Master's Thesis

Neural Incremental Speech Recognition Towards Simultaneous Speech Translation

NOVITASARI Sashi
Program of Data Science
Graduate School of Science and Technology
Nara Institute of Science and Technology

Supervisor: Professor Satoshi Nakamura
Augmented Human Communication Lab. (Division of Information Science)

Submitted on 16/09/2020

A Master's Thesis
submitted to Graduate School of Science and Technology,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
MASTER of ENGINEERING

Sashi Novitasari

Thesis Committee:

Professor Satoshi Nakamura
(Supervisor, Division of Information Science)
Professor Taro Watanabe
(Co-supervisor, Division of Information Science)
Research Associate Professor Sakriani Sakti
(Co-supervisor, Division of Information Science)

Neural Incremental Speech Recognition Towards Simultaneous Speech Translation*

Sashi Novitasari

Abstract

Simultaneous speech interpretation or translation is a required task to bridge a real-time multilingual human-to-human communication. Speech-to-speech translation (S2ST) system attempts to mimic human interpreters to translate a speech. As a pipeline, S2ST system consists of three components: automatic speech recognition (ASR), machine translation (MT), and text-to-speech synthesis (TTS) systems. Unlike human interpreter who can do a simultaneous interpretation, conventional S2ST system costs a high output latency because the data passing between the components is done based on the complete input and output sequences. To enable automatic speech translation in a real-time situation, a simultaneous S2ST system that works within a low delay is required.

Among all components, ASR has a critical role to determine the performance and delay of a simultaneous S2ST system in the first place. Despite its remarkable performance, the *state-of-the-art* attention-based neural ASR costs a high recognition delay because of the global attention mechanism. As a result, it cannot be used for a simultaneous S2ST task. Several studies recently proposed the sequence mechanisms for incremental speech recognition (ISR) that produces the output within a low delay. To do a low delay recognition, ISR model needs to decide the incremental steps to begin or end the recognition of a short part of the speech input. For this reason, the existing neural ISR systems use a more difficult training mechanism and framework than the standard non-incremental neural ASR.

*Master's Thesis, Graduate School of Science and Technology, Nara Institute of Science and Technology, September 16, 2020.

Towards simultaneous S2ST, this thesis focuses on addressing the current ISR problem by investigating whether possible to perform ISR and handle the input segmentation without introducing a complicated training mechanism. By using the standard neural ASR as the base, the challenges here are to (1) reducing recognition delay, (2) keeping system complexity, and (3) maintaining recognition performance.

We perform two tasks to achieve our goal. First, as our proposal, we construct a neural ISR system using attention transfer from a standard neural ASR model with an identical structure. Transfer learning is a method to train a student model by using the knowledge that a teacher model has. In our proposed method, we treat the standard neural ASR as a teacher model that transfers its attention-based knowledge to an ISR model, the student. Our experiments show that the proposed ISR with a recognition delay of 0.54 sec can achieve a close performance to the teacher model whose delay is more than 6 sec. In the second task, we utilize the proposed ISR in speech translation task and see how ISR affect MT performance. We explored various approaches to adapt the ISR output in accordance with MT input unit to achieve a good translation performance. Our experiments on English-French translation task show that end-to-end ISR with the matching subword representation as MT input side achieves the best speech recognition and translation performance.

Keywords:

attention transfer, incremental speech recognition, simultaneous speech-to-speech translation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Speech-to-speech Translation for Multilingual Conversation

Speech is one of tools that humans use to express their thoughts and communicate with others. In this era, as globalization rapidly expands, language barriers continue to be the most notorious restriction on free communication among different language speakers. One of the ways to break those barriers is by learning the languages by ourselves. However, as there are many languages in this world, learning all languages is impossible for a human. Another way is by asking another person, an interpreter, who understands the language to translate it for us.

Human interpreter is able to break the language barrier by translating a speech into another language that can be understood by the listener. They perform the task simultaneously to the timing of the source speech, so the source speaker and the listener can do a direct communication even though they speak in different languages. Despite the high demand, interpretation is a complex skill that takes many years to master, therefore, using a professional interpretation service can be expensive. The availability of language pairs also remains scarce.

