Doctoral Dissertation

Direct End-to-End Speech Translation for Distant Language Pairs

Takatomo Kano

March 13, 2020

Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Takatomo Kano

Thesis Committee:

Professor Satoshi Nakamura	(Supervisor)
Professor Yuji Matsumoto	(Co-supervisor)
Professor Tsukasa Ogasawara	(Co-supervisor)
Professor Laurent Besacier	(University Grenoble Alpes)
Associate Professor Sakriani Sakti	(Co-reviewer)

Direct End-to-End Speech Translation for Distant Language Pairs*

Takatomo Kano

Abstract

Directly translating spoken utterances from a source language to a target language is challenging because it requires a fundamental transformation of both linguistic and para/non-linguistic features. Traditional speech-to-speech translation approaches concatenate automatic speech recognition (ASR), text-to-text machine translation (MT), and text-to-speech synthesis (TTS) through text information. The traditional speech translation performance is worse than that of the MT because the translation results are affected by ASR errors. An end-to-end speech translation system has the potential to recover from ASR errors and achieve higher performance than that of traditional speech translation systems.

The current state-of-the-art models for ASR, MT, and TTS have mainly been built using deep neural networks. More specifically, they use attention-based encoder-decoder neural networks with an attention mechanism. Recently, several works have constructed end-to-end direct speech-to-text translation by combining ASR, MT, and TTS into a single model. However, the usefulness of these models has only been investigated on language pairs of similar syntax and word order (e.g., English-French or English-Spanish). For syntactically distant language pairs (e.g., English-Japanese), speech translation requires distant word reordering.

This thesis addresses how to build a speech translation system for syntactically distant language pairs that suffer from long-distance word reordering. I focus mainly on English (subject-verb-object (SVO) word order) and Japanese ((SOV) word order) language pair. First, I propose a speech translation model that

*Doctoral Dissertation, Graduate School of Information Science, Nara Institute of Science and Technology, March 13, 2020.

does not require significant changes in the cascaded ASR and MT structure. Specifically, I construct a neural network model that passes all ASR candidate scores to the MT module. The MT can then consider the ASR hypothesis in the translation process. Therefore the MT model can learn how to recover from ASR errors during translation. I demonstrate how the acoustic information helps to recover from ASR errors and improves translation quality.

Next, I propose a first attempt to build an end-to-end speech translation system for syntactically distant language pairs that suffer from long-distance reordering. To guide the encoder-decoder attention-based model for this challenging problem, I construct an end-to-end speech-to-text translation module using a transcoder and Curriculum Learning (CL) strategies that gradually train the network for the end-to-end speech translation tasks by adapting the decoder or encoder parts.

I then focus on the text-to-speech translation task and apply speech information to the target text decoding process. My proposed approach shows that speech information helps target text generation, and the generated results are much closer to the reference sentence.

Finally, I propose a complete end-to-end speech-to-speech translation system and compare its performance with the current state-of-the-art end-to-end speech-to-speech translation system. My experiment results show that the proposed approach provides significant improvements in comparison with the baseline end-to-end speech translation models.

Keywords:

Speech Recognition, Machine Translation, Text-to-Speech Synthesis, Speech Translation, Deep Neural Network

Contents

List of Figures

List of Tables

1. Introduction

1.1 Breaking Language Barriers: Speech Translation Technology Overview

As globalization expands, many people go abroad for business, education, or sightseeing. Countries become more dependent on each other and pay attention to their relationships. Many companies conduct business all over the world. While this globalization has made international exchange essential, language barriers remain notorious obstacles to free communication. Therefore, people are required to learn a *lingua franca*. In our current global economy, this free language is English. However, speaking fluent English is difficult for non-native English speakers of distant languages. Because their pronunciation and listening ability is rooted in the phonetic inventory of their native language, and languages vary wildly with regard to their phonetics and other linguistic properties, learning a distant language requires a lot of effort.

Speech translation is an innovative technology that enables people to communicate with speakers of different languages. Ideally, it will translate the input speech to target language speech automatically with little error. A user can communicate with foreigners using their own language using such speech translation systems. Speech translation technologies have developed and grown very rapidly. The first rule-based speech translation systems were proposed in the 1980s, operating on a limited small vocabulary. Today, state-of-the-art speech translation systems are built using deep neural network models trained on large vocabulary datasets across many domains. Recently many companies have started to provide translation services and devices to consumers for use when traveling abroad.

1.2 Speech Translation Technology Limitations

The speech translation task is challenging. Human speech signals consist of both linguistic and paralinguistic information. Therefore speech translation is much more difficult than text translation. The traditional speech translation system solves a speech translation task using concatenated automatic speech recognition (ASR), text-to-text machine translation (MT), and text-to-speech synthesis (TTS) via text information as shown in Fig. 1.

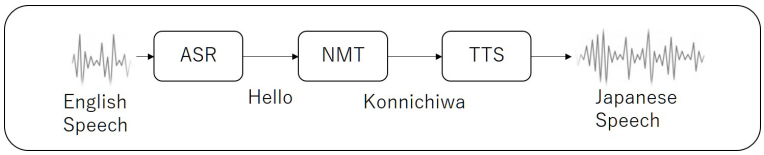


Figure 1. Traditional speech-to-speech translation system overview.

There are several problems when training these traditional speech translation systems. The most common problem is ASR errors that affect the MT process [12]. While ASR models are trained to minimize word error rates, the model will sometimes output incorrect words that have a similar pronunciation. Therefore speech translation performance is usually lower than that of text translation. The MT models map source sentences to target sentences considering their respective contents. However, the model will sometimes output the correct word at an incorrect position, or a semantically similar word. Usually, the MT module is trained individually without considering potential ASR errors. Therefore the model will not be able to properly handle input sentences that feature ASR errors and will be unable to recover from these errors during the translation process.

As for the data itself, the collection and alignment of parallel speech corpora are very costly. Add to this the fact that over half of the world's languages are *only* spoken and do not have a written form. This means that constructing speech translation that heavily relies on parallel texts can be very difficult.

Finally, the speech signal generally involves both linguistic and paralinguistic information (e.g., rhythm, emphasis, or expression of an emotional state). This paralinguistic information is generally not encoded in written communication, and consequently is lost in the ASR process. This means the MT ends up translating only linguistic information. Some studies have proposed methods that

include additional components in order to handle paralinguistic translation, but this introduces additional complexity and delay [17, 9, 1]. Therefore, we require an architecture that can translate source speech to target speech without using text as an intermediary.

1.3 Related Works

Deep neural networks (DNN) have shown remarkable performance on many tasks. E.g. sequence-to-sequence attention-based encoder-decoder networks can learn powerful models for ASR, MT, and TTS [8, 3, 36]. Several recent works have attempted to build end-to-end direct speech-to-text translation systems that perform a combination of ASR and MT tasks using a single model. Duong et al. published the first study to attempt speech-to-text translation with DNN, in which they proposed alignment and translation re-ranking directly from source language speech with the target text translations[11]. However, their chosen language pair was Spanish and English, which has similar syntax and word order (SVO), and the results failed to outperform the traditional cascade approach based only on statistical word-level MT (MOSES) [21]. Their proposed attention-based model achieved a BLEU score of 14.6 [27], where the MOSES baselines had much better scores ranging between 18.2 and 20.2.

Later, Berard et al. attempted to build a full-fledged end-to-end speech-to-text translation system [6], which is also the first work that uses speech generated by TTS for data augmentation. However, they only compared performance with statistical text-based MT systems. They also used another close language pair, English and French, which both use SVO word order. For such language pairs, only local Berard is sufficient for translation. Berard et al. [5] further investigated speech-to-text translation using various units of speech, such as characters, sub-words, and words. They applied beam search, greedy search, as well as an ensemble search that successfully improved translation accuracy. These experiments also concentrated on English and French, using both natural and generated speech data.

Weiss et al. focused on Spanish-English speech-to-text translation. The authors proposed parameter sharing of an ASR encoder and a speech translation model's encoder. Their study revealed that an encoder could learn to transform

speech into a consistent interlingual sub-word unit representation, which the respective decoders were able to assemble into phrases in either language [37]. Bansal et al. then performed speech translation with natural speeches from multiple speakers and used unsupervised term discovery to cluster repeated patterns in the audio to create pseudo-text instead of performing ASR [4]. Ultimately, all of the previous research presented above still focused on syntactically similar language pairs.

1.4 Contribution of this Research

In this thesis, I propose four different speech translation architectures.

First, speech translation performance is usually affected by ASR errors; therefore, many speech translation systems focus on recovering from ASR errors in the translation process, and on getting closer to text-to-text MT performance. Human speech carries more information than text. Therefore, in my proposed end-to-end speech translation model, I develop a system that not only recovers from ASR errors during translation but also aims to outperform text-based MT systems. I find that end-to-end speech translation has the potential to outperform text-based MT systems, and I investigate how the speech information is utilized in the translation process through further analysis.

The second model focuses on an end-to-end speech-to-text translation. The model translates input source speech to target text directly. All related works performed the direct translation on syntactically similar language pairs. In syntactically similar language pairs, the attention shape in the translation is known to be monotonic. Then the model can translate the input source sequence one by one without drastic reordering. On the other hand, translating syntactically distant languages results in much more complex attention shapes. There are many deletions and insertions, frequent reordering, as well as many-to-one and one-to-many word alignments. The model also needs to learn dependencies for long input sequences. In this research, I attempt to build an end-to-end speech translation for syntactically distant language pairs. The proposed model uses a deeper architecture than standard attention-based encoder-decoder translation models. Therefore training the model is a challenging problem. I propose learning strategies that gradually train the network for end-to-end speech translation tasks by adapting each part of the model one by one.

In the third model, I first focus on the end-to-end text-to-speech translation. Much of the existing work focuses on the speech-to-text translation, and on improving the translation performance over traditional speech translation. However, there is no work that focuses on text-to-speech translation, and on outperforming text-to-text translation performance. My proposed model targets the text-to-speech task, using speech information during the decoding to target text. This speech information helps to generate a more accurate target sequence. Finally, my

proposed model outperformed the state-of-the-art text-to-text translation model.

Finally, I propose an end-to-end speech-to-speech translation model. I compare my proposed model to the state-of-the-art end-to-end speech translation model for syntactically distant language pairs. My proposed model outperforms the-state-of-the-art model with regard to translation performance, and I demonstrate that the internal translation process differs from that of the state-of-the-art model. We, therefore, show that we can solve the complex speech translation task by integrating the individual modules of a traditional cascaded translation system into a single end-to-end framework.

1.5 Thesis Outline

In this thesis, I describe the general sequential models in chapter 2. In this section, I introduce the Recurrent Neural Network (RNN), sequence-to-sequence model, attention-based encoder-decoder model, and transformer model. In chapter 3, I describe the traditional cascade speech translation architecture and introduce the end-to-end ASR, MT, and TTS models. Moreover, I mention the cascade speech translation problems. In chapter 4, I introduce end-to-end speech-to-text translation researches and compare models with my proposed model. I also perform analysis of ASR error recovery mechanisms and find out speech translation benefits. In chapter 5, I focus on the text-to-speech translation tasks and propose a model that can utilize speech information in the target language decoding. In chapter 6, I compare the state-of-the-art speech-to-speech translation model with my proposed model and claim the prerogative of my work. Finally, I summarise and discuss my work in chapter 7.

2. Neural Sequential Modeling

2.1 Sequential Data

2.1.1 Text Data

In neural network modeling, a word is represented by its index in a one-hot vector representation of the vocabulary $s = [0, 0, 1, ..., 0]$, where the vector size is identical to the vocabulary size, and the values of the word index dimension are 1 or 0. A sentence is represented as a sequence of word one-hot vectors $\mathbf{s} = [s_1, ..., x_N]$, with s_1 being the first word, and N the length of sentence. To handle a sentence, a model processes the word one-hot vectors from 1 to N, not only extracting the meaning of each word but also including word dependencies. Therefore the model needs to memorize a sequence of words.

2.1.2 Speech Data

Speech data is represented as acoustic waveforms and generally processed in such a way that the signal is transformed into a spectrogram that represents signal power in various frequency bands. For this, the speech is windowed every few milliseconds (ms), and the spectrogram is extracted for each of these windows. The spectrogram feature is a vector, with each dimension representing a frequency band. In this manner, the speech can be represented as $\mathbf{x} = [x_1, ..., x_I]$, where the x_1 is the first window of the speech signal, and I is the number of windows in the signal. Speech data is not stable since it is affected by factors such as environmental noise or individual speakers. This is one reason why speech translation is more complicated than text-to-text MT.

2.2 Sequential Modeling with Neural Network

2.2.1 Overview

Feed-forward Neural Networks (FNN) have been used for a long time. Often used for standard pattern classification tasks, they are less applicable to the modeling of sequential data, as the network should be able to take into account past inputs and make use of long term context information. Since FNN can only consider

the current input and output, this makes processing sequential data with them difficult. Recurrent Neural Networks (RNN), on the other hand, forward their last hidden state to the current step as following:

$$h_n = \phi(W^x x_n + W^h h_{n-1}) \tag{1}$$

Here, W^x is a weight vector that maps the current input x_n, and W^h is the weight vector for the previous hidden state. The ϕ denotes a non-linear activation function, e.g., the *sigmoid* function. Long Short-Term Memory (LSTM) networks are an extension of RNNs. These networks can remember context over longer stretches of time than regular RNNs. LSTM cells have three gates the input gate, output gate, and forget gate, as well as a cell state. These are updated for each step n as follows:

$$\begin{aligned} i_n &= \phi(W^{ix} x_n + W^{ih} h_{n-1} + b^i), \\ f_n &= \phi(W^{fx} x_n + W^{fh} h_{n-1} + b^f), \\ o_n &= \phi(W^{ox} x_n + W^{oh} h_{n-1} + b^o), \\ c_n &= f_n \circ c_{n-1} + i_n \circ \phi(W^{cx} x_n + W^{ch} h_{n-1} + b^c), \\ h_n &= o_n \circ \phi(c_n). \end{aligned} \tag{2}$$

Here, i_n, f_n, and o_n refer to output vectors of the input gate, forget gate, and output gate. W^* and b^* denote a weight vector and bias for the gates and the cell state. The $\circ$ denotes a Hadamard product function. The three gates use a *sigmoid* function as activation function ϕ, while the cell and hidden state use *tanh*.

The LSTM can memorize the context information using its memory cell. However, since these gates each have their own set of weight parameters, the training will be much more expensive compared to a basic RNN. Cho et al. simplified this LSTM design, creating the Gated Recurrent Unit (GRU) [7]. They merged the forget and output gates into a single update gate, and removed the cell state in order to reduce the number of parameter of the neural net:

$$\begin{aligned} i_n &= \phi(W^{ix} x_n + W^{ih} h_{n-1} + b^i), \\ g_n &= \phi(W^{gx} x_n + W^{gh} h_{n-1} + b^g), \\ h_n &= (1 - i_n) \circ h_{n-1} + i_n \circ \phi(W^{hx} x_n + W^{hh} h_{n-1} + b^h). \end{aligned} \tag{3}$$

Here g_n is the output vector of the update gate, where W^{gx}, W^{gh}, b^g are the weight vectors for the input gate $\mathbf{x}$, the weight vector for the previous hidden state h_{n-1}, and the bias of update gate, respectively. The performance of GRUs is equivalent to that of LSTMs for speech modeling tasks. Moreover, the GRU can successfully train on smaller datasets than an LSTM. On the other hand, the LSTM is better at taking long-distance context information into account. For this reason, GRUs are less suited to handle language modeling than LSTMs.

2.2.2 Sequence-to-Sequence Transformation

The first sequence to sequence transformation was done by Sutskever et al. [31]. They transformed a given input sequence $\mathbf{x}$ to a target sequence $\mathbf{y}$ using RNNs, as shown in Fig. 2. The sequential model has two RNN components, referred to

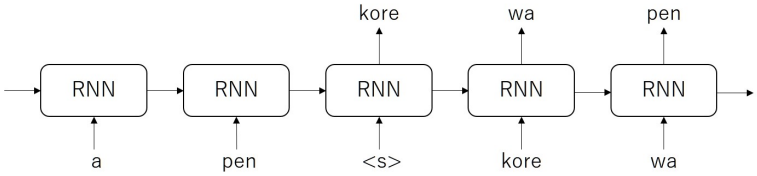


Figure 2. Sequence to sequence translation overview.

as encoder and decoder. Given a sequence of inputs $\mathbf{x} = [x_1, ..., x_N]$, the encoder computes a sequence of hidden states $\mathbf{h} = [h_1, ..., h_N]$ via the following equation:

$$h_n = RNN(x_n, h_{n-1}) \tag{4}$$

The initial hidden state h_0 is given as a zero-vector. Then the decoder computes a target sequence $\mathbf{y} = [y_1, ..., y_M]$ using the last hidden state h_N generated by the encoder as follows:

$$y_m = RNN(y_{m-1}, h_{m-1}) \tag{5}$$

For the decoder, the initial hidden state is the last encoder state h_N. The first decoder input y_0 is a unique token that represents the start of the target sequence. The encoder-decoder model can easily map sequences to sequences without considering the alignment between the inputs and outputs. However, the encoder-decoder model has difficulty when it comes to handling long-distance

context information. There are two issues when modeling this kind of context. One is the encoder's inability to properly memorize long term dependencies. The RNN processes an input sequence and passes on a hidden vector that contains past context information. This means that information from the beginning of the input sequence will fade away with each encoder step. Another problem is that the encoder and decoder are only sharing the last hidden vector. Therefore the encoder has to map all given information into a single vector, and the decoder needs to decode all targets from it. The encoder-decoder model is difficult to handle long sequential data due to these two limitations.

