [7] hidden size	1024
RNN setting	
Encoder layers	4
Decoder layers	2
RNN type	LSTM
Multi-head	8
RNN hidden size	256
Speech input and output layer setting	
Speech input size	80 (mel) * 3 (frames)
Speech mel out size	80 (mel) * 5 (frames)
Optimizer setting	
Warm-up steps for Transformer	8000
Optimizer method	Adam
Learning rate decay	0.8
Training steps	2,000,000
Batch size (ASR / MT / TTS / ST)	64 / 128 / 32 / 16

We conducted our experiments using a basic travel expression corpus (BTEC) [16, 17]. We chose English-to-Spanish and Japanese-to-Korean as syntactically similar language pairs as well as English-to-Japanese and Japanese-to-English as syntactically distant language pairs. The BTEC English-Japanese parallel text corpus consisted of 480-k training data. The BTEC English-Spanish and Japanese-Korean parallel text corpus consisted of 160-k training data. Since the corresponding speech utterances for this text corpus are unavailable, we used the Google text-to-speech synthesis[1] to generate a speech corpus. We also utilized the BTEC corpus that consists of 190-k utterances of natural English speech and 140-k utterances of natural Japanese speech. However, it only has 5-k speech-to-speech parallel data of English-Spanish and English-Japanese and 7-k speech-to-speech parallel data of Japanese-Korean and Japanese-English. During a training step, the input was both natural and generated speech, and the target was generated speech. At the test step, the input was natural speech only. We segmented the speech utterances into multiple frames with a 50-ms window and 12-ms steps and extracted 80-dimension Mel-spectrogram features using LibROSA[2]. We concatenated 3 frames of the acoustic features into one super vector for the input and output speech. We describe the model setting in Table 1. We

[1]Google TTS: https://pypi.python.org/pypi/gTTS

[2]LibROSA: https://librosa.github.io/librosa/

Table 2. ASR WER

Model	Natural English	Natural Japanese	Generated English	Generated Japanese
RNN	9.1	10.3	-	-
Transformer	6.8	8.2	3.5	5.1

Table 3. BLEU and METEOR scores of text-to-text translation

Model	Syntactic similar				Syntactic distant			
	En to ES		Ja to Ko		En to Ja		Ja to En	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
RNN	45.9	62.1	46.3	63.8	42.0	58.4	43.4	59.8
Transformer	47.1	66.4	48.2	67.3	43.2	59.8	45.1	61.0

Table 4. BLEU and METEOR scores of speech-to-speech translation

Model	Syntactic similar				Syntactic distant			
	En to Es		Ja to Ko		En to Ja		Ja to En	
	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
Baseline: Cascade (RNN)	38.9	47.7	38.7	49.1	32.5	44.2	32.0	43.2
Baseline: Cascade (Transformer)	41.3	52.1	41.0	51.1	34.1	45.2	35.0	45.3
Google (RNN) [4]	38.8	48.2	39.1	49.9	33.2	45.5	34.2	45.0
Google (Transformer)[1]	43.1	58.8	42.5	58.3	36.9	52.6	38.3	48.4
Transcoder (Transformer)	44.0	59.3	42.9	58.8	40.6	56.6	41.0	55.8

[1] In this experiment we constructed a Google system using a Transformer network.