The automation of speech translation to bridge multi-language communication can be done with speech-to-speech translation (S2ST) system. S2ST technology [8], in the other words automatically recognizing a speech and translating

it into a speech in a different language, is an innovative technology that can support many everyday situations that relate to multi-language communication. S2ST system consists of three components: automatic speech recognition (ASR), machine translation (MT), and text-to-speech synthesis (TTS) systems. In the conventional pipeline S2ST system, as shown in Figure 1.1(a), the three components are interconnected and each of them processes the output of the previous adjacent component.

S2ST system process begins with the source speech recognition by the ASR system. In the conventional system, ASR system performs a recognition process on a completed speech and passes the output to the MT system. MT system does the translation textually from the ASR output into another language [9] and then passes the translation result to the TTS system. By taking the MT's completed output, TTS system synthesizes a speech signal, which speech has the same meaning as the source speech but in a different language [10]. Not only the pipeline system but the recent studies also took an interest in end-to-end S2ST systems [11, 12, 13] that perform all processes within a single model.

The conventional pipeline and end-to-end S2ST systems, however, cost a long processing delay. These systems need to wait for a completed speech to begin the process. As a result, the longer the source speech is, the longer the waiting time will be. Consequently, such systems are not practical in situations where the time delay between the source speech and translation speech is critical. The example of such a situation is real-time lecture translation, where the lecturer's speech and its translation should be delivered at the same time so the audience can follow the lecture. A solution to this problem is a real-time or simultaneous S2ST system that mimics human interpreters, who are able to recognize and translate the speech simultaneously to the time when the speech is uttered.

1.2. Simultaneous Speech-to-speech Translation

Simultaneous S2ST system is a system that automatically translates a speech and delivers the output as a speech in a different language, where the time delay between the start timing of the source and the output speech is very small. This system aims to do the speech translation in a similar way to the human

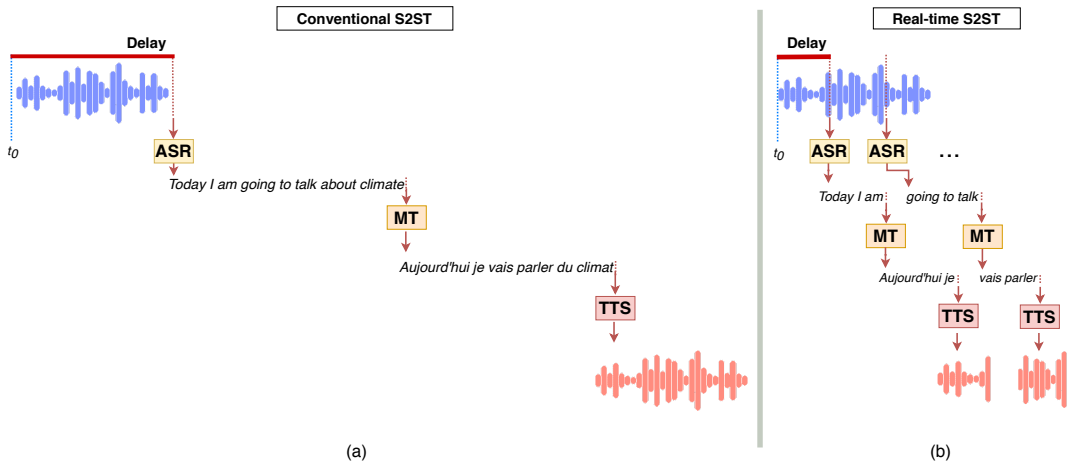


Figure 1.1. Pipeline S2ST systems: conventional framework (a) and real-time or simultaneous framework (b). Examples in English-French translation task.

interpreter, who does the translation simultaneously to the source speech [14, 15, 16].

Simultaneous S2ST system needs to perform low-latency processing in the ASR that is followed by a low-latency MT and a low-latency TTS. An illustration of the simultaneous S2ST system is shown in Figure 1.1(b). Simultaneous S2ST system consists of the same components as the conventional S2ST system. The main difference between the conventional and the simultaneous S2ST systems lies in the starting condition of each component's process. In the conventional system, each component has to wait for the complete output from the previous component. On the other hand, the simultaneous S2ST system does not limit the components to wait for a complete output from the previous one. They just wait for a part of the input, instead of a complete input, and works on the fly.