2.2.3 Attention-based Encoder-Decoder Model

To solve the problem of sequence-to-sequence transformation with long-distance context information, Long et al. have proposed an attention-based encoder-decoder model that consists of an encoder, a decoder, as well as an attention module [25]. Given an input sequence $\mathbf{x} = [x_1, x_2, ..., x_N]$ with length N, the encoder produces a sequence of vector representations $h^x = [h_1^x, h_2^x, ..., h_N^x]$, using a bidirectional LSTM (Bi-LSTM) [15]. The forward LSTM reads the input sequence from x_1 to x_N and estimates forward $\overrightarrow{h^x}$, while the backward LSTM reads the input sequence in reverse order from x_N to x_1 and estimates backward $\overleftarrow{h^x}$. For each input x_n, we then obtain h_n^x by concatenating forward $\overrightarrow{h^x}$ and backward $\overleftarrow{h^x}$. The decoder predicts the target sequence $\mathbf{y} = [y_1, y_2, ..., y_M]$ with length M by estimating the conditional probability $p(\mathbf{y}|\mathbf{x})$. This conditional probability is estimated based on the entire sequence of the previous output:

$$p(y_m|y_1, y_2, ..., y_{M-1}, x) = softmax(W_y \tilde{o}_m^y). \tag{6}$$

The decoder output vector o_m^y is computed by applying a linear layer W^o to the context information c_m and the current hidden state h_m^y:

$$\begin{aligned} h_m^y &= RNN(y_{m-1}) \\ c_m &= Attention(h_m^y; h^x) \\ o_m^y &= W^o[c_m; h_m^y]. \end{aligned} \tag{7}$$

Here c_m is the context information of the input sequence when generating the current output at time m, estimated by the attention module over encoder hidden

states h_n^x:

$$c_m = \sum_{n=1}^{N} a_m(n) * h_n^x, \tag{8}$$

where a_m is a variable-length alignment vector. Its size equals the length of input sequence $\mathbf{x}$, and is computed as

$$\begin{aligned} a_m(n) &= align(h_n^x, h_m^y) \\ &= softmax(dot(h_n^x, h_m^y)). \end{aligned} \tag{9}$$

This step helps the decoder find relevant information on the encoder side, based on the decoder's current hidden states. Fig. 3 gives an overview of this model.

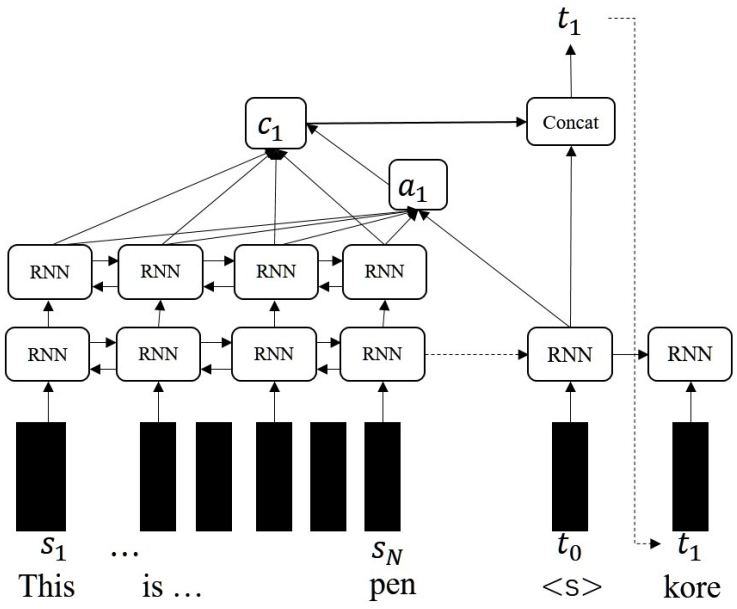


Figure 3. Attention-based encoder-decoder model overview.

2.2.4 Transformer Model

When using RNNs, each step's calculations need to weight the previous step process; therefore, the model can not compute the sequence data in parallel. Vaswani et al. proposed an encoder-decoder sequence-to-sequence model without

the use of recurrent mechanisms referred to as *transformer* [34]. The encoder maps an input sequence of symbol representations $\mathbf{x} = [x_1, ..., x_N]$ to a sequence of continuous representations $\boldsymbol{h} = [h_1, ..., h_N]$ using stacked FNNs. Given $\boldsymbol{h}$, the decoder then generates an output sequence $\boldsymbol{y} = [y_1, ..., y_M]$ one element at a time.

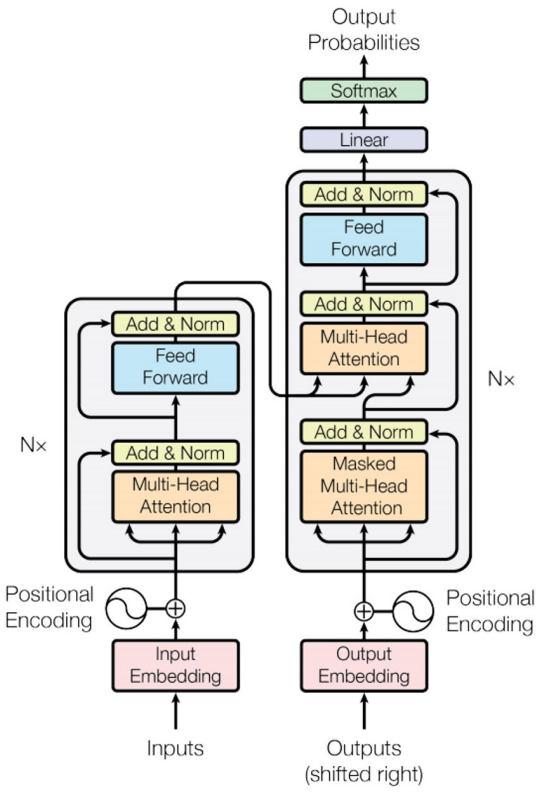


Figure 4. Transformer model overview [33].

The transformer follows this overall architecture using stacked self-attention and point-wise, FNN for both the encoder and decoder. The encoder is composed of a stack of multiple layers, each of which has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a position-wise FNN. The transformer has a residual connection around each of the two sub-layers, followed by layer normalization [2, 14]. The decoder is also composed of multiple layers, just like the encoder. In addition to the two sub-layers in each encoder layer, the decoder includes a third sub-layer, which computes multi-head attention over the

encoder stack's output. The attention function resembles mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. Fig. 4 illustrates the overall architecture of the transformer. The transformer model offers two benefits: (1) it enables parallel training by removing recurrent connections; (2) self-attention provides an opportunity for injecting the global context of the whole sequence into each input frame to model long-range dependencies directly. When performing forward and backward processing for current input and output, the transformer only computes the calculation graph path for relative states that are attended by a self-attention mechanism. However, an RNN-based encoder-decoder model will compute the calculation graph path for all previous states because it models the global context with recurrent structures.

2.3 Summary

In this section, I introduced the concepts of sequential data (text and speech) as well as that of sequential modeling using DNNs. Handling long sequence data is a difficult task, as the model needed to memorize past inputs and retrieved relevant information from memory. RNNs utilize a recurrent function to solve this problem. The network was given the last hidden state as past content information in addition to current input data. However, the primary sequence to sequence model can not pass on the input sequence context all the way to the target output sequence properly. The attention mechanism is a great tool to pass on context information in long sequential data from source to target. It helps the decoder to find relevant information in the encoder's output states at each step. The transformer model uses an attention function for modeling sequential data context instead of a recurrent function. This helps to reduce processing time and memory usage, and thus improves performance.

3. Cascade Speech Translation

3.1 Overview

Translating source language speech to target language speech is a very challenging task. Traditional speech-to-speech translation systems are composed of individual ASR, MT, and TTS modules. These modules are trained individually and share information via text representations. In this thesis, each module of a cascaded speech translation system will be an attention-based encoder-decoder model, with the encoder and decoder using either RNNs or a transformer architecture.
The input source language speech feature sequence $\mathbf{x} = [x_1, ..., x_N]$ is transcribed into a source text sequence $\mathbf{s} = [s_1, ..., s_I]$ by the ASR module. Then the MT module receives the source speech transcription $\mathbf{s}$ and translates it to the target language text $\mathbf{t} = [t_1, ..., t_J]$. Finally, the TTS module generates target speech $\mathbf{y} = [y_1, ..., y_M]$ from the target text $\mathbf{t}$. Each module only passes the output to the next process, as shown in Fig. 5. Here, each model (ASR, MT, and TTS)

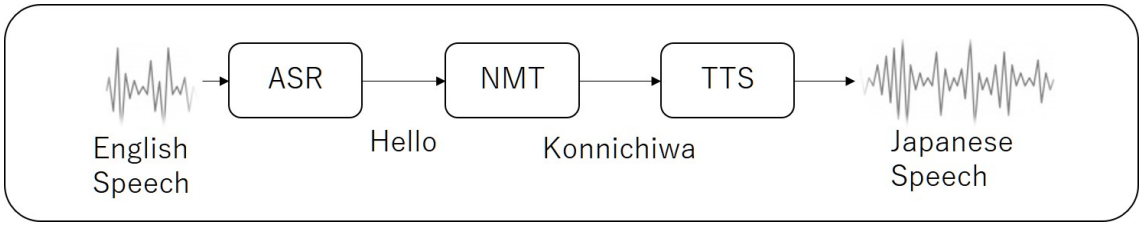


Figure 5. Cascade speech translation system.

is trained individually, and each module is built to solve a separate task. The ASR aims to reduce Word Error Rate (WER); the MT tries to convert the source sentence content into the target language, and the TTS focuses on generating clear and natural speech from the given text. Therefore these models have several limitations (see Section 1).

3.2 Speech Translation Components

3.2.1 Neural Automatic Speech Recognition

ASR refers to a system that transcribes input speech as text. The traditional approach referred to as GMM-HMM, models acoustic information using Gaussian mixture models, hidden Markov models, and linguistic information in the form of n-gram language models. The first attempts at performing end-to-end ASR were published by Chorowski et al. [8]. They used an attention-based encoder-decoder architecture [25] to map input source speech $\mathbf{x}$ to the target text $\mathbf{s}$ directly. The input sequence $\mathbf{x} = [x_1, ..., x_N]$ is a sequence of acoustic features representing the input speech, and the target sequence $\mathbf{s} = [s_1, ..., s_I]$ is the predicted output text (see Fig. 6). The RNN-based encoder consists of three bidirectional GRU layers.

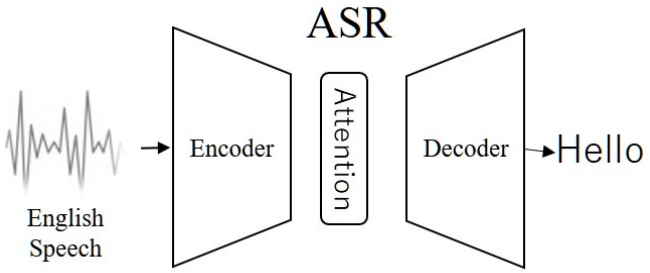


Figure 6. End-to-end ASR system overview.

All layers use an even number of input vector sequences to reduce memory usage and calculation time. For the decoder, I use Luong's design [25], as shown in Fig. 7, but with GRU cells.

The first Transformer based end-to-end ASR attempt was made by Dong et al. [10]. An ASR transformer has the same architecture as the original transformer. The ASR system consists of a multi-layer FNN, using rectified linear units (ReLu) as activation functions and convolutional neural network (CNN) layers with max-pooling. This is prepossessing for ASR, and this network reduces input speech noise and supports better feature extraction for post-processing.

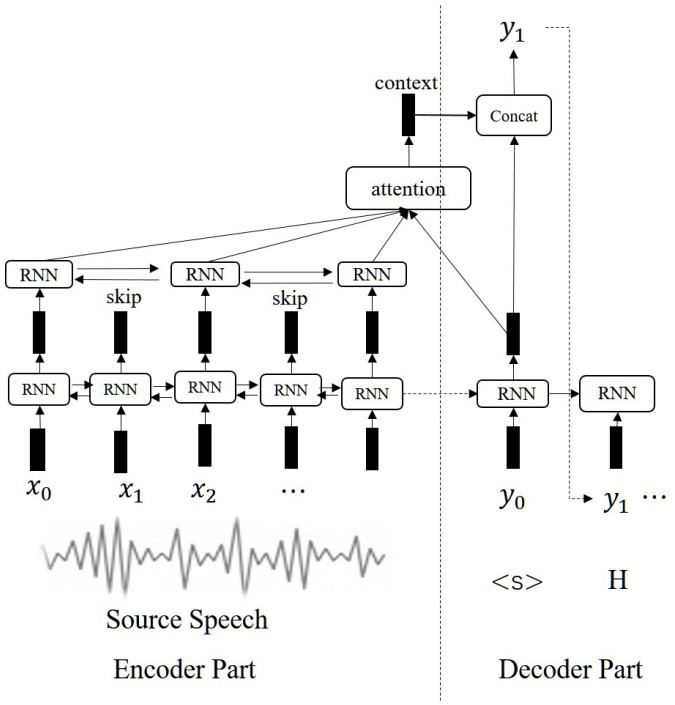


Figure 7. End-to-end speech translation skipping odd index states.

3.2.2 Neural Machine Translation

An end-to-end MT system translates source language text to target language text (see Fig. 8).

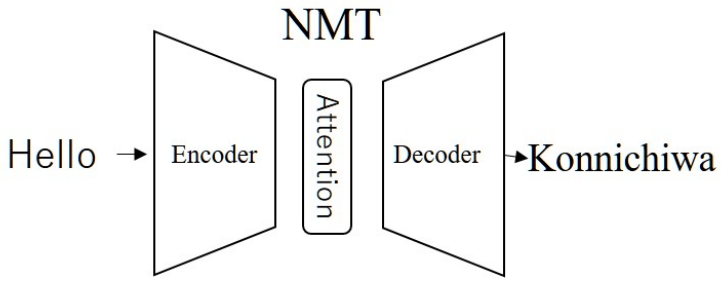


Figure 8. End-to-end MT system overview.

The input sequence $\mathbf{s} = [s_1, ..., s_I]$ consists of a sequence of one-hot-vectors with a size equal to the source language vocabulary, and the target sequence $\mathbf{t} = [t_0, t_1, ..., t_n, t_{n+1}]$ consists of a sequence of one-hot vectors the size of the target language vocabulary. t_0 and t_{n+1} are special tokens representing the start and of a sentence, respectively. The MT system has an embedding layer for both the encoder and the decoder. These embedding layers are FNNs that map input from each vocabulary dimension to a fixed dimension. Through this embedding process, words with similar meaning will be mapped more closely in the embedding space. Transformer-based MT systems are the current state-of-the-art. The model processes source and target sentences using stacked FNN layers and a self-attention function. The encoder maps the input sentence to the encoder hidden layer. Furthermore, the decoder uses previous outputs and current position information to decode a target word.

3.2.3 Neural Text-to-Speech Synthesis

End-to-end TTS systems generate speech given an input text sequence, as shown in Fig. 9

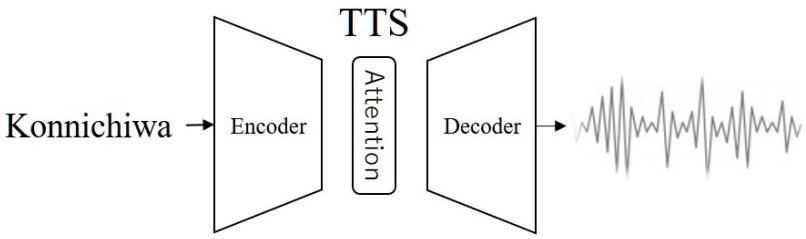


Figure 9. End-to-end TTS system overview.

The first end-to-end TTS system was created by Wang et al. [35]. The proposed Tacotron uses an RNN- and attention-based encoder-decoder model with a so-called Convolution Highway Bidirectional GRU network (CHBG). It is shown in Figs. 10 and 11.

End-to-end TTS generates a speech feature sequence $\mathbf{y} = [y_1, ..., y_J]$ from an input text sequence $\mathbf{t} = [t_1, ..., t_N]$. The input text sequence is represented by a one-hot vector identical to those described for MT above. Then the source

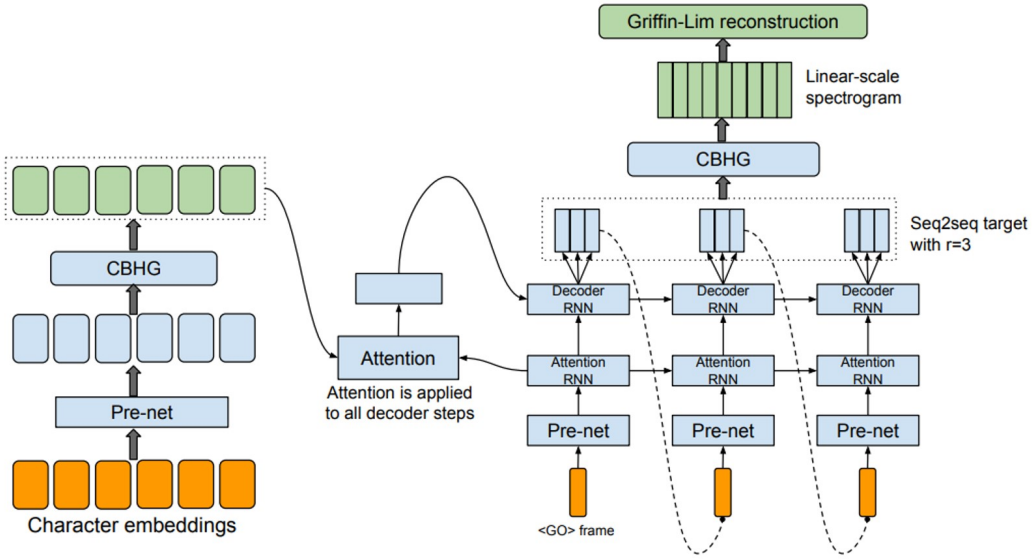


Figure 10. Tacotron [35] overview.