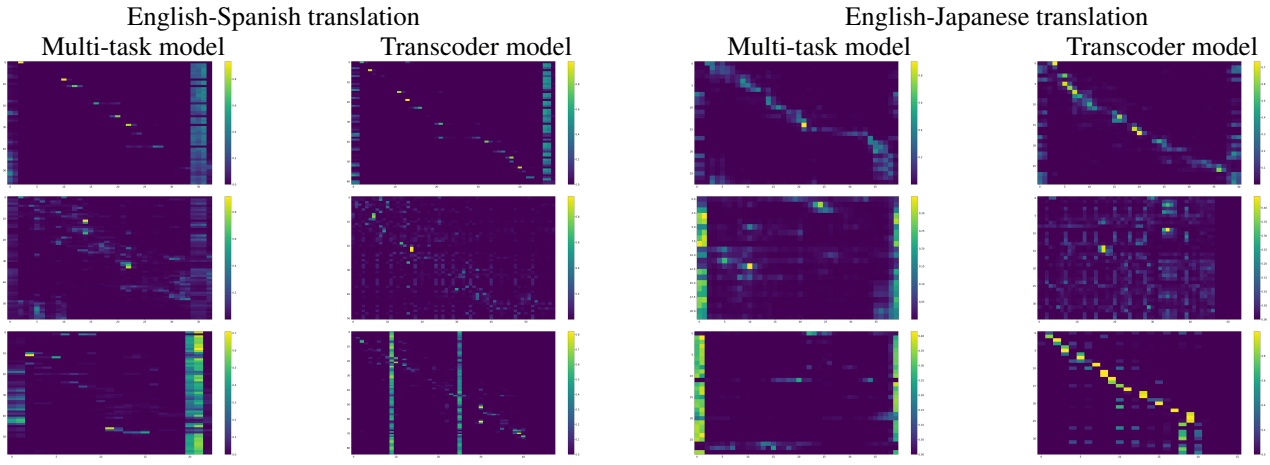

Fig. 3. Attention table of multi-task and Transcoder speech translation: Source-language-text, target-language-text, and target-language-speech attention tables are arranged from top to down.

used OpenNMT[3] to make a multi-task-based Google model and implemented our proposed model on it.

3.2. Experiment results

We translated the input speech to the target language speech using our proposed Transcoder-based models. As we mentioned earlier, for comparison, we reproduced the previous Google's speech translation system with RNN [4] and also built the Transformer version. To evaluate the speech outputs' translation performance, we transcribed the speech using an ASR system that was trained with TTS-generated acoustic features and calculated the BLEU [18] and METEOR [19] scores from the transcription to evaluate each model's performance. First, we show the RNN and Transformer-based ASR and MT performances in Tables 2 and 3. From these works, we found that by training and tuning all components, the system could learn how to handle the error propagation from each component task and optimize overall performance. Thus the end-to-end models outperformed the cascade models. Thus the end-to-end models outperformed the cascade models. Table 4 lists the translation performances. First, we describe why the end-to-end frameworks outperformed the cascade models. Several works show that end-to-end speech-to-text translation can sometimes outperform the cascade model. Sperber et al. [6] and Osamura et al. [20] concluded that dataset size and ASR error rate are important factors to improve the end-to-end model and outperform the cascade model. From these works, we achieved good ASR performance and prepared enough parallel datasets using the synthesized data. Thus the end-to-end models outperformed the cascade models.

Second, we explain why our Transcoder model outperformed Google's multi-task-based speech translation model. The multi-task model shared the same encoder sequence for transcription and translation. In this framework, the encoder needs to move the source encoder hidden states closer to both the source-text and target-text hidden vectors to get attentions. It is possible in syntactically similar language translation since both general transcription and syntactically similar language translation do not require long context information. In syntactically similar language translation, since there are many one-to-one mappings in translation, the speech translation only needs to change the target side word id.

However, syntactically distant language translation needs to consider long context information and long distant multi-to-multi word mapping. Examining long context information at the encoder is difficult because the input speech sequence is long. When the translation task becomes difficult, the multi-task encoder is challenging. Thus the multi-task model performances drop on syntactically distant language pairs. On the other hand, the Transcoder helps address the long context and complex mapping problem that is unnecessary for transcription. This process does not affect the source encoder performance and provides memory sequences that are essential for translation. Therefore, the Transcoder outperformed Google's multi-task-based framework, especially on syntactically distant language translation tasks.