The performance of the simultaneous S2ST task is affected by the output delivery speed and the translation quality. Output delivery speed corresponds to the delay or lag that occurs during the speech translation process. Here delay is the time difference between the start time of the source speech and the initial time when the system produces the output[17]. The minimum S2ST delay by the conventional system (Figure 1.1(a)) equals the length of the complete source speech. On the other hand, the minimum delay in the simultaneous system

(Figure 1.1(b)) equals the size of the first-recognized speech segment, which is shorter than the conventional system's delay. The actual delay also includes the computational delay.

For human interpreters, translation delay can cause an impact on their performance. In many situations, speech translation with a short delay is preferable. A short translation delay is able to relax the target listeners and facilitate the communication between the original speaker and the target listeners [14]. In the simultaneous speech translation of English speech by human interpreters, the delay generally ranges from two to six seconds [18, 19], or roughly about four to twelve words [20, 21]. Translation with a short delay is also can be beneficial for human interpreters because it can reduce their short-term memory burden. On the other hand, a long term information can provide clearer information about the message in the source speech, better than the short term information, so the understanding of it for translation will be better [22], even though the waiting delay is long. In the case of human interpreters, translation with a long delay may burden their working memory so the translation quality may also decrease in some cases. But unlike the human interpreters, memory load is not a vital issue in automatic speech interpretation by machine. A machine, however, cannot understand the speech well like human does, unless the model is trained using a large amount and variety of data.

1.3. Automatic Speech Recognition for Simultaneous Speech-to-speech Translation

One challenge to achieving simultaneous S2ST technology is the construction of low-delay ASR or incremental ASR (ISR). ASR serves as the foremost component in the S2ST system, therefore, the S2ST system's delay and accuracy are highly affected by the ASR system. The S2ST system will not able to do simultaneous speech translation if it has to wait for a completed speech to begin the ASR's process. Therefore, a low-delay ASR or ISR system that is able to recognize speech immediately as the speech start is necessary for a full-fledged simultaneous S2ST system.

1.3.1 Existing Approach

Researchers have been working on speech recognition technology for decades. The current *state-of-the-art* ASR systems are end-to-end neural ASR systems that consist of neural network framework architecture with encoder and decoder with an attention module [1, 23, 24, 25]. Many works reported the excellency of attention-based ASR. Despite it, standard attention-based ASR may result in a long recognition delay. The standard attention mechanisms are based on a global attention property that requires the computation of a weighted summarization of the entire input sequence that is generated by the encoder states. This means the system can only generate the speech transcription after receiving the complete speech sequence as its input. Therefore, standard attention-based ASR is not suitable for simultaneous S2ST task.

Among the existing ASR works, several approaches are capable to recognize a speech utterance within a low delay without waiting for the end of the speech. The descriptions of the existing low-delay ASR frameworks are the following.

- **Conventional Hidden Markov Model ASR**
  Hidden Markov model-based (HMM) is a classic framework for large vocabulary continuous speech recognition system [26, 27, 28]. HMM-based ASR consists of three separate components: acoustic model, lexicon, and language model. The acoustic model predicts the phoneme sequence from the input speech by modeling the speech features distribution using a Gaussian Mixture Model (GMM) and finding the optimal phoneme sequence using HMM. The second component, lexicon, consists of word-to-phonemes dictionary. This component proposes words that contains the phonemes that are recognized by the acoustic model. From a phoneme sequence, multiple words can be proposed because some words may consist of the same phoneme sequence. From the words candidates, the final word sequence is decided by the language model by maximizing the word sequence probability score. Basic HMM-based ASR works in a left-to-right transition structure, where the recognition is done from the earliest speech features and can be done without waiting for the end of speech. Therefore, HMM-based ASR is able to produce output within a low delay to the speech utterance.

- **Recurrent Neural Network ASR with Connectionist Temporal Classification Training**

  Recurrent neural network (RNN) ASR based on connectionist temporal classification (CTC) training objective is one of the end-to-end ASR that models the mapping of speech features into text token directly. [29]. The modeled text token is generally character-level tokens. The recognition with an RNN network is done by predicting a corresponding token for each speech feature frame. However, the length of speech frame sequence is usually significantly longer than its transcription. Because of it, outputting one token for each recurrent step may result in a bad transcription. In this framework, this problem is solved by using a CTC objective during the model training. Here CTC [30] enables the RNN to decide whether to output a token or not, so the ASR will not output a long and incorrect transcription. An ISR system using this approach can be made by using unidirectional RNN across the model layers. ISR with unidirectional RNN-CTC process the speech frames starting from the first frame unidirectionally and does not depend on the future frames, so it only requires a short time to recognize the speech incrementally [3]. The details of unidirectional RNN-CTC for ISR task can be seen in Appendix B.1.