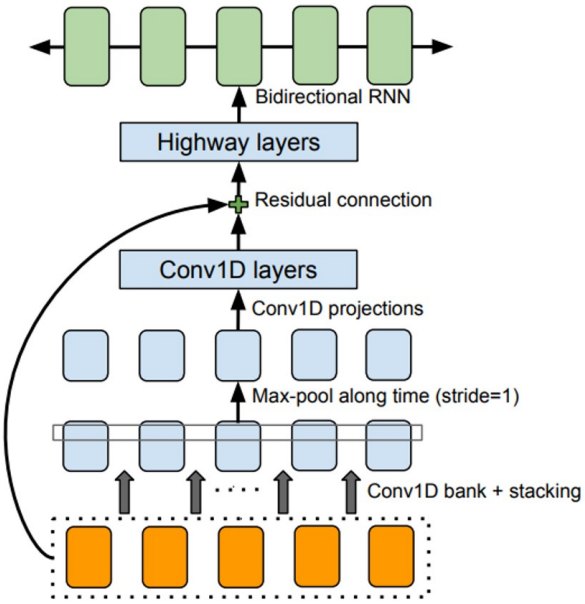


Figure 11. Convolution Highway Bidirectional GRU network [35] overview.

embedding vector is passed trough multiple CNN layers and encoded by a bi-directional GRU. The authors state that the CHBG network reduces over-fitting and results in better attention. The decoder generates h^y from the previous speech feature and a context vector. Then it generates multiple target feature vectors from one decoder hidden state. This technique can reduce calculation time. Finally, the Tacotron's up-sampling generates speech features as linear spectrograms via a CHBG, and generates waveforms using the Griffin-Lim reconstruction algorithm [13]. The first transformer-based end-to-end TTS was proposed by Li et al. [23]. Transformer-based TTS has the same architecture as the original Transformer, except for the pre-processing and post-processing parts. Instead, it uses the pre- and post-processing modules of the Tacotron2 [24]. Tacotron2 employs a three-layer CNN. I applied a CNN Network to the input text embeddings in order to handle the long-term context in the input character sequence. The original Transformer TTS model [23] uses an English *phoneme* sequence as input. However, in this model, I use English *character* sequences, like Tacotron2.

3.3 Problems

Cascade speech translation has three components. Each module is trained individually using separate loss functions. Each component passes its output text on to the next module. Because of this, the post-processing cannot see the original input information. It, therefore, has to "trust" the input it is handed by the previous component and process it as is. Therefore, if the ASR process produces errors, the MT results are affected by the erroneous input words it receives. Since each module is only sharing text information, the paralinguistic information (rhythm, emphasis, or emotion) in the original input speech is not retained in the target speech. In spoken communication, people expect others to understand how they feel from this paralinguistic information. Finally, over half of the world's languages are only spoken and have no written form. Thus, constructing speech translation that heavily relies on a text representation of spoken language would be limiting.

3.4 Summary

In this section, I described the overall structure of cascaded speech translation systems. Speech-to-speech translation is a challenging task. Therefore a cascaded speech translation system attempts to separate it into three individual modules and applies ASR, MT, and TTS models to an input sequence one after the other. I described the individual components involved and their deep learning architectures. I also summarized the problems faced when using a cascaded speech translation approach.

4. End-to-End Speech-to-Text Translation

4.1 From Cascade to End-to-End Speech-to-Text Translation

4.1.1 Proposed Speech-to-Text Translation with Posterior Vector

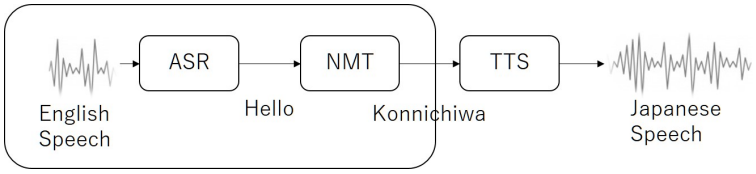


Figure 12. Speech-to-text translation overview.

First, I proposed a neural speech translation without requiring significant changes in the ASR and MT structure, as shown in Fig. 12. I perform a neural sequence-to-sequence ASR as feature processing that is trained to produce word posterior features given spoken utterances. This might resemble the word confusion networks (WCNs) that can directly express the ambiguity of the word hypotheses at each time point. The resulting probabilistic features are used to train MT with just a slight modification. Such vectors are expected to express the ambiguity of speech recognition output candidates better than the standard way using the 1-best ASR outputs while also providing a simpler structure than the lattice outputs. Through this model, I can observe each ASR candidate that input to MT, and how MT process those ASR candidates during training. This model passing ASR output models as following:

$$p(s_i|s < i, x) = softmax(W^s h_i^s). \tag{10}$$

Here, I train an end-to-end ASR using the standard attention-based encoder-decoder neural network architecture described in the previous section. But instead of providing 1-best outputs of the most probable word sequence to the translation system, I utilize the posterior probability vectors before the $argmax$ function, as shown in Fig. 13. I describe my proposed speech translation using the proposed word posterior vector in Fig. 14.

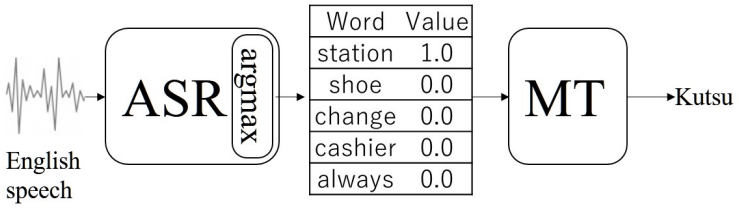


Figure 13. One-hot vector cascade speech translation overview.

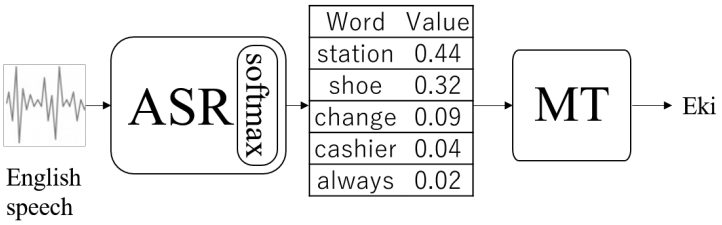


Figure 14. Word posterior vector cascade speech translation overview.

This way, the ASR output vectors can still express the ambiguity of the ASR output candidates with probabilities. The resulting probabilistic features are then used to train the back-end neural MT (MT) with only a slight modification. I train the end-to-end MT using the standard attention-based encoder-decoder neural network architecture described in the previous section. The only difference is in the input features. Instead of training the model with the one-hot vector of the most probable words, I utilize the posterior vectors obtained from the ASR. The dimension used in a standard one-hot vector and the proposed posterior vectors is the same.

4.1.2 Experiments for Word Posterior Speech Translation

First, I conducted my experiments using a basic travel expression corpus (BTEC) [18, 19]. The BTEC English-Japanese parallel corpus consists of training (480-k) and test (500) utterances. Since the corresponding speech utterances for this text corpus are unavailable, I used the Google text-to-speech synthesis[1] to generate

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

a speech corpus of the source language. I 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]. I further used these data to build an attention-based ASR, an MT system. I display the hyperparameter setting of these models in Tables 1 and 2.

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

Table 1. ASR settings.

RNN based ASR system	
Input units	80
Down-sampling ratio	0.25
FNN hidden units	256, 1024, 256
Encoder RNN layers	LSTM,GRU
LSTM and GRU hidden units	256,256
Encoder dropout ratio	0.3
Attention	MLP
Decoder layer	GRU
Decoder dropout ratio	0.3
Embed size	256
Decoder Embed dropout ratio	0.5

Table 2. MT settings.

MT system	
Encoder layers	LSTM,GRU
LSTM and GRU hidden units	256, 256
Encoder dropout ratio	0.3
Attention	MLP
Decoder layer	GRU
Decoder dropout ratio	0.3
Embed size	256
Decoder Embed dropout ratio	0.5

I find the end-to-end speech translation system could recover ASR error and improve the BLEU score, as shown in Fig. 5. Here, I perform farther survey using the word posterior speech translation system. I show examples of sentence

output by cascade speech translation and Word Posterior speech translation in Tables 3 and 6. In the first example in Table 3, ASR misrecognized "shoe" as "station." This error impacted the baseline, where it translated "station" as "eki" (the correct translation for "shoe" is "kutsuya"). However, in the proposed method, it was still able to translate it to "kutsuya." This might be because the ASR provided the recognition candidate, and each a probability is weighted (Table 4). Here, "shoe" information was still contained in the output vector with an only slightly lower probability than "station," and based on the context information, the machine translation translated the word as "kutsuya."

Table 3. Speech translation with ASR error.

ASR reference	Excuse me where is the closest **shoe** store?
ASR result	Excuse me where is the closest **station** store?
Cascade Model	Sumimasen ichiban chikai **eki** wa doko desuka?
Word Posterior Model	Sumimasen ichiban chikai **kutsuya** wa doko desuka?
MT reference	Sumimasen ichiban chikai **kutsuya** wa doko desuka?

Table 4. A word posterior example.

Recognized	Posterior
station	0.439
shoe	0.321
change	0.086
cashier	0.036
always	0.016

Table 5. Evaluation of Word posterior speech translation on BTEC natural speech.

	WER	BLEU
Baseline cascade ST	24.9	35.9
Proposed Word posterior ST	24.9	32.6

In this part, I analyze how the MT process could recover ASR error, and if the ASR output is correct, what is the difference between speech translation and MT. First, Table 6 shows the no ASR error case; then, the cascade speech translation identical the text-based MT. However, the cascade speech translation faced translation errors. I show the attention table in Fig. 15. From Fig. 15, I can find the attention weight spread to two encoder states "perm" and "haircut." This is because these two words have a similar meaning and map to close in hidden space. Then the decoder got an unclear context vector and generated error like Table 6. On the other hand, I describe the proposed speech translation attention table in Fig. 16, from this figure, I can find the decoder attend the correct word. I guess the "perm" and "haircut" is a similar meaning word pair. If I input the word id to MT, then MT maps those words close in hidden space. However, those pronunciation are different the ASR candidate should differ, then those candidate helps MT find the difference between "perm" and "haircut," and maps different place. Therefore the decoder could find a correct word using speech information.

Table 6. Speech translation without ASR error.

ASR reference	i d like to have **a perm and a haircut** please
ASR result	i d like to have **a perm and a haircut** please
Cascade Model	**paama to paama** o onegai shitai nodesuga
Word Posterior Model	**paama to katto** o onegaishimasu
MT reference	**paama to katto** o onegaishimasu

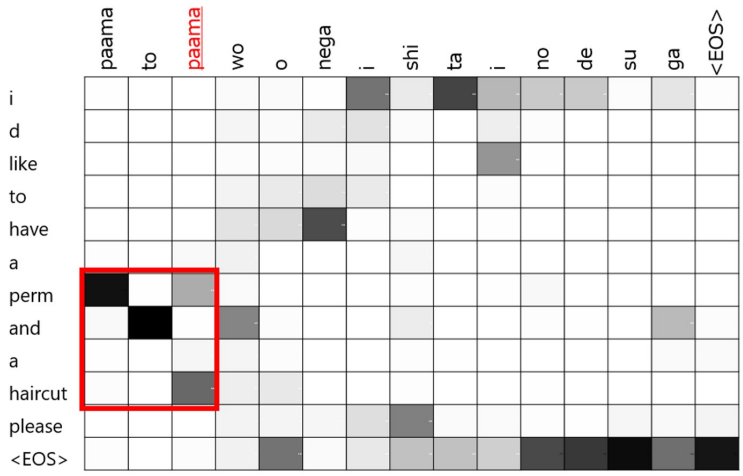


Figure 15. Attention table of Cascate speech translation without ASR error.

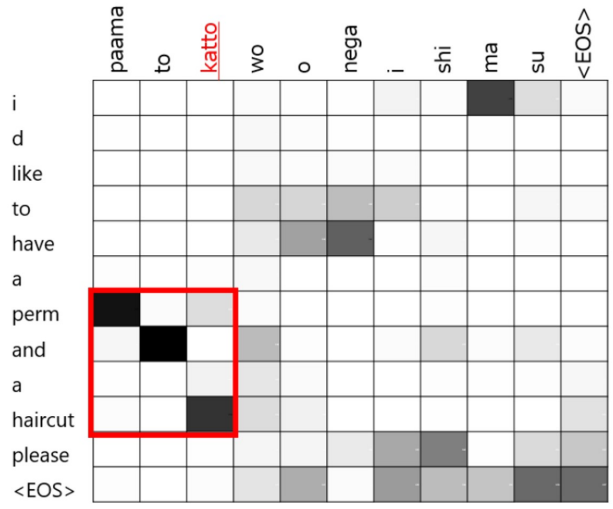


Figure 16. Attention table of Word Posterior speech translation without ASR error.

4.1.3 Discussions

In this section, I proposed a speech-to-text translation with a posterior vector. The posterior vector consists of more source speech information than that of text information. Therefore, the MT process could learn how to recover the ASR error by considering other ASR candidate. Moreover, speech translation could outperform text-based translation in some cases. In neural MT, the word input is represented as an id. Then the MT model maps the id to hidden space by embedding and encoder layer. Then the similar meaning words are mapped closely, however sometimes it maps too close, and the decoder cannot identify differences during decoding. The speech information given as a posterior vector helps identify these similar words and improve the attention performance in speech translation. Finally, the proposed speech translation could outperform text-based MT in some cases.

4.2 Direct Speech-to-Text Translation

4.2.1 Existing Works

The first direct speech-to-text translation system is proposed by Duong et al. [11]. The direct speech-to-text translation system gets input source language speech and output target language text, as shown in Fig. 17. In the end-to-end speech

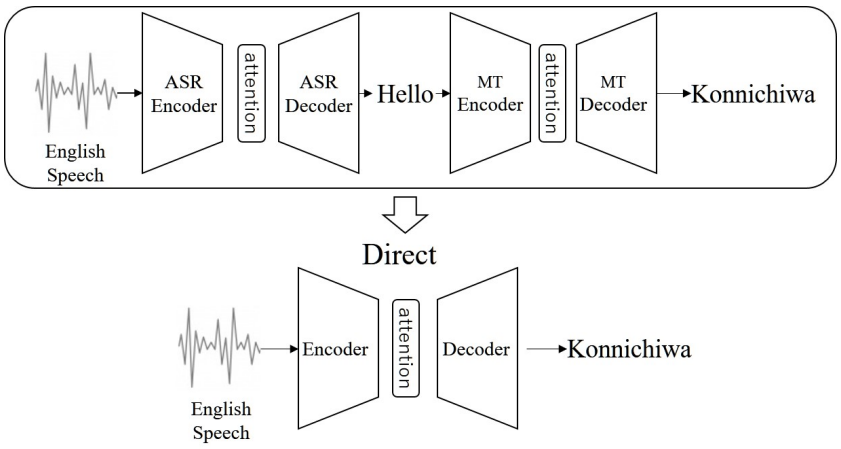


Figure 17. Direct speech translation system.

to text translation, the model should consider the speech transcription and input and linguistic aliment between source and target languages at once. This is a much more complicated task than other single tasks ASR and MT. Duong et al. mainly focus on aliment performance between source and target. Berard et al. utilize generated speech for training data and try to translate French speech to English text directly [6]. They also used a simple RNN based attention-based encoder-decoder model for translation.

$$\begin{aligned} \mathbf{h}^{\mathbf{x}} &= Encoder(\mathbf{x}), \\ h_m^t &= Decoder(t_{m-1}), \\ c_m &= Attention[h_m^t; \mathbf{h}^{\mathbf{x}}], \\ t_m &= argmax[W^{ht} h_m^t + W^{ct} c_m + b]. \end{aligned} \tag{11}$$

Here, $\mathbf{x}$ is input speech feature sequence, $\mathbf{h}^{\mathbf{x}}$ is source language speech encoded state sequence, h^t is target-language text encoded state sequence. The c_m is a context vector at m steps decoding. The t_* is the target at each step. Direct speech translation encodes source language speech, and the decoder finds attention point using target language word. In this thesis, I follow their approach and prepare RNN and Transformer based attention-based encoder-decoder models as a baseline. The model should solve two problems, and one is how encoder segments and groups input speech features, second is whether the decoder needs to find aliment between given previous target word and encoder states. The encoder state is a very long continuous vector sequence, and encoder and decoder handle different language data. Therefore direct speech translation is a very complex task.

Weiss et al. [37] utilize two decoders for end-to-end speech translation. One decoder for source language text transcription and another decoder for target language translation, as shown in Fig. 18.

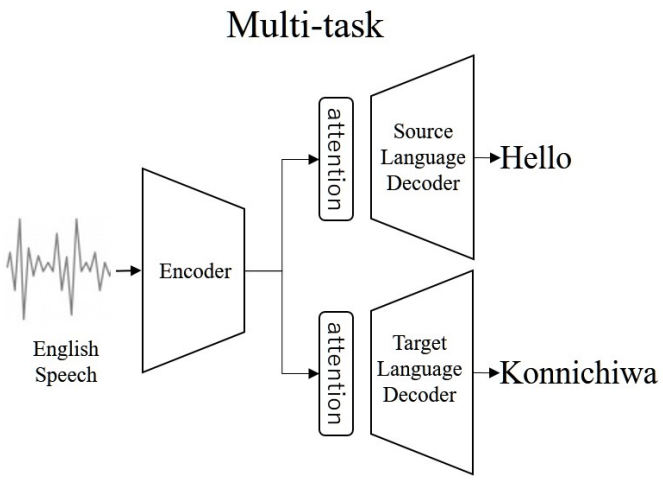


Figure 18. Multitask Speech Translation system.