For more discussion, we analyzed the attention map and the model behavior. Fig. 3 also shows the attention matrices of three decoders that generate (1) the text of the source language in the top row, (2) the text of the target language in the middle row, and (3) the speech of the target language in the bottom row. Although the multi-task-based model only used single attention and a decoder during the inferences, it still had three individual attention modules that had been trained for those three tasks. On syntactically similar language translations, both the multi-task and Transcoder models have a similar monotonic shape attention. However, on syntactically distant language translations, the proposed Transcoder model retains a monotonic shape attention for the first and third tasks, although the multi-task model does not. This is because, in the multi-task-based speech translation system, all the decoders share the same encoder states, and thus the attention model provided the information of the (1) speech-to-text of source language, (2) the speech-to-text from the source to the target language, and (3) the speech-to-speech from the source to the target language. On the other hand, since the Transcoder-based speech translation solved the problem sequentially, the attention provided the information of the (1) speech-to-text of the source language, (2) the text-to-text from the source to the target language, and (3) the text-to-speech of the target language. Since speech-to-speech translation is very challenging, our proposed approach, which solves the problem by breaking it into a sequence of sub-tasks, worked effectively and outperformed the multi-task-based speech translation.

4. CONCLUSION

We proposed a Transformer-based Transcoder network for end-to-end speech-to-speech translation and applied our proposed framework to various language pairs, including syntactically similar and distant language pairs. We compared our Transcoder-based model with the state-of-the-art, end-to-end speech-to-speech translation model that was trained based on a multi-task scheme. Our results revealed that the proposed model's translation performance surpassed the state-of-the-art model in all the language pairs. In the future, we will further investigate the performance of our proposed model in other language pairs using completely natural speech-to-speech translation corpora.

5. ACKNOWLEDGEMENTS

Part of this work was supported by JSPS KAKENHI Grant Number JP17H06101.

[3]OpenNMT: http://opennmt.net/

6. REFERENCES

[1] Satoshi Nakamura, Konstantin Markov, Hiromi Nakaiwa, Gen-ichiro Kikui, Hisashi Kawai, Takatoshi Jitsuhiro, Jin-Song Zhang, Hirofumi Yamamoto, Eiichiro Sumita, and Seiichi Yamamoto, "The ATR multilingual speech-to-speech translation system," *IEEE Transaction Audio, Speech & Language Processing*, vol. 14, no. 2, pp. 365–376, 2006.

[2] Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," *CoRR*, vol. abs/1612.01744, 2016.

[3] Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin, "End-to-end automatic speech translation of audiobooks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2018, pp. 6224–6228.

[4] 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 *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*, 2019, pp. 1123–1127.

[5] Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura, "Structured-based curriculum learning for end-to-end english-japanese speech translation," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, 2017, pp. 2630–2634.

[6] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel, "Attention-passing models for robust and data-efficient end-to-end speech translation," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 313–325, 2019.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017, pp. 5998–6008.

[8] Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Yalta, Tomoki Hayashi, and Shinji Watanabe, "ESPnet-ST: All-in-one speech translation toolkit," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL*. 2020, pp. 302–311, Association for Computational Linguistics.

[9] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, 2015, pp. 577–585.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, 2015.

[11] Philipp Koehn, Franz Josef Och, and Daniel Marcu, "Statistical phrase-based translation," in *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL*, 2003.

[12] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, 2017, pp. 4006–4010.

[13] Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura, "End-to-end speech translation with transcoding by multi-task learning for distant language pairs," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 1342–1355, 2020.

[14] Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen, "Sequence-to-sequence models can directly translate foreign speech," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, 2017, pp. 2625–2629.

[15] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel, "Neural lattice-to-sequence models for uncertain inputs," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 2017, pp. 1380–1389.

[16] Gen-ichiro Kikui, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto, "Creating corpora for speech-to-speech translation," in *EUROSPEECH 2003 - INTERSPEECH 2003, 8th European Conference on Speech Communication and Technology,*, 2003.

[17] Gen-ichiro Kikui, Seiichi Yamamoto, Toshiyuki Takezawa, and Eiichiro Sumita, "Comparative study on corpora for speech translation," *IEEE Transaction Audio, Speech & Language Processing*, vol. 14, no. 5, pp. 1674–1682, 2006.

[18] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[19] Michael J. Denkowski and Alon Lavie, "Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems," in *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT*. 2011, pp. 85–91, Association for Computational Linguistics.

[20] Kaho Osamura, Takatomo Kano, Sakriani Sakti, Katsuhito Sudoh, and Satoshi Nakamura, "Using spoken word posterior features in neural machine translation," in *Proceedings of the 15th International Conference on Spoken Language Translation, IWSLT*, 2018, vol. 21, p. 22.