- **Neural Transducer ASR**

  Neural transducer (NT) ASR [4, 31] is a low-delay end-to-end ASR framework based on neural network that incorporates attention mechanism. The structure of NT consists of an encoder with unidirectional RNN-type of network and a transducer that predicts the output. NT ASR recognizes the speech segment-by-segment with a fix-sized window. The segment-based recognition is learned by looking at segment-level alignment during the model training process. There are two existing methods for alignment generation. In the first method, segment-level alignment is generated by performing forced-alignment using HMM-GMM ASR. In the second method, the alignment computation is done during the model training by applying a dynamic programming type of method based on the model state and output probability. The computation and alignment update are done multiple times to get the approximately best alignment according to the model qual-

ity at that time. The details of the NT framework can be seen in Appendix B.2.

1.3.2 Challenges to Overcome

As we discussed before, simultaneous S2ST system firstly depends on the ISR system. The challenge of ISR mainly lies in the mechanism to do the incremental step. In an incremental step, given an unfinished speech utterance, ISR must decide the input boundary and output boundary on the fly to produce a transcription. Figure 1.2 shows the illustration of these boudaries. Input boundary and output boundary represent a pair of speech part and transcription part that aligns with. ISR can begin the recognition and generate some outputs if they find an input boundary. If the ISR finds an output boundary during output generation based on a certain speech part, the ISR can stop the generation and move to the next speech part. These boundaries enable ISR to do a partial input processing so it can produce an output immediately after the speech starts.

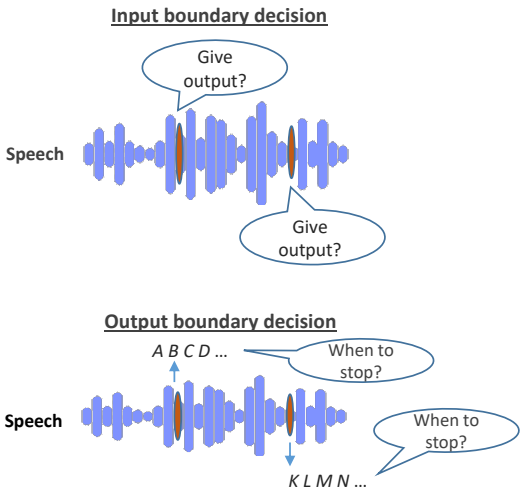


Figure 1.2. Input boudary decision and output boudary decision in incremental speech recognition.

Because of the nature of the ISR task, most available ISR systems use different frameworks and learning algorithms that are more complicated than the standard *state-of-the-art* ASR model. HMM-GMM ASR recognizes a speech utterance

incrementally, but it is not an end-to-end framework. A non-end-to-end ASR system is known for the difficulty to optimize because each system component has an individual objective function and they are not optimized jointly [32]. The RNN-CTC ASR typically has a lower performance than attention-based ASR [33]. Among the ISR approaches that we mentioned above, NT ASR has the closest inference mechanism to the standard neural ASR, however, it has a more complicated mechanism to train than the standard model. NT ASR learns the incremental step by learning the transcription that is aligned with the current short speech segment. The complicated training mechanism is mainly caused by the alignment preparation process. The existing approaches for alignment generation are alignment generation using an HMM-GMM system and online alignment computation during NT ASR training. The first approach requires us to create an HMM-GMM ASR that can be difficult to be optimized, while the second approach requires a high time and computation complexity to create the ISR system. To simplify the simultaneous S2ST system development, an ISR with a similar approach to the standard ASR and with a less complicated construction mechanism than the existing frameworks is required.