The multitask-based speech-to-text translation is a probability model $P(\mathbf{s}, \mathbf{t}|\mathbf{x})$. Their model predicts both source language text $\mathbf{s}$ target language text $\mathbf{t}$ at once given source speech $\mathbf{x}$ as following:

$$\begin{aligned}
\mathbf{h}^{\mathbf{x}} &= Encoder(\mathbf{x}) \\
h_n^s &= Decoder^s(s_{n-1}) \\
\mathbf{c}_n^s &= Attention[h_n^s; h^x] \\
s_n &= argmax[W^{hs} h_n^s + W^{cs} c_n^s + b] \\
h_n^t &= Decoder^t(t_{m-1}) \\
\mathbf{c}_m^t &= Attention[h_m^t; h^x] \\
t_m &= argmax[W^{hy} h_m^y + W^{ct} c_m^s + b]
\end{aligned} \tag{12}$$

Here, $Decoder^s$ means source language text decoder. $Decoder^t$ is a target language text decoder. The model train to reduce both ASR error and speech translation error. The author claims the source speech transcription training helps

to make a better encoder state, and the trained encoder state helps to improve speech translation performance.

4.2.2 Proposed Transcoder-based Speech-to-Text Translation

Utilizing attention-based encoder-decoder architecture for constructing a direct speech translation task is difficult because the model needs to solve two complex problems:
(1) learning how to process a long speech sequence and map it to the corresponding words, similar to the issues focused on in the ASR field [8];
(2) learning how to make proper alignment rules between the source and target languages, similar to the issues discussed in the MT field [3, 22]. The multitask speech translation utilizes source text decoder to solve (1) and improve speech translation performance. In my proposed method, the model is not trained directly for speech translation tasks using parallel data. I train the attention-based encoder-decoder architecture by starting from a simple task, switch a specific part of the structure (encoder or decoder) in each training phase, and set it to a more difficult target task. In this way, the difficulty of the problems gradually increases in each training phase, as in the CL strategies. I describe each training phase's input and target sequence with their structures in Figs. 19-21.

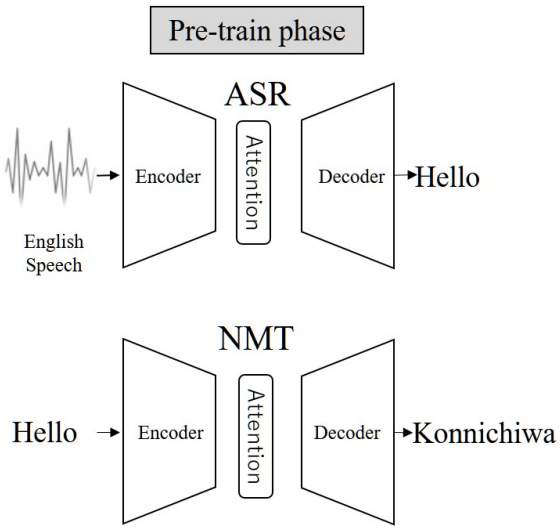


Figure 19. Proposed: Pre-training phase.

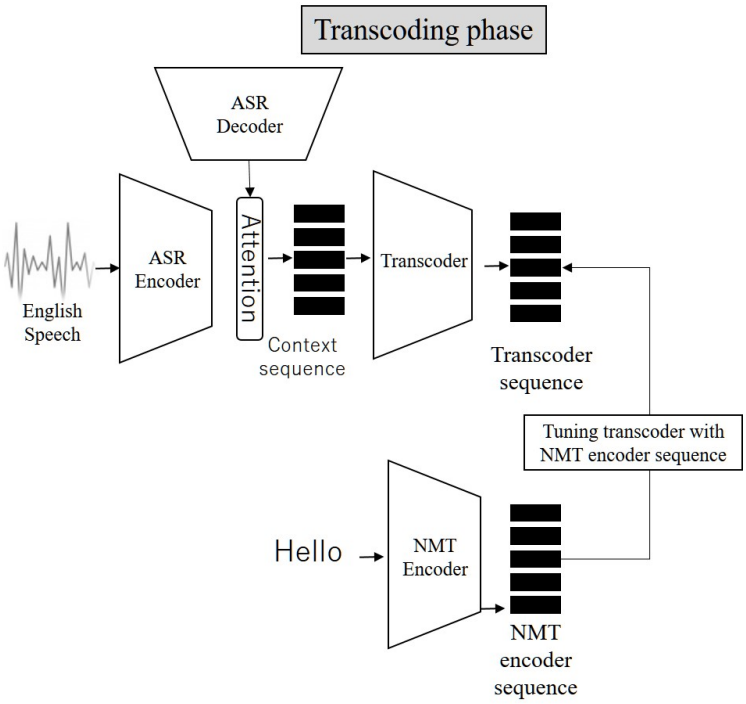


Figure 20. Proposed: Training transcoding phase.

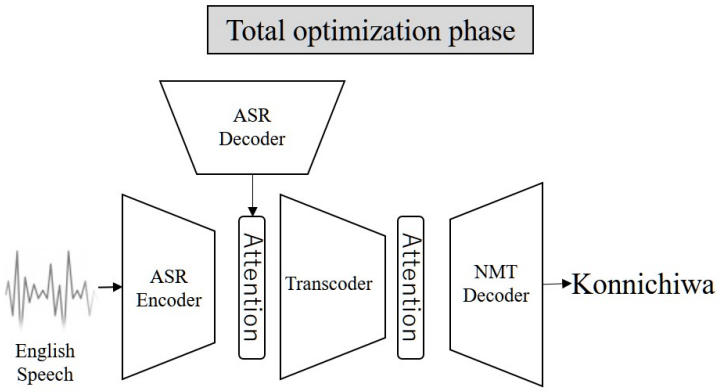


Figure 21. Proposed: Total optimization phase.

In this thesis, a transcoder is a unit that transfers acoustic hidden representations to the linguistic hidden representations. The acoustic hidden representations are the ASR context vectors, and the linguistic hidden representations are the MT encoder hidden states. The encoder generates a hidden state sequence given a speech features $H^{\text{enc}} = [h_0^{\text{enc}}, ..., h_n^{\text{enc}}]$. I perform the ASR decoding process to receive outputs h_m^{dec}. I use the MLP attention mechanism for the RNN-based model and a Multi-head attention mechanism for the transformer-based model. The ASR model provides the context vector sequence $C^{\text{ASR}} = [c_0^{\text{ASR}}, ..., c_m^{\text{ASR}}]$, where m represents the length of a source word:

$$\begin{aligned} h_m^{dec} &= Decoder^{ASR}(emb_{m-1}^{y}) \\ a_m &= attention(h^{enc}, h_m^{dec}) \\ c_m &= \sum_{n=1}^{N} a_t(n) * h_n^{enc}. \end{aligned} \tag{13}$$

$Decoder^{ASR}$ is a pre-trained ASR decoder, where y denotes the source language-text (words, subwords, or character sequences) and emb_{m-1}^{y} denotes the previous ASR decoder embedding vector. a_m is an attention score vector at the decoding step m. The ASR decoder aligns source speech to the source text and shares the alignment information as a context for the next process. The c^{s} denotes a hidden representation of the source-language acoustic information. I input source text embedding into the pre-trained MT encoder and generated output as linguistic hidden states. The transcoding process maps these acoustic hidden states to the linguistic hidden states of the source language. This process improves the attention result between the source and target sequences in the NMT decoder. The transcoding receives the ASR context vector sequence c^{x} and generates the transcoder output h^{tc} using a BiRNN. I then use MT encoder hidden states h^{t} as a target to optimize the transcoder:

$$\begin{aligned} h^{\text{tc}} &= Transcoder(c^{\text{s}}), \\ h^{\text{t}} &= Encoder^{s}(s). \end{aligned} \tag{14}$$

Here, s is a source-sentence embedding, $Encoder^{s}$ means pre-trained MT encoder. The number of h^{tc} and h^{t} states are equal to the source text length. During

this transcoding process training, I froze the NMT encoder parameters and only updated the ASR encoder and transcoder parameters. I thoroughly optimized the transcoder to minimize the smooth L1 loss between h^{tc} and h^{t}:

$$loss(h^{\mathrm{tc}}, h^{\mathrm{t}}) = \begin{cases} 0.5 * (h_n^{\mathrm{tc}} - h_n^{\mathrm{t}})^2, \\ \qquad if|h_n^{\mathrm{tc}} - h_n^{\mathrm{t}}| < 1, \\ |h_n^{\mathrm{tc}} - h_n^{\mathrm{t}}| - 0.5, \\ \qquad otherwise. \end{cases} \tag{15}$$

The model can learn a difficult problem on a small dataset using CL. End-to-end speech translation is difficult. Solving this problem requires the preparation of a more deep neural network and larger amounts of data compared to regular text NMT tasks. Since preparing parallel speech data is very expensive, I start training a model on a simple task and proceed to a more difficult task. In this way, the difficulty of the problems gradually increases with each training phase. In the end, the model can perform end-to-end speech translation using only a small initial training dataset. I first performed end-to-end speech translation with a small linguistic distant-language-pair dataset to confirm the CL benefits and compared the translation performance of several model architectures. I also performed speech translations with various language-pair large datasets to evaluate my proposed model and the baseline model. Finally, I used TED natural speech and performed end-to-end speech translation to confirm the effectiveness of my proposed approach.

4.2.3 Experiments for End-to-End Speech-to-Text Translation on BTEC

First, I use generated speech constructed in Section 4.1.2 for training data. To investigate the performance of my proposed system in natural speech, I also utilize BTEC corpus consists of 190-k utterances of natural English speech. However, it only has an 8-k speech-to-text parallel data of English-French and English-Japanese. Therefore, I use natural and generated speech to train ASR and speech translation systems, and I test the system on BTEC natural speech data.

Throughout this experiment, I describe the benefits and potential of my proposal's results. First, I demonstrate the BTEC translation task with the RNN-based model on the generated speech. Then I performed end-to-end translation tasks on natural speech with the Transformer model [33, 10], which is a state-of-the-art sequential model. I changed all the RNN networks to the FC layer and the self-attention function and applied my proposed method to confirm how it works with natural speech translation tasks. I 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[3]. I further used these data to build an attention-based ASR, an NMT system, the baseline Direct ST system, and my proposed ST system. The hyperparameter settings of these models are displayed in Tables 1-8.

For each system, I prepared characters, subwords [29], and words as translation sequences. At the evaluation steps, my final goal is to increase translation accuracy, which is the word level. Therefore, I combined characters or subwords into words for evaluation.

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

Table 7. Direct speech translation settings.

speech translation system	
Input units	80
Downsampling ratio	0.25
MLP hidden units	256
Encoder layers	LSTM,GRU
LSTM and GRU hidden units	256
Encoder dropout ratio	0.3
Attention	MLP
Decoder layer	GRU
Decoder dropout ratio	0.3
Embed size	256
Decoder Embed dropout ratio	0.5

Table 8. Proposed speech translation settings.

speech translation system	
Input units	80
Downsampling ratio	0.25
MLP hidden units	256
Encoder layers	LSTM,GRU
LSTM and GRU hidden units	256
Transcoder layers	LSTM,GRU
LSTM and GRU hidden units	256
Decoder layer	GRU
Decoder dropout ratio	0.3
Embed size	128
Decoder Embed dropout ratio	0.5

Table 9. Optimizer settings.

Optimization	
Initial learning rate	0.001
Optimizing method	Adam [20]

Table 10. Vocabulary size of each language and segment.

Vocabulary size			
Language	word	sub-word	character
English	27011	2918	29
Japanese	32794	2899	2691
Korean	55092	2902	1422
French	14802	2789	31

Next, I summarize the network parameters. For all systems, I used the same learning rate and adopted Adam [20] in all of the models, as shown in Table 9.

I applied the attention-based encoder-decoder architecture described in Section 3.2 to train the ASR, MT, and direct speech translation systems. I also constructed a cascade speech translation system and multitask-based speech translation system, as described in Fig. 5 and 18. For my proposed models, I applied my proposed CL-based training strategy to the attention-based encoder-decoder architecture described in Section 4.2.2.

Baseline Cascade ST:

A conventional speech-to-text translation model that cascades ASR and NMT systems (Fig. 5).

Baseline Direct ST:

A direct end-to-end speech translation model that uses a single attention-based neural network (Fig. 17).

Baseline multitask-based ST:

An end-to-end speech translation model that uses a pre-trained ASR encoder and an NMT decoder (Fig. 18).

Proposed CL-Transcoder ST:

My proposed direct end-to-end speech translation model trained with transcoder and CL strategies (Fig. 21).

To confirm my assumptions and the behavior of the proposed method, I extracted a small dataset from the original dataset. It was only 45k utterances for training and 500 utterances for testing, and I also limited the length of the input speech to less than 500 frames to save memory resources. The ASR system achieved an 8% word error rate (WER). For translation quality, I compared the BLEU+1 scores of each model's performance. I chose BLEU+1 because a BTEC corpus consists of many short utterances, BLEU+1 is a more suitable objective evaluation method than BLEU [27] scores for short translations [26].

First, I show how my proposed method works during training with a small amount of data. In this experiment, I limited the training data to only 45-k of generated speech utterances. I report the validation set softmax cross-entropy (CE) of each model in Fig. 22. From this figure, I conclude that the direct speech translation model encounters difficulties in the training process. This leads me to suspect that I require more training data. On the other hand, my proposed model and the pre-trained ASR encoder and NMT decoder concatenation model reduced the validation loss even with fewer training data. Note also that the ASRenc-NMTdec ST model's first epoch validation loss is as high as that of the direct translation model. Furthermore, my transcoding method begins and converges with better validation loss than the multitask-based speech translation model.

Table 11 shows the translation results of the baseline and proposed systems with the BLEU+1 scores. I also include text-to-text translation results (text-based NMT). The baseline Direct ST system with a single attention module failed to translate the English speech to Japanese text. Learning such syntactically distant languages as Japanese and English is difficult when the training data are limited.

However, the performance greatly improved when I applied pre-trained ASR and NMT parameters to the encoder and decoder in the multitask-based ST

system, which achieved identical performance as the baseline Cascade ST systems.

The proposed transcoder-based speech translation system achieved the best performance. It can be stably trained and successfully outperformed all baseline systems with a significant margin of BLEU+1 score. The performance of the proposed method even surpassed the performance of the text-based MT. The MT model trained on a small dataset. Therefore I guess there are many cases that I described in Section 4.1.2.

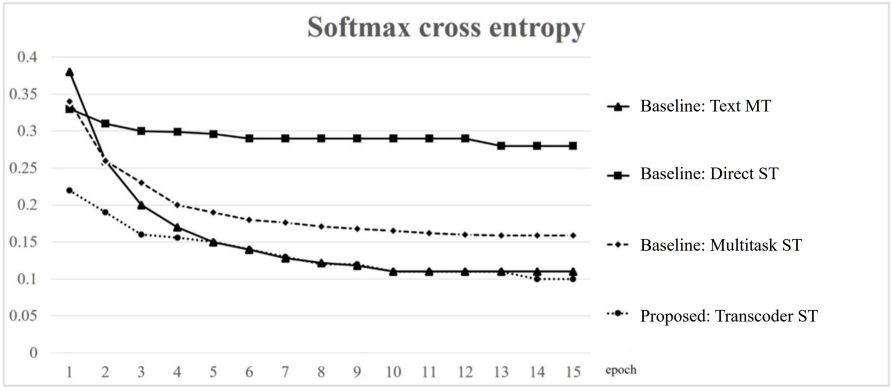


Figure 22. Softmax cross-entropy loss on a validation set.

Table 11. English-to-Japanese translation results (BLEU+1) on a small dataset.

Model	BLEU+1
Baseline: Cascade speech translation	28.6
Baseline: Direct speech translation	14.0
Baseline: Multitask-based speech translation	28.2
Proposed: Transcoder-based speech translation	34.3
Text-based MT	33.2

Table 12. ASR word error rate.

Language	Characters	Sub-words	Words
Ja	14.3%	7.1%	6.9%
En	10.1%	6.0%	5.9%

Next, I evaluated with a complete dataset and further investigated the performance of the systems in various units (character, subword, and word units) and various language pairs. I first calculated the WER for each ASR system on a small BTEC dataset (Table 12). The ASR achieved a satisfactory performance below 10% WER. I achieved higher performance on ASR because I used speech generated from TTS to train and evaluate the models. A single speaker generated TTS speech, and the speaking style is very stable. I then evaluated the translation quality for each system and showed the results in Tables 13-15. Tables 13 and 14 demonstrate that the performances of the baseline Cascade ST and Direct ST approaches are similar on subword and word translation on syntactically similar language pairs. However, similar to the phenomena with the small dataset, the baseline Direct ST did not perform well for syntactically distant language pairs. In such language pairs, more deep architecture and a better training strategy are necessary. In contrast, my proposed models outperformed both baseline systems on syntactically distant language pairs in the character-, subword-, and word-based systems (Table 15). Even on similar language pairs, my proposed approach successfully improved the end-to-end speech translation quality in the subword and word units.

Table 13. BLEU score of Baseline Cascade speech translation system.

Language pair	Characters	Sub-words	Words
Ja to En	25.5	30.5	32.6
Ja to Ko	31.0	40.1	41.9
En to Fr	34.8	39.9	39.7
En to Ja	29.2	32.7	33.1

Table 14. BLEU score of Baseline Direct speech translation model.

Language pair	Characters	Sub-words	Words
Ja to En	17.3	18.7	19.3
Ja to Ko	29.6	39.4	39.0
En to Fr	22.3	35.2	36.8
En to Ja	20.4	27.0	22.3

Table 15. BLEU score of the proposed Transcoder-based speech translation model.

Language pair	Characters	Sub-words	Words
Ja to En	30.5	36.2	37.4
Ja to Ko	30.1	42.8	43.0
En to Fr	33.0	41.4	42.7
En to Ja	30.9	37.0	38.6

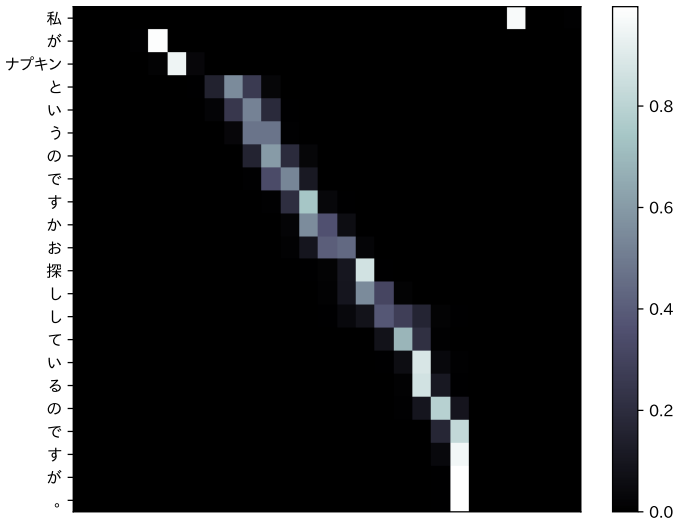


Figure 23. Japanese ASR attention.

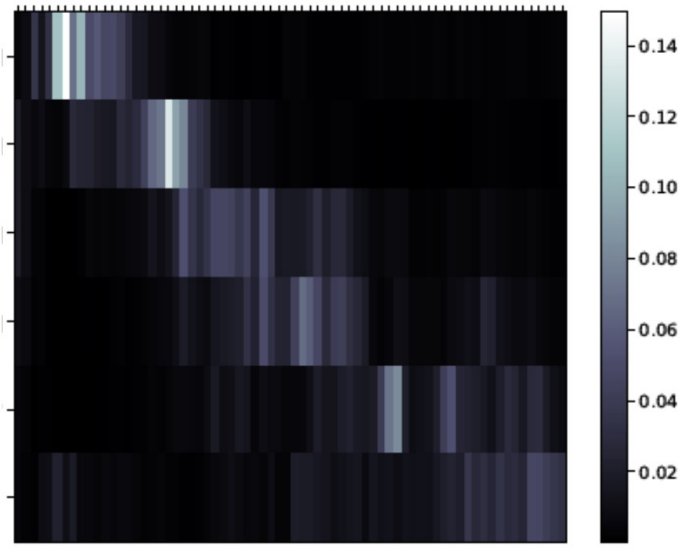


Figure 24. Japanese to Korean direct translation attention.

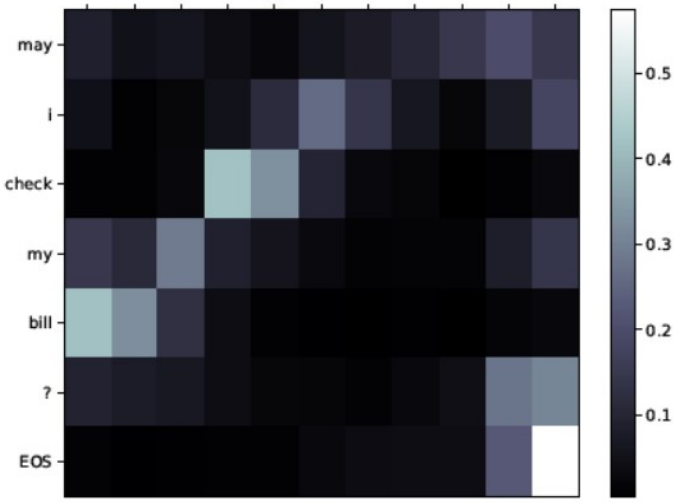


Figure 25. Japanese to English cascade translation attention.

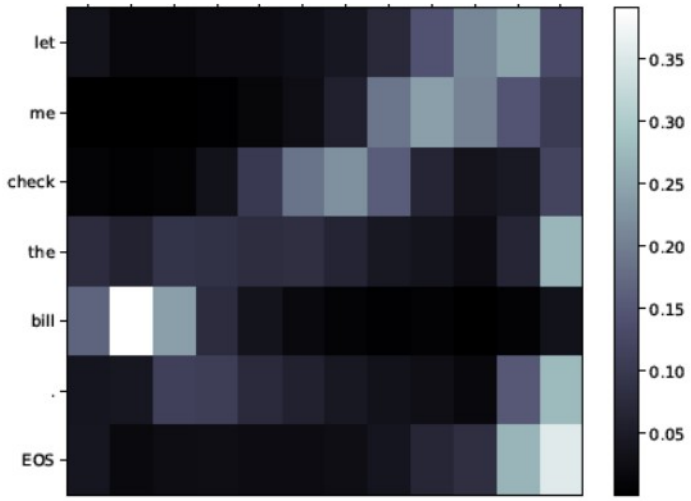


Figure 26. Proposed Japanese to English translation attention compared with cascade translation.

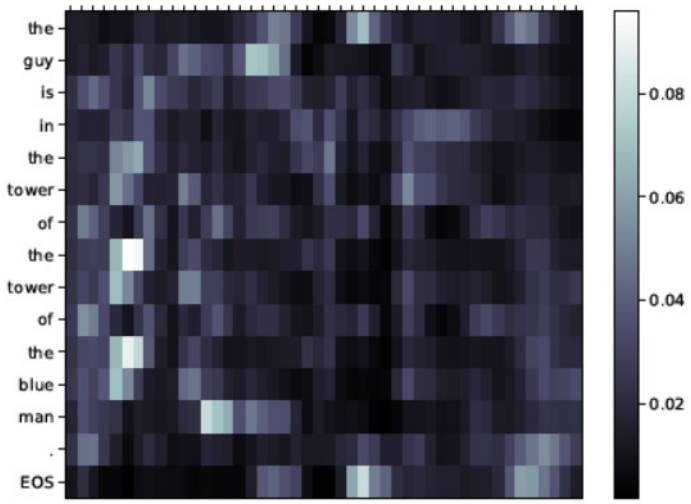


Figure 27. Japanese to English direct translation attention.

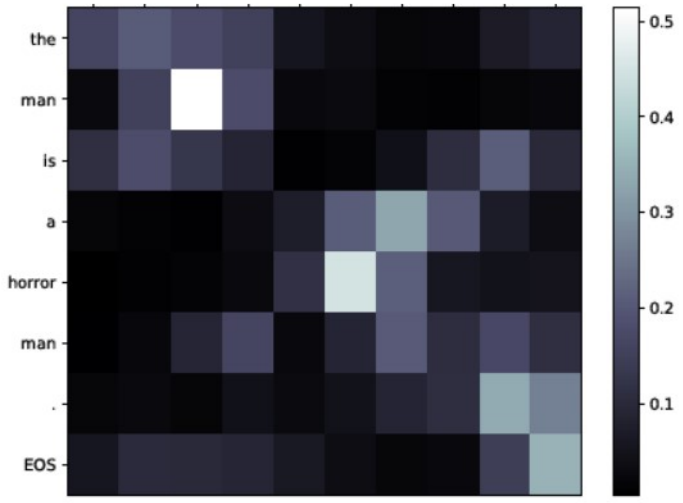


Figure 28. Proposed Japanese to English translation attention compared with a direct translation.

These experiments show that my proposed system has the potential to outperform the Cascade ST model. However, these results are based on generated speech data; therefore, I also performed a BTEC translation task with natural speech with a state-of-the-art sequential model Transformer for the ASR and NMT models. In this experiment, I used a character as a basic unit for the ASR model. I prepared a BTEC English natural speech dataset for English-French and English-Japanese translation. I summarize my Transformer parameters in Table 17. I used the same Transformer model for ASR and NMT, but ASR has a prenet module (FC and Convolution network) instead of a source embedding layer [10]. I added 20% Gaussian noise to the decoder embedding vectors to increase the beam-search performance on the test set. The transcoder model has three FC layers and a self-attention function, which is identical to the Transformer encoder layers. First, I trained the ASR model using the BTEC natural and generated speech. ASR achieved a 6% WER on the BTEC natural speech test set (500 utterances). The BTEC text NMT systems were given English character sequences as input and output subword target sequences. I show the ASR model's performance in Table 20 and present the text NMT, Cascade ST, and Proposed ST BLEU scores in Table 21. RNN-based model's natural speech translation per-

formances are slightly worse compared to that of generated speech translation. However, if I use the Transformer instead of RNN that is trained using both natural and generated speech, I achieved a high ASR performance of 6% WER. Also, the Transformer framework improved text translation performance when compared to the RNN-based model. From these results, I used the Transformer architecture below as the standard architecture of my proposed model.

Table 16. ASR pre-net settings.

ASR pre-net functions	
Input FC units	80
Output FC units	256
Convolution layers	3
Convolution input units	256
Convolution kernel size	5
Convolution dropout ratio	0.2
Batch Normalize	True
Post FC units input and output units	256

Table 17. Transformer settings.

Transformer parameters	
Encoder layers	3
Decoder layers	3,3
input and hidden units	256
transformer FC units	1024
Multi-head number	8
dropout ratio	0.2

Table 18. Text embedding settings.

Text embedding layers.	
English embedding input units	32
German embedding input units	44
French embedding input units	46
Embedding output units	256
Position Encoding	True
Decoder noising rate	0.2

Table 19. Transformer optimizer settings.

Optimization	
Optimizing method	Adam [20]
Warm-up steps	4000
Initial learning rate	0.001

Table 20. ASR word error rate of each model.

Model	BTEC test speech data	
	Generated	Natural
RNN ASR	8 %	9 %
Transformer ASR	1 %	6 %

Table 21. BLEU scores of BTEC natural speech translation.

Model	En to Fr	En to Ja
Direct speech translation	36.2	28.2
Multi-task speech translation	40.3	35.0
Proposed Transcoder-based speech translation	43.8	40.0

4.2.4 Experiments for End-to-End Speech-to-Text Translation on TED

I also performed experiments on the TED corpus[4], which consisted of 270k English sentences. All of these sentences have a corresponding French translation, but only 210k have German translations. I used these text datasets to train each English-French and English-German MT model. For the speech-to-text model, I used the TEDLIUM English 58k natural speech. Only 6k English utterances overlap between the TEDLIUM 58k natural speech and the TED talk parallel text dataset. I made a 6k-utterance English-French and English-German natural

[4]TED talks: https://www.ted.com/talks

speech-to-text corpus using these overlapping data. To add more data, I used Google TTS to generate another 270k English speech utterances from a parallel text corpus. Finally, I also utilized the IWSLT2018 English-German speech-to-text TED dataset, which consists of 2k talk waveform data. Based on the provided IWSLT2018 alignment information, I segmented those waveforms and got 178k English-German parallel utterances. Therefore, I trained my English ASR and transcoder with the TEDLIUM 58k utterances of natural speech, the IWSLT2018 178k utterances of natural speech, and 270k generated speech utterances. The English-French ST model was trained with a 270k generated speech-to-text corpus and a 6k natural speech-to-text corpus, and the English-German ST model was trained with a 210k generated speech-to-text corpus, a 6k natural speech-to-text corpus, and the IWSLT2018 178k utterances of the natural speech-to-text corpus. For evaluation, I used the IWSLT2018 "dev2010" dataset for a validation set as well as a "tst2015" and "tst2018" datasets for test sets.

4.2.5 Experimental Results on TED Talk

I also trained the ASR model using TED natural speech, BTEC natural speech, and TED generated speech. ASR achieved an 18% WER on the TED natural speech test set shown in Table 22. The TED natural speech included lots of noise, and therefore the generated speech and the BTEC natural speech did not improve the TED natural test speech ASR performance. I also trained the NMT model using only the TED corpus. I present the text NMT, Cascade ST, and Proposed ST BLEU scores in Table 23. For comparison in the same condition, the TED text NMT systems were given English character sequences as input and output subword target sequences. I trained the text NMT for each language pair and chose the best performance setting's output segments to train the ST model.

Table 22. ASR word error rate of TED natural speech.

Model	tst2015	tst2018
Transformer ASR	18 %	19 %

The results of this experiment are displayed in Table 23. The performances

Table 23. BLEU scores of TED natural speech translation.

	En to Fr	En to De	
Model	tst2015	tst2015	tst2018
Text based MT	29.8	25.1	25.3
Cascade ST	16.3	13.1	12.7
Proposed ST	17.1	13.8	13.0

of Cascade ST and Proposed ST were affected by ASR errors. Although the Transformer can give better performance than that of RNN, the Transformer still has some weaknesses. If the Transformer NMT got incorrect inputs (i.e., ASR errors), then NMT may output very short sequences (e.g., "so" or "and"). However, my proposed method has the potential to recover the ASR errors in the translation process. Therefore, my proposed method outperformed Cascade ST in natural speech translation tasks.

4.3 Discussion

4.3.1 Attention Passing for Speech-to-Text Translation

After my publication, Sperber et al. [30] simplified the transcoder-based speech translation model and proposed attention passing speech translation model shown as Fig. 29. They remove transcoding training process and passing Source language context c^s to MT model as following:

$$\begin{aligned} h^{\mathrm{x}} &= Encoder(x), \\ h^s &= Decoder(s), \\ c^s &= Attention[h^s; h^x], \\ h^{\bar{s}} &= RNN(c^s), \\ h^t_{m-1} &= RNN(t_{m-1}), \\ c^t &= Attention[h^t_{m-1}; h^{\bar{s}}], \\ t_m &= argmax[W^{ht} h^t_{m-1} + W^{ct} c_m + b], \end{aligned} \tag{16}$$

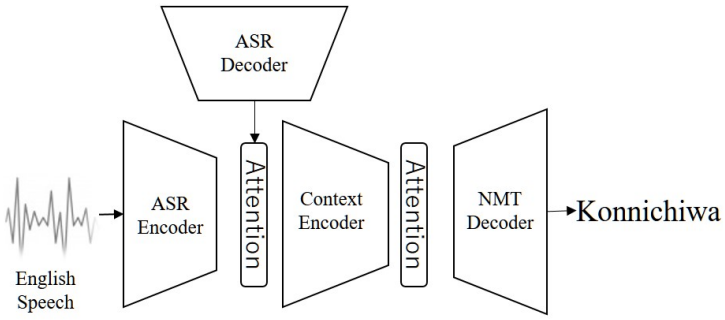


Figure 29. Attention Passing speech translation system overview.

Here, h^s is source language text decoder hidden states, h^s is an encoder hidden state that encoder speech context sequence and working as an MT encoder. Sperber et al. claim that their attention passing method transfers much more input speech information to the translation part, and this information helps recover ASR error during speech translation than transcoder models. The transcoder-based speech translation system maps ASR context to MT encoder hidden state during their second training phase. It helps to smooth connect ASR with MT pre-trained encoder, but the transcoder target is text encoder states. Therefore the speech information could be lost during transcoder training.

4.3.2 Experiments for Attention Passing Model and Proposed Model

I compare the attention passing speech translation and my proposed transcoder-based speech translation model in following Section 4.2.4. Both models have the same modules and similar architecture. The difference is that the transcoder uses the pre-train MT encoder states as a target in training. I describe the BLEU scores of each model in Figs. 24 and 25. In the TED English-French and English-German translation, the Attention Passing Model slightly outperform my proposed model. In the BTEC English-French and English-Japanese, my proposed model could outperform the attention passing model.

Table 24. Computation of Attention Passing model on BLEU scores of TED natural speech translation.

Model	En to Fr	En to De
Text based MT	28.2	18.4
Proposed End-to-end speech translation	14.8	10.2
Attention Passing speech translation	14.9	10.4

Table 25. Computation of Attention Passing model on BLEU scores of BTEC natural speech translation.

Model	En to Fr	En to Ja
Proposed End-to-end speech translation	43.8	40.0
Attention Passing	45.2	38.2

4.3.3 Experiments for Various Language Pairs

To compare the direct speech-to-text translation model and my proposed transcoder-based speech-to-text translation model on various language pairs, I use the BTEC1 dataset that includes 160k utterances for each of its 17 languages. In these experiments, I use both the direct and transcoder-based models to translate generated English and Japanese speech to various target languages.

Language	Code	word order
English	en	SVO
French	fr	SVO
Spanish	es	SVO
Dutch	nl	SVO
German	de	SVO
Danish	da	SVO
Vietnamese	vi	SVO
Indonesian	id	SVO
Portuguese	pt	SVO
Malay	ms	SVO
Italian	it	SVO
Thai	th	SOV
Chinese	zh	SVO
Tagalog	tl	VSO
Arabic	ar	VSO/SVO
Korean	ko	SOV
Japanese	ja	SOV
Hindi	hi	SOV

Table 26. List of target languages with word order.

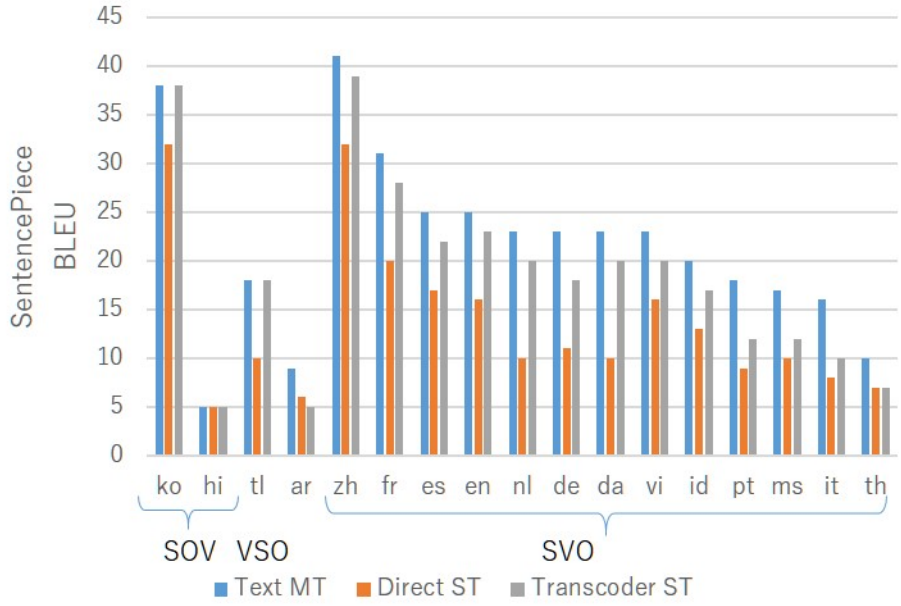


Figure 30. Results for speech-to-text translation of Japanese (SOV word order) to languages with various types of word order.

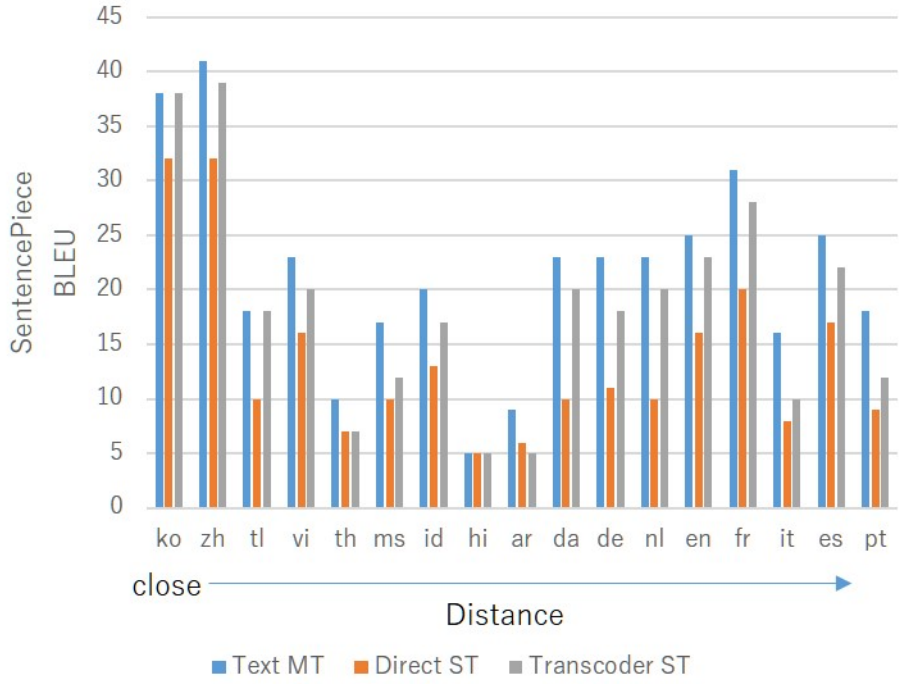


Figure 31. Results for speech-to-text translation of Japanese to various global languages.

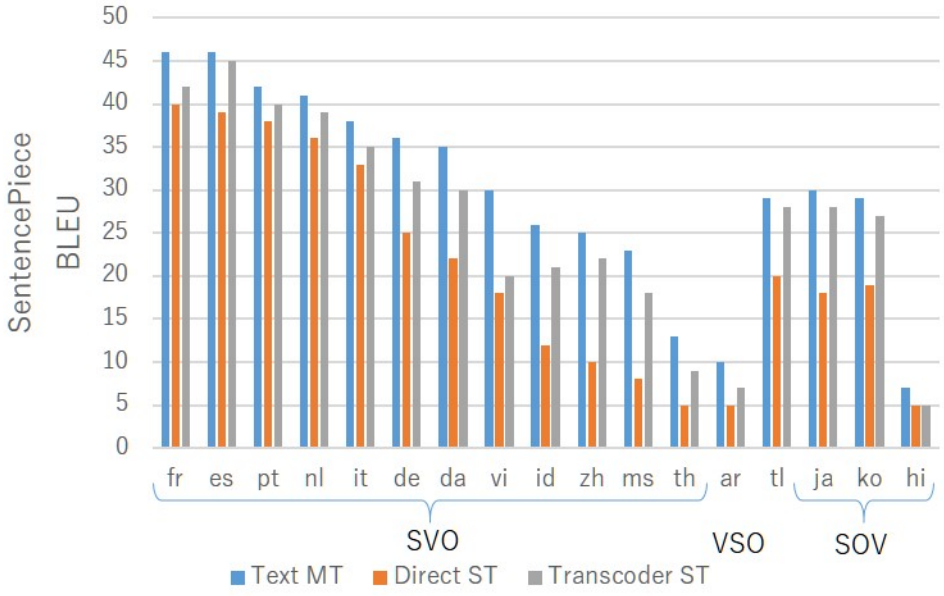


Figure 32. Results for speech-to-text translation of English (SVO word order) to languages with various types of word order.

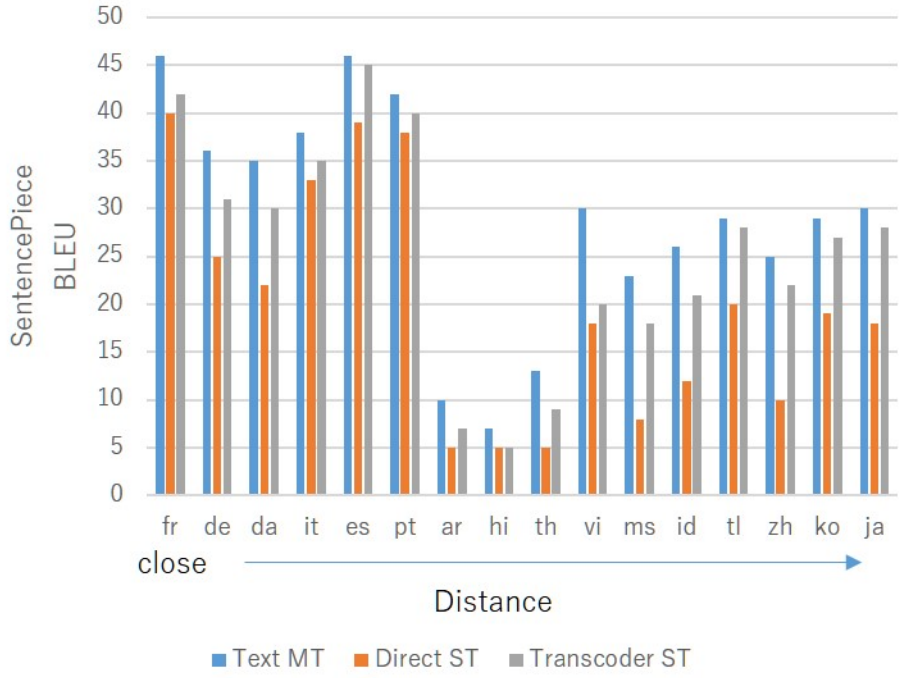


Figure 33. Results for speech-to-text translation of English to various global languages.

Figs. 30-33 show the sentence Piece-based BLEU scores. The figures show that the translation performance is not only based on similarities in word order. For example, when translating from English, BLEU scores are consistently higher for European target languages such as German or Danish when compared to performance for non-European target languages like Korean or Japanese. For the opposite direction, using Japanese as the source language, while it is the case that performance for target languages that are considered syntactically or culturally close (Korean and Chinese) is considerably higher, it is also obvious that for other, Asian languages there does not seem to be much of a correlation between closeness and the recorded BLEU scores. On the other hand, the geographical areas hosting Arabic and Hindi speaker bases are located further away from Japan and England, and they do not have as much of a history of cultural interaction with these countries. This could explain why performance when translating to these languages from either source language is consistently low.

For the comparison shown in Table 27, I defined the pairings "En to Es, Fr; and Ja to Ko, Zh" as syntactically similar, "Ja to En, Fr, ES; and En to Ja, Ko, Zh" as syntactically distant. From these results, I can find that my proposed model can outperform the direct translation model for both syntactically similar and distant languages.

Language Pairs	Direct	Transcoder	Difference
English to Similar Languages	39.5	43.5	4.0
English to Distant Languages	15.6	25.6	10.0
Japanese to Similar Languages	32.0	38.5	6.5
Japanese to Distant Languages	17.6	24.3	6.7

Table 27. Evaluation of similar language pairs and distant language pairs.

Furthermore, the direct translation model has difficulties achieving good performances for distant language pairs. In end-to-end speech translation, when the text translation is difficult, the speech translation task will also become more difficult. This is because speech translation needs to solve the speech-to-text alignment and text translation at the same time.

My proposed method, on the other hand, solves the speech translation step by step, with the ASR decoder supporting the alignment of the speech sequence to text. Therefore, the improvements in distant languages are larger than the improvements in similar languages.

4.4 Summary

I presented the construction of end-to-end speech translation for distant language pairs with transcoding based on CL strategies that gradually train the network for end-to-end speech translation tasks by adapting the decoder or encoder parts. My experimental results demonstrated that the translation quality outperformed the cascaded speech translation, standard direct speech translation, Multi-task speech translation systems, and attention passing models on syntactically distant language pairs. The results reveal that my proposed model effectively decreased loss, even using direct complex problems. Furthermore, my proposed word posterior speech translation model demonstrates how speech information recovers ASR error and improve the performance during the translation process, and end-to-end speech translation potentially could over perform text MT. This result promotes to perform end-to-end speech translation.

5. End-to-End Text-to-Speech Translation

5.1 Proposed Neural Machine Translation with Acoustic Embedding

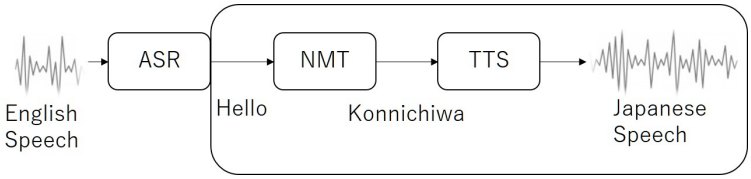


Figure 34. Text-to-speech translation overview.

Many works focus on end-to-end speech-to-text translation. However, no works focus on text-to-speech translation like a Fig. 34. In this section, I consider the benefit of applying speech information from the target side. In this section, I make a translation model that could utilize speech information from target language speech. An encoder-decoder translation model maps an input sequence into fixed-dimension vectors [31]. Such representations are sensitive to the meaning of the sequence and accurate word order, but they are insensitive to the replacement of the active voice with the passive voice. In speech synthesis models, these representations are sensitive to the replacement of the active voice with the passive voice but insensitive to the meaning of the sequence.

These attributes create models that are robust to test sets and generate natural sequences. On the other hand, the model sometimes confuses output with similar meaning words and context like a “dog” and “cat” and “may I” and “can I.” In this research, I map an input sequence into vectors of intermediate representation that is sensitive to both the sequence’s meaning and its pronunciation. I expect the pronunciation information to help discriminate among words with similar meanings and contexts in translation. I consider meaning and pronunciation using speech as either input or output. However, end-to-end speech translation usually decreases the translation quality. Moreover, preparing a natural speech parallel corpus is difficult. In this research, I used a pre-trained TTS embedding weight for the MT output layer. The transformer decoder has two modules that

handle the target word: a target word embedding layer and an output layer. I treated these two as inverse mappings and tied their weights [28]. In this research, I tied a decoder embedding layer weight and a decoder output layer weight. I added a new output layer where the mapping decoder hidden states using TTS embedding weights that do not update during training. This model has two types of output layers. The standard decoder output layer weight is tied to the decoder embedding weight. This output layer maps decoder hidden to output for a sensitive sequence meaning. The output layer is tied with a TTS embedding weight, maps decoder hidden for sensitive sequence pronunciations. I use these to output the results and back-propagate the loss:

$$\boldsymbol{o}_{nmt} = \boldsymbol{W}_{nmt}\boldsymbol{h}_t, \tag{17}$$

$$\boldsymbol{o}_{tts} = \boldsymbol{W}_{tts}\boldsymbol{h}_t, \tag{18}$$

$$loss = (1-\lambda)CE(\boldsymbol{o}_{nmt}, \boldsymbol{y}) + \lambda CE(\boldsymbol{o}_{tts}, \boldsymbol{y}). \tag{19}$$

Here, h_t is a decoder hidden sequence, $\boldsymbol{W}_{nmt}$ denotes a decoder word embedding weight, and $\boldsymbol{W}_{tts}$ denotes the TTS encoder embedding weight. In this thesis, $\boldsymbol{W}_{tts}$ do not update during training. I only update $\boldsymbol{W}_{nmt}$ through training. I use softmax CE to individually calculate the loss for each output, and λ is the weight for each loss. I named my proposed method as multi-task learning:

$$\boldsymbol{o}_y = \boldsymbol{o}_{nmt} + \boldsymbol{o}_{tts}, \tag{20}$$

$$loss = CE(\boldsymbol{o}_y, \boldsymbol{y}). \tag{21}$$

I sum both MT and TTS weight mapping. Since I do not update the TTS embedding weight during training, the model updates the MT embedding weight scale based on the degree of each layer's contribution. If the output from the MT embedding scale greatly exceeds the output from the TTS embedding, then the proposed model resembles a standard MT. I call my proposed method as "joint learning." I summary this section in Fig. 35.

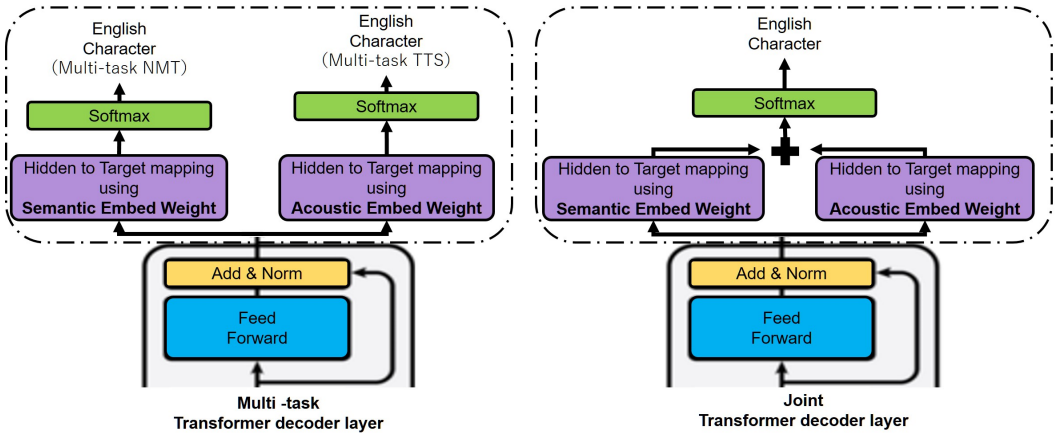
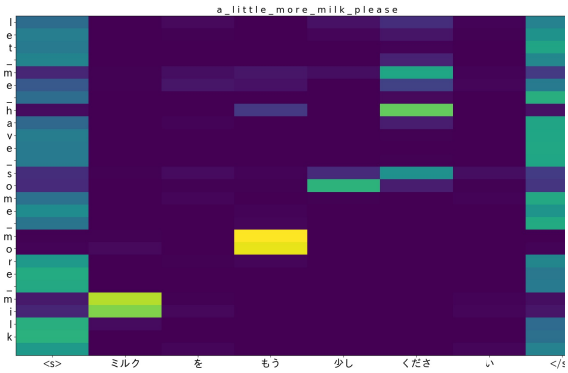
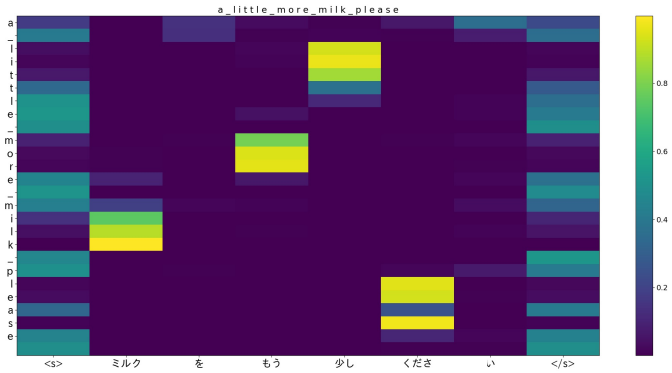


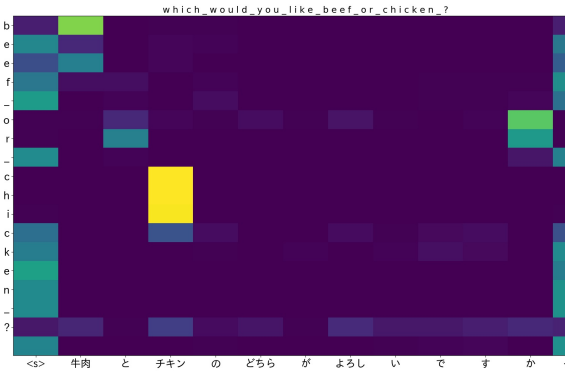
Figure 35. Multitask embedding MT architectures.



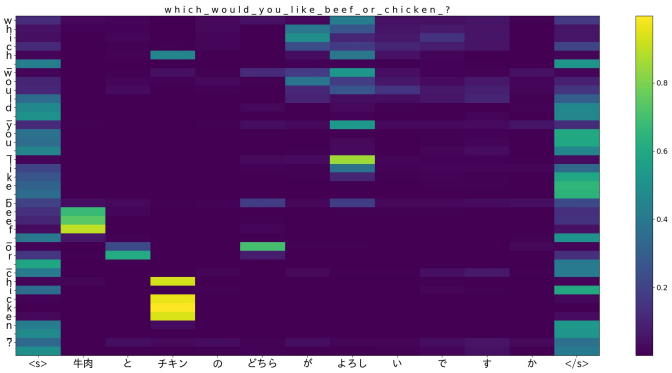
(a) Character-based MT with correct attention



(a) Proposed model with correct attention



(b) Character-based MT with wrong attention



(b) Proposed model with correct attention

Figure 36. Attention table of character-based MT and proposed model.

Source	Miruk wo mou sukoshi ku dasa i
Target	a little more milk please
Character-based MT	**let me have some** more milk
Subword-based MT	**let me have some** more milk
Proposed Multi-task$_{MT}$	a little more milk please
Proposed Multi-task$_{TTS}$	a little more milk please
Joint	a little more milk please
Source	Gyuniku to Chikin no dochira ga yoroshi i de su ka
Target	which would you like beef or chicken ?
Character-based MT	** beef or chicken ?
Subword-based MT	** beef or chicken ?
Proposed Multi-task$_{MT}$	which would you like beef or chicken ?
Proposed Multi-task$_{TTS}$	which would you like beef or chicken ?
Joint	which would you like beef or chicken ?
Source	i i o tenki de su ne
Target	it 's a lovely day is n't it ?
Character-based MT	**beautiful weather** is n't it ?
Subword-based MT	it 's **nice** day is n't it ?
Proposed Multi-task$_{MT}$	**nice** day is n't it ?
Proposed Multi-task$_{TTS}$	**nice** day is n't it ?
Joint	**nice weather** is n't it ?

Table 28. Translation results of Japanese-to-English part 1.

5.2 Experiment for Neural Machine Translation with Acoustic Embedding

I conducted my experiments using BTEC Japanese-English parallel corpus following Section 4.1.2. I removed the sentences that have more than 100 characters and used this dataset to build a baseline and proposed sub-words for the characters for the Transformer MT. I also used these data to build a Transformer TTS. My proposed model uses this pre-trained TTS acoustic embedding weight.

Fig. 37 illustrates the attention matrix of the pre-trained transformer. My

Source	zyo han shin wo kita e ta i nn de su kedo dono ma shin wo tuka e ba i de su ka
Target	i 'd like to work on my upper torso which machines should i use ?
Character-based MT	i would like to **build you medicine** my upper body ?
Subword-based MT	i 'd like to work on my upper torso which machines **can** i use ?
Proposed Multi-task$_{MT}$	i 'd like to work on my upper body **what** machine should i use ?
Proposed Multi-task$_{TTS}$	i 'd like to work on my upper body **what** machine should i use ?
Proposed Joint	**excuse me** i 'd like to **keep** my upper body which machine should i use ?
Source	san de su
Target	three
Character-based MT	three
Subword-based MT	three
Proposed Multi-task$_{MT}$	**from** three
Proposed Multi-task$_{TTS}$	three **of us**
Proposed Joint	**us** three

Table 29. Translation results of Japanese-to-English part 2.

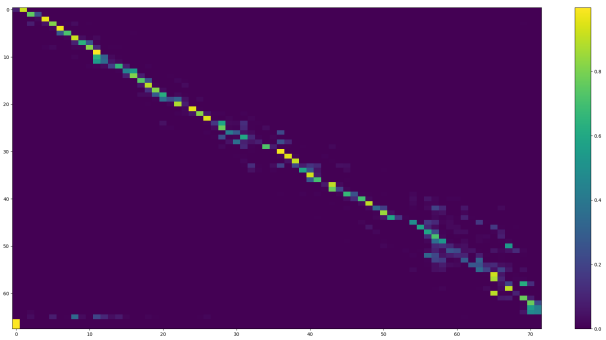


Figure 37. TTS attention table: TTS Mel-spectrogram L1 loss is 0.05 with grand truth.

TTS model shows clear monotonic shape attention and achieves a 0.05 L1 loss of the TTS Mel-spectrogram from grand truth decoding. The TTS module embedding layer is well trained from these results.

Next, I demonstrate my proposed translation performance and compare it with the standard text-based MT model. I used OpenMT[5] to make a baseline and implemented my proposed model on it. Here is a summary of the baseline and my proposed models:

Baseline Subword- and character-based MT

This is a standard text-to-text translation model. Input Japanese.

Proposed Multi-task$_{MT}$

Proposed model with multi-task learning and MT embedding weight output layer in test decoding.

Proposed Multi-task$_{TTS}$

Proposed model with multi-task learning and TTS embedding weight output layer in test decoding.

Proposed Joint

Proposed model with joint learning and both output layer in test decoding.

All models performed a beam search (the beam size is 5) algorithm for character sequence auto-regressive decoding. Here, I summary the model parameters in Tables 30-31. The baseline text-based MT and my proposed model used the same settings, and the trainable number parameters are the same between the proposed model and the baseline.

[5]OpenMT: http://opennmt.net/

Table 30. Transformer settings.

Embeddings	
Source vocabulary	6439
Target subword vocabulary	6102
Target subword vocabulary	35
Embedding size	512
Noisy ratio	0.2
Transformer block	
Hidden size	512
Number of layers	3
Transformer FFN	2048
Self-attention head	8
Dropout ratio	0.1
Attention mechanism	Multi-head

Table 31. Optimizer setting.

Optimizer	
Method	Adam [20]
Adam $\beta 1$	0.9
Adam $\beta 2$	0.998

Table 32. Translation quality of Japanese-to-English.

Model	BLEU score	WER
Character-based MT	45.10	35.5%
Subword-based MT	49.06	32.0%
Proposed Multi-task$_{MT}$	**50.51**	30.5%
Proposed Multi-task$_{TTS}$	50.23	**30.1%**
Joint	48.12	32.4%

Table 32 shows that my proposed method successfully improves the BLEU score by 5 points compared to the character-based MT and 1.4 points compared to the subword-based MT. For further discussion of the model behaviors, Tables 28 and 29 show the translation results from each model. Each proposed model output a sentence whose meaning was very similar to the meaning of the target sentence. This means that each proposed model extracted the meaning of the source sentence and mapped it to the decoder state. However, in comparison to the character-based MT baseline, the character-based MT model failed to choose the correct word from the decoder state. The output layer is usually one simple linear regression layer that maps a vector from a continuous narrow space to a large discrete space. If the model maps a similar word too closely, then the output layer cannot separate it again.

On the other hand, my proposed model output the correct word for each sentence. This reveals that by incorporating acoustic embedding and constructing a model in a multi-task fashion with two output layers, each layer can map the decoder state to different outputs with different weights. The hidden representation might be sensitive for both semantic and pronunciation similarities. Therefore, my proposed model can choose the correct word that not only depends on its meaning but also its pronunciation.

I also show two attention table pairs to compare my proposed model multi-task and the baseline. Both models generated similar sentences with correct attention. I believe translation error occurred at the decoder output layer, not at the encoder or attention sides. My model also attended to the same word but generated correct words. My decoder part separated sequences with similar

meanings, as I expected.

I show another attention table where the baseline made some attention errors. The baseline only focused on "beef" and "chicken." These are the most important words for translating this input sentence. Other source words (except the last words) are formula words and last words denote questions. In Japanese-English travel conversation translation, since there are many insertions and deletions, not all of the source inputs are attended during translation. Therefore, my baseline MT only attended to "beef" and "chicken." On the other hand, my model correctly attended to all of the words. My model became sensitive to both the meaning and the pronunciation in the decoder hidden state. A benefit also appears in the encoding and attention module through back-propagation. My model found correct attention in this case. But in another case, at the bottom of Table 29, my proposed method generated unnecessary words.

5.3 Summary

I use TTS embedding weight to map translation results. This approach created an MT model that is sensitive to sequence meaning and pronunciation. My proposed method outperformed a standard transformer with BLEU scores. I first considered MT and TTS collaboration. My proposed method made an MT that can learn such multi-modal information as text meaning and pronunciation from a text.

6. End-to-End Speech Translation

6.1 Existing Work

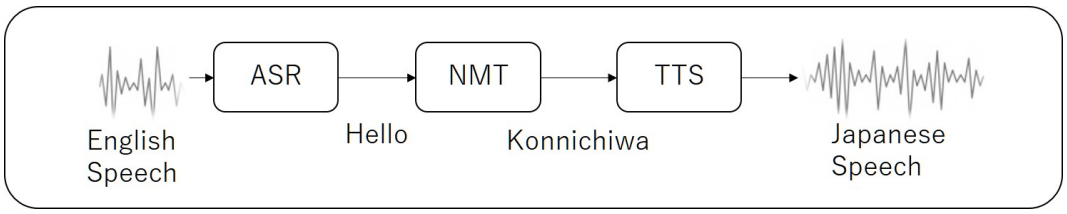


Figure 38. Speech-to-speech translation overview.

In this section, I focus on end-to-end speech translation task, as shown in Fig. 38.

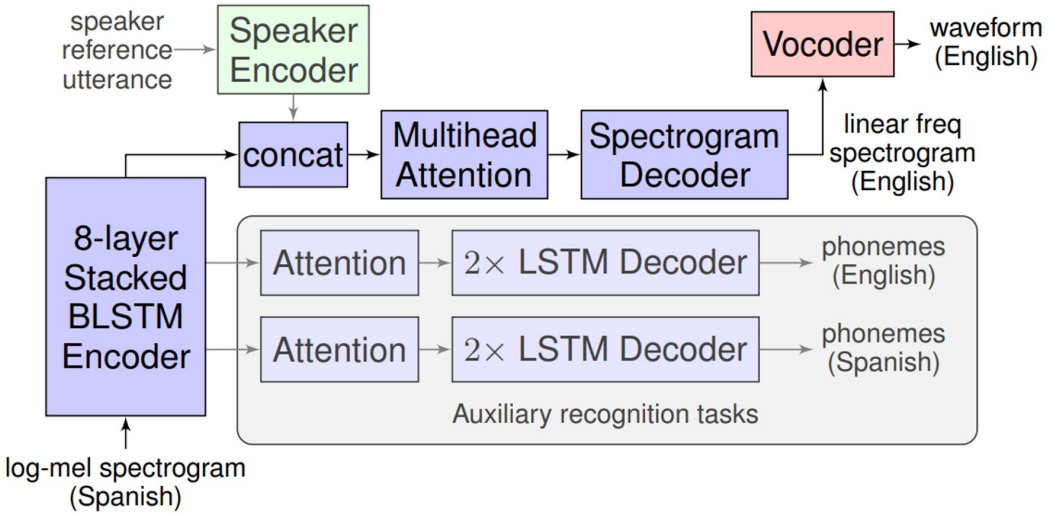


Figure 39. Multitask end-to-end speech-to-speech translation architectures [16].

The first end-to-end speech translation attempts were made by Ye et al. [16]. They used three individual decoders sharing the same encoder hidden states. They then apply multiple decoders step by step to these encoder states.

The first decoder generates a source language text given the encoder states, and then the decoder generates monotonic attention. This decoder helps the encoder to learn the alignment between source speech and text, and the encoder

itself can map the input speech to the hidden states while considering source text information. Next, the second decoder translates a target language text given the encoder states. Since the encoder states are already mapped considering source text alignment, the second decoder can easily find the corresponding encoder state. This second decoder helps the encoder to learn the source to target language text translation. Through this process, the encoder can map input speech to hidden states while considering the meaning of the target text. Finally, then the encoder states are mapped following the target text meaning; therefore, the third decoder can decode the target speech feature from the encoder states.

In this architecture, the three decoders work individually to process encoder states step by step, with each decoder producing encoder states for the next. However, this method has several problems. During training, it performs ASR, MT, and ST individually. Because of this, the decoders do not share their attention and decoding results. Instead, they only share the encoder states, and each decoder attends to the encoder states directly. Therefore, the third decoder needs to solve speech translation directly. In this way, the ASR and MT results provided by each decoder can not guarantee the quality of the speech translation. The first and second decoders are not necessary for testing. During the test step, the model only uses the source speech encoder and the target speech decoder. Thus, this model solves the speech translation task with a single attention module.

6.2 Proposed Transcoder-based Speech-to-Speech Translation

The proposed transcoder is extended transcoder method described in Section 4.2.2 to the end-to-end speech-to-speech translation task. While this transcoder model also has three decoders and learns speech translation step by step, it solves the speech translation task using the *combination* of these three decoders.

The first decoder generates a transcription of the source speech. The model aligns the input speech to the source text sequence using the attention module, and the decoder generates the source text sequence. Then the model passes a context vector to the next process. After this, the model uses the transcoder

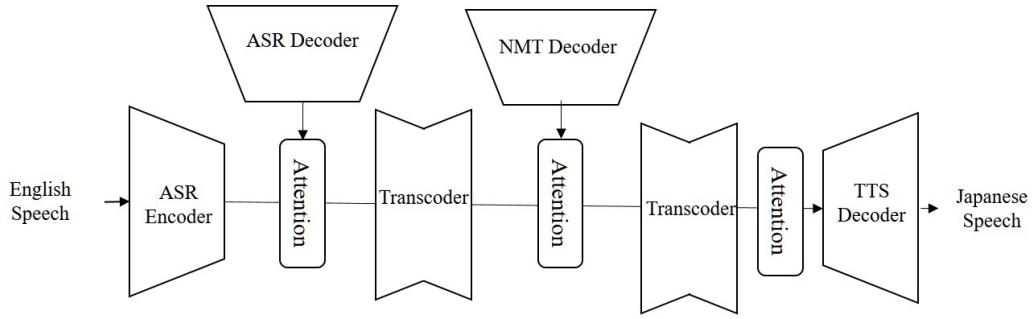


Figure 40. Proposed end-to-end speech-to-speech Translation architectures.

and the second decoder to solve the speech-to-text translation task. I solve the speech-to-speech translation task using a combination of the first decoder, the second decoder, and the transcoder. The first decoder helps to align the input speech sequence and source language text sequence. The second decoder helps to align the source text sequence and target language text sequence and passes the context vector to the next process. It then maps the second transcoder maps the context into pre-trained TTS hidden space. The third decoder attends the transcoder states to generate a target speech sequence. Here, the input context vectors are aligned with the target text sequence by the second decoder, meaning the third decoder can focus on generating speech features from the input context vectors. Each decoder passes its attention result on to the next process; thus, the second decoder does not need to concern itself with the speech-to-text alignment, and the third decoder does not need to solve the source-to-target text aliment again.

The benefit of my transcoder method is that the model solves speech translation using the combination of three attention modules. Each individual attention aligns the input to the target and provides the alignment result as a context vector. Thus, my proposed method can solve a difficult problem in the end-to-end setting. On the other hand, the multitask approach only uses one attention module to align input speech to target speech. This means that if the translation task becomes difficult, then the translation performance will drop significantly.

In this thesis, I construct pre-trained ASR, MT with acoustic embedding, and TTS. I also construct a multitask- and transcoder-based end-to-end speech-to-text translation model. First, I swap the end-to-end speech-to-text model's

second decoder to pre-trained MT decoder that using acoustic embedding. I tune the speech-to-text translation model. After the tuning, I extend the TTS decoder part to a multitask-based end-to-end speech-to-text model and perform total optimization. Also, I extend the second transcoder and the pre-trained TTS decoder to transcoder-based end-to-end speech-to-text model and apply my proposed training method.

In this thesis, I construct pre-trained ASR, MT with acoustic embedding, and TTS. I also construct a multitask- and transcoder-based end-to-end speech-to-text translation model. First, I swap the end-to-end speech-to-text model's second decoder to pre-trained MT decoder that using acoustic embedding. I tune the speech-to-text translation model. After the tuning, I extend the TTS decoder part to a multitask-based end-to-end speech-to-text model and perform total optimization. Also, I extend the second transcoder and the pre-trained TTS decoder to transcoder-based end-to-end speech-to-text model and apply my proposed training method.

6.3 Experiments for Direct Speech-to-Speech Translation on Distant Language Pairs

I conducted my experiments following Section 4.2.4 using the natural BTEC speech translation settings. I prepare the pre-trained Transformer based ASR, MT with multitask embedding and TTS, multitask- and transcoder-based speech translation. I utilize these models' parameters as the initial states. To prepare the parallel speech with the same speaker is very difficult; therefore, in this thesis, I use a generated speech as a target side. In this experiment, I translate input speech to target language speech using multitask- and transcoder-based models. I then transcribe the speech using ASR. Finally, I calculate a BLEU score from the transcription to evaluate each model's performance. I describe the translation performance in Table 33. I also show the attention tables for these three parts. The multitask-based model has three individual attention modules for ASR, MT, and speech translation, as shown in Fig. 41-43. From Fig. 41 I can find clear monotonic attention, then the MT and the speech translation have similar attention tables. In a multitask-based speech translation system, all

decoders sharing the same encoder states, the MT and speech translation decoder generate the same content in the same ordering. Thus the attention table has a reasonable shape as expected. On the other hand, the transcoder-based speech translation ASR and TTS have a monotonic shape; this means each attention focus on a different task and solve speech translation task in order. Speech translation is a challenging task; therefore, the approach that solves the problem step by step worked effectively and could outperform multitask-based speech translation, moreover the multitask-based speech translation, and MT performance are more different than transcoder's performance as shown in Table 33.

Table 33. Translation quality of speech-to-speech translation.

Model	BLEU score
Multitask-based speech translation *speechtranslation*	36.9
Multitask-based speech translation *MT*	40.8
Transcoder-based speech translation *MT*	42.1
Transcoder-based speech translation *speechtranslation*	40.6

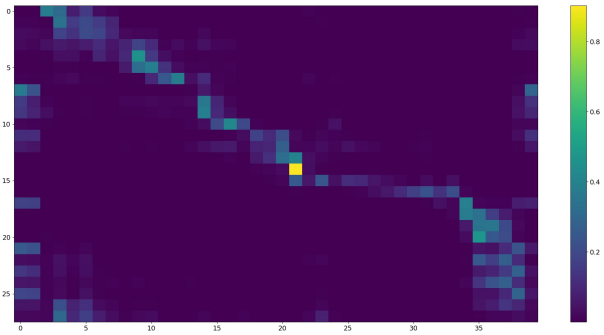


Figure 41. Attention table of Multitask speech translation's ASR part.

6.4 Summary

In this section, I compared the state-of-the-art speech-to-speech translation model and proposed model. The analysis showed that these two models solved speech

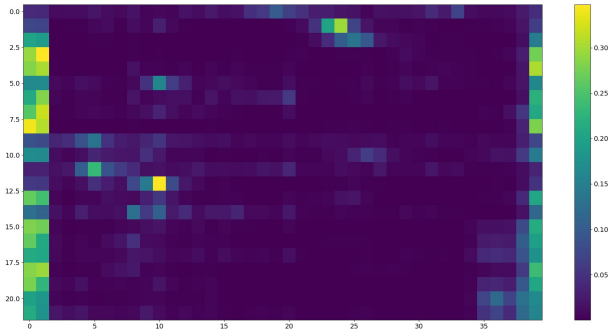


Figure 42. Attention table of Multitask speech translation's MT part.

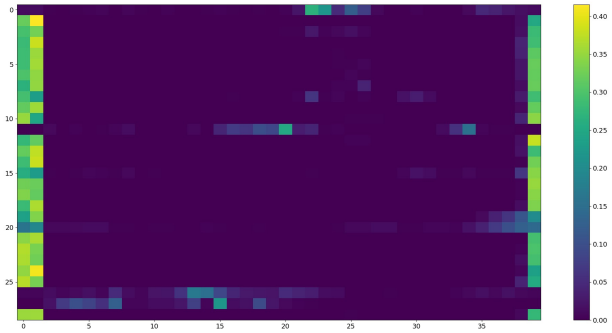


Figure 43. Attention table of Multitask speech translation's TTS part.

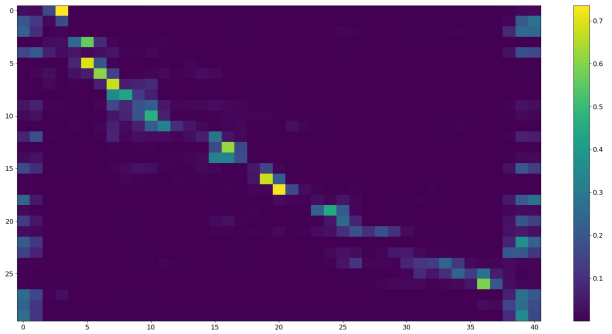


Figure 44. Attention table of Transcoder-based speech translation's ASR part.

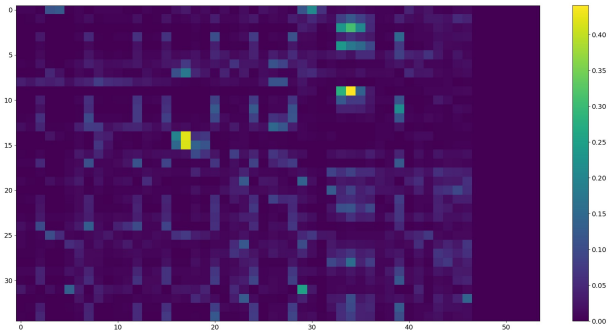


Figure 45. Attention table of Transcoder-based speech translation's MT part.

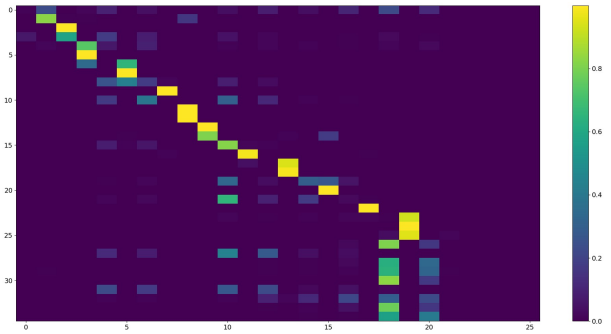


Figure 46. Attention table of Transcoder-based speech translation's TTS part.

to speech translation tasks in different approaches. The multitask task model's decoder only shared the source speech encoder states and decoding each target individually. Therefore, the output synthesis not strongly depended on the source and target decoder's performance. The source and target decoders worked for making good source speech encoder states for target speech decoder. On the other hand, the transcoder-based speech translation models passed the context vector of each source speech and text to the target text. Moreover, each transcoder used pre-trained MT and TTS encoder states as a target. Therefore the synthesis speech depended on the MT output than that in a multitask approach. The attention results also show that the transcoder approach following the traditional speech translation process could solve speech to speech translation tasks one by one.

7. Conclusions and Future Directions

7.1 Conclusions

In this thesis, I proposed four speech translation architectures.

First, I focused on the traditional cascade speech-to-text translation. I proposed the word posterior vector for ASR and MT information sharing. The ASR passed all candidate confidence scores to MT. Then MT could find the correct candidate and utilize the acoustic information to improve translation performance. This result shows, the speech translation system has the potential to outperform the text-based MT system.

Second, the proposed model focused on end-to-end speech-to-text translation. The model translated input source speech to target text directly. All related works performed the direct translation on syntactically similar language pairs (SVO to SVO word order). The proposed model focused on translating syntactically distant language pairs (SVO to SOV word order). Syntactically distant language pairs translation is a challenging task. To solve this task with an attention-based encoder-decoder model, I proposed a transcoder-based speech translation model.

This model has a more deep structure than a standard attention-based encoder-decoder model. Therefore I proposed a training strategy inspired by CL. I compared the proposed model and other models in several related works using BTEC and TED talk translation tasks. The proposed model showed excellent performance in both translation tasks. Moreover, the proposed training strategy helped the model learning on a small dataset. The third approach first focused on end-to-end text-to-speech translation. I used pre-trained TTS embedding parameters as MT target output layer parameters. In this approach, I forced the model to decode the target text by considering the target language's acoustic information. I put a limitation on the MT target text generation process by utilizing TTS embedding. It made the MT model sensitive to the meaning and pronunciation of the target sequence, and the generated result is more accurate to reference. Finally, the proposed model outperformed the state-of-the-art MT translation model on BTEC translation experiments.

Finally, I proposed an end-to-end speech-to-speech translation model for syntactically distant language pairs. I compared my transcoder-based model with

state-of-the-art end-to-end speech-to-speech translation model. The proposed model outperformed the state-of-the-art model in translation performance, and the proposed model showed the translation process was different from the state-of-the-art model. It demonstrates the proposed model could solve the complex speech translation task one by one following the traditional speech translation system in the end-to-end framework.

7.2 Future Directions

I describe the speech translation research road map in Fig. 47.

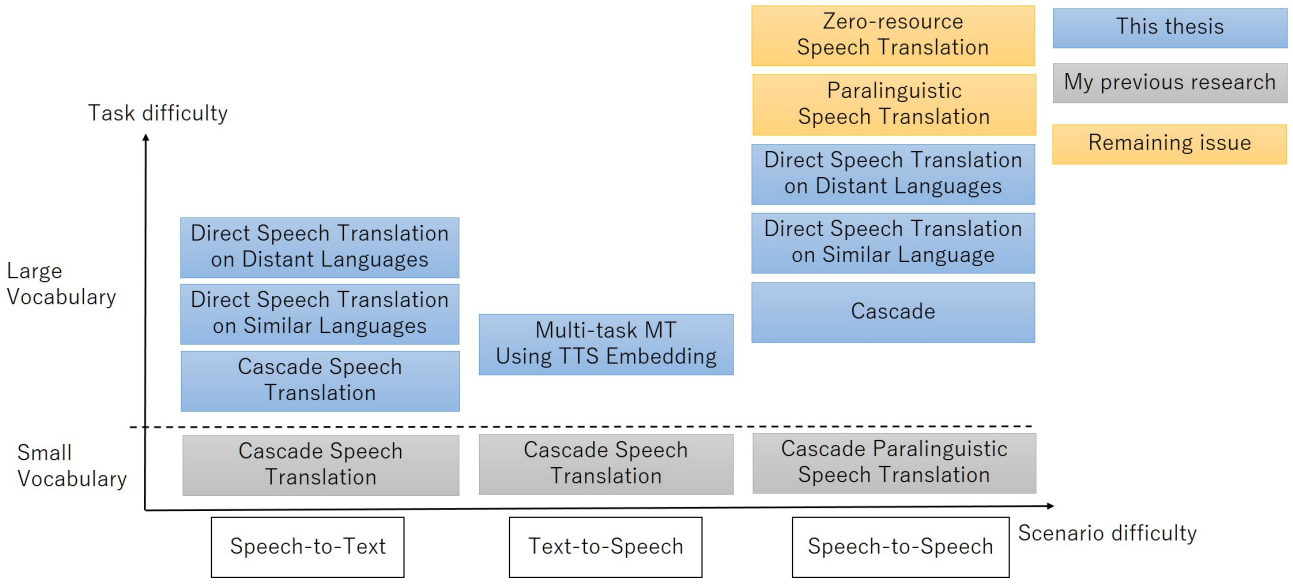


Figure 47. A road map for end-to-end speech translation.

In this thesis, I built an end-to-end speech-to-speech translation model to improve speech translation quality. My model outperformed existing end-to-end models on the BTEC and TED corpora. The proposed approach also demonstrates that speech translation can outperform text translation in some cases. However, performing speech translation on TED talk data proved to be difficult, and should be further investigated. For this, I will apply other speech and parallel text datasets to the ASR and MT as an additional training procedure for the TED talk task, in order to improve both models' initial states. Also, preparing parallel speech data is expensive. I, therefore, plan to use a semi- or un-supervised approach, e.g., speech chain [32] or CycleGAN [38].

The speech sometimes features a variety of paralinguistic information. In this thesis, my proposed model can translate the input speech to target speech in an end-to-end fashion. This means my model has the potential to translate speech while taking into account paralinguistic information. However, in my experiments, I have only used generated speech that does not include paralinguistic

information as a target speech. In future work, I will perform speech translation on speech that features specific paralinguistic information, while attempting to transfer this information in the translation process. Initially, I intend to do this for speech emphasis.

Finally, I proposed an end-to-end speech-to-speech translation model for syntactically distant language pairs. I compare my transcoder-based model with state-of-the-art end-to-end speech-to-speech translation model. My proposed model outperformed the state-of-the-art model in translation performance, and my model shows the translation process is different from the state-of-the-art model. It demonstrates my proposed model can solve the complex speech translation task one by one following the traditional speech translation system in the end-to-end framework.

Acknowledgements

Here, I thank many people who have helped me to complete this thesis. My supervisors, Professor Satoshi Nakamura and Assistant Professor Sakti Sakriani are excellent teachers. They have given me both great advice and a lot of discussion for my research. I can enjoy my research under their supports. They have taught me a lot about what research is, how to concern, and how to write papers. I also want to thank Associate Professor Katsuhito Sodoh, Assistant Professor Koichiro Yoshino, Assistant Professor Hiroki Tanaka, and all members in my AHC lab for discussion and advice. I would also like to thank my thesis committee, Professor Satoshi Nakamura, Professor Laurent Besacier, Professor Yuji Matsumoto, Professor Tsukasa Ogasawara, Assistant Professor Sakti Sakriani. I also thank my friends Andros Tjandra, Seitaro Shinagawa, and Marco Vetter. I also want to thank my NAIST lab assistant Ms. Manami Matsuda for helping my life in AHC lab. Finally, I want to thank my parents. They always support me.

References

[1] Pablo Daniel Agüero, Jordi Adell, and Antonio Bonafonte. Prosody generation for speech-to-speech translation. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2006, Toulouse, France, May 14-19, 2006*, pages 557–560, 2006.

[2] Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[4] Sameer Bansal, Herman Kamper, Adam Lopez, and Sharon Goldwater. Towards speech-to-text translation without speech recognition. *CoRR*, abs/1702.03856, 2017.

[5] Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. End-to-end automatic speech translation of audiobooks. *CoRR*, abs/1802.04200, 2018.

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

[7] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734, 2014.

[8] 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, December 7-12, 2015, Montreal, Quebec, Canada, pages 577–585, 2015.

[9] Quoc Truong Do, Sakriani Sakti, Graham Neubig, and Satoshi Nakamura. Transferring emphasis in speech translation using hard-attentional neural network models. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 2533–2537, 2016.

[10] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 5884–5888, 2018.

[11] Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. An attentional model for speech translation without transcription. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 949–959, 2016.

[12] Marcello Federico, Luisa Bentivogli, Michael Paul, and Sebastian Stüker. Overview of the IWSLT 2011 evaluation campaign. In *2011 International Workshop on Spoken Language Translation, IWSLT 2011, San Francisco, CA, USA, December 8-9, 2011*, pages 11–27, 2011.

[13] Daniel W. Griffin and Jae S. Lim. Signal estimation from modified short-time fourier transform. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '83, Boston, Massachusetts, USA, April 14-16, 1983*, pages 804–807, 1983.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[16] 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. *CoRR*, abs/1904.06037, 2019.

[17] Takatomo Kano, Shinnosuke Takamichi, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura. Generalizing continuous-space translation of paralinguistic information. In *INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France, August 25-29, 2013*, pages 2614–2618, 2013.

[18] Gen-ichiro Kikui, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto. Creating corpora for speech-to-speech translation. In *8th European Conference on Speech Communication and Technology, EUROSPEECH 2003 - INTERSPEECH 2003, Geneva, Switzerland, September 1-4, 2003*, 2003.

[19] Gen-ichiro Kikui, Seiichi Yamamoto, Toshiyuki Takezawa, and Eiichiro Sumita. Comparative study on corpora for speech translation. *IEEE Trans. Audio, Speech & Language Processing*, 14(5):1674–1682, 2006.

[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[21] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, 2007.

[22] 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, Edmonton, Canada, May 27 - June 1, 2003, 2003.

[23] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu, and Ming Zhou. Close to human quality TTS with transformer. *CoRR*, abs/1809.08895, 2018.

[24] Yifan Liu and Jin Zheng. Es-tacotron2: Multi-task tacotron 2 with pre-trained estimated network for reducing the over-smoothness problem. *Information*, 10(4):131, 2019.

[25] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[26] Preslav Nakov, Francisco Guzmán, and Stephan Vogel. Optimizing for sentence-level BLEU+1 yields short translations. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 8-15 December 2012, Mumbai, India*, pages 1979–1994, 2012.

[27] 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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318, 2002.

[28] Nikolaos Pappas, Lesly Miculicich Werlen, and James Henderson. Beyond weight tying: Learning joint input-output embeddings for neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 73–83, 2018.

[29] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[30] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. Attention-passing models for robust and data-efficient end-to-end speech translation. *TACL*, 7:313–325, 2019.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112, 2014.

[32] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. End-to-end feedback loss in speech chain framework via straight-through estimator. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 6281–6285, 2019.

[33] 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, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[35] 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: A fully end-to-end text-to-speech synthesis model. *CoRR*, abs/1703.10135, 2017.

[36] 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, Stockholm, Sweden, August 20-24, 2017*, pages 4006–4010, 2017.

[37] 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, Stockholm, Sweden, August 20-24, 2017*, pages 2625–2629, 2017.

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2242–2251, 2017.

Publication List

Peer review journal paper

1. Takatomo KANO, Shinnosuke TAKAMICHI, Sakriani SAKTI, Graham NEUBIG, Tomoki TODA, Satoshi NAKAMURA. An end-to-end model for cross-lingual transformation of paralinguistic information. Machine Translation. Vol. 3 Page 353-368.

2. Takatomo KANO, Sakriani SAKTI, Satoshi NAKAMURA. End-to-end Speech Translation by Multi-task Learning with Transcoding for Distant Language Pairs. IEEE Transactions on Audio, Speech, and Language Processing. Under review.

Peer review international conference

1. Takatomo KANO, Sakriani SAKTI, Satoshi NAKAMURA. Neural Machine Translation with Acoustic Embedding. The 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2019).

2. Takatomo KANO, Sakriani SAKTI, Satoshi NAKAMURA. Structured-Based Curriculum Learning for End-to-End English-Japanese Speech Translation. INTERSPEECH 2017.

3. Takatomo KANO, Shinnosuke TAKAMICHI, Sakriani SAKTI,Graham NEUBIG,Tomoki TODA, Satoshi NAKAMURA. Generalizing continuous-space translation of paralinguistic information. INTERSPEECH 2013.

4. Kaho OSAMURA, Takatomo KANO, Sakriani SAKTI, Katsuhito SUDOH, Satoshi NAKAMURA. Using Spoken Word Posterior Features in Neural

Machine Translation. International Workshop on Spoken Language Translation (IWSLT 2018).