To create a simultaneous S2ST system, not only the ISR performance as an individual system but we also must consider the usability of it in the translation task. In simultaneous S2ST system, ISR acts as an initiator and it will determine how MT works. Here we consider the MT as a neural MT that does the recognition in a sequence-to-sequence manner by taking a sequence of tokens. To enable the utilization of ISR for speech translation tasks, firstly, MT must be able to recognize the tokens that are produce by ISR. Generally in an utterance-based end-to-end S2ST system, such a factor receives less attention since all of the processes are done by a single model. An end-to-end simultaneous S2ST system, however, remains as a challenge especially for translations between languages whose syntaxis word orders are different. As a step to achieve a simultaneous S2ST system, we must enable the ISR to be utilized in speech translation task and investigate the impact of ISR on MT to achieve a good speech translation performance.

1.4. Thesis Objective and Contribution

In this thesis, we focus on neural ISR construction that is intended as a component of simultaneous S2ST system. The ISR is aimed to meet the following conditions:

- Reduce the recognition delay of the standard neural ASR.
- Use a similar mechanism to the standard neural ASR to keep the system complexity.
- Maintain a close performance to the standard neural ASR.

After the neural ISR construction, we utilize the ISR in speech translation tasks. The ISR optimization is also done to optimize the quality of the translation result from the ISR text.

To do an incremental recognition, ISR needs to learn the incremental step. We propose to teach the ISR to do the incremental step by using the attention-based knowledge from a standard neural ASR system. We keep the ISR system complexity by retaining the structure of a standard neural ASR system. For these factors, we consider the proposed ISR construction approach as a teacher-student learning task. The ISR acts as a student model, while the teacher is a standard neural ASR. The attention-based information, where the ISR learns from, is generated by the teacher once before training the ISR model. As those models have an identical structure, the ISR can use the teacher's hyperparameter as it is. Accordingly, the construction of ISR only requires an attention-based ASR and does not rely on an external system or excessive alignment computation.

Taking a look at our discussion in Section 1.3.2, we are not only considering the ISR performance as an independent component but also as a system that is connected to an MT system. To achieve that, we explore several approaches to adapt the ISR output so MT can recognize the tokens that are produced by ISR to do speech translation.

To summarize the objectives, this thesis has three main contributions.

1. ISR
    - Developing neural ISR by employing sources from standard neural ASR.

- Investigating the impact of delay to speech recognition performance.

2. ISR in speech translation task

   - Utilizing neural ISR in speech translation task by designing ISR-MT integration framework to improve speech translation performance.
   - Investigating the impact of low delay speech recognition to the translation task.

As we only focus on neural ISR, this work does not include the study for incremental MT and TTS components for simultaneous S2ST. The speech translation experiments were done using a non-incremental neural MT system.

1.5. Thesis Overview

This thesis is organized as described below.

In Chapter 2, we discuss the standard end-to-end neural ASR that consists of encoder and decoder with an attention module. The discussed framework here is our basis to create the proposed neural ISR. Here we describe the architecture of the standard ASR and the mechanisms to train and use the model.

In Chapter 3, we introduce the proposed neural ISR system. The description includes the system construction method and usage method. After that, we describe the speech recognition experiment that we conduct using our neural ISR.

In Chapter 4, we start with the description of the important factors that we have to consider to utilize the non-incremental ASR or ISR in speech translation task. Our discussion is followed by the introduction of proposed methods to connect the ISR and MT in a pipeline system. Finally, we describe the experiments that we conduct by applying ISR on speech translation tasks of English-French and English-Japanese.

Chapter 2

Neural Automatic Speech Recognition

End-to-end neural ASR framework that we used in this work has a sequence-to-sequence neural network structure, which consists of an encoder and decoder components with an attention mechanism [1, 34]. Given a sequence of framed speech features $\mathbf{X} = [x_1, x_2, x_3, ...x_S]$ with a length of S, a neural ASR model is trained to predict the speech's transcription text $\mathbf{Y} = [y_1, y_2, y_3, ...y_T]$ with a length of T by directly modeling the conditional probability in Equation 2.1. In the basic framework, $\mathbf{Y}$ is a sequence of character units.

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{T} P(y_t|\mathbf{X}, y_{<t}) \tag{2.1}$$

2.1. Components

Figure 2.1 shows an overview of the standard neural ASR system with a sequence-to-sequence structure. A neural sequence-to-sequence ASR consists of encoder and decoder components with an attention mechanism. Each component consists of a neural network structure, where all components are trained jointly. The components' joint training in the sequence-to-sequence model diminishes the optimization problem that is faced by a non-end-to-end ASR system. The details of each component are described as follows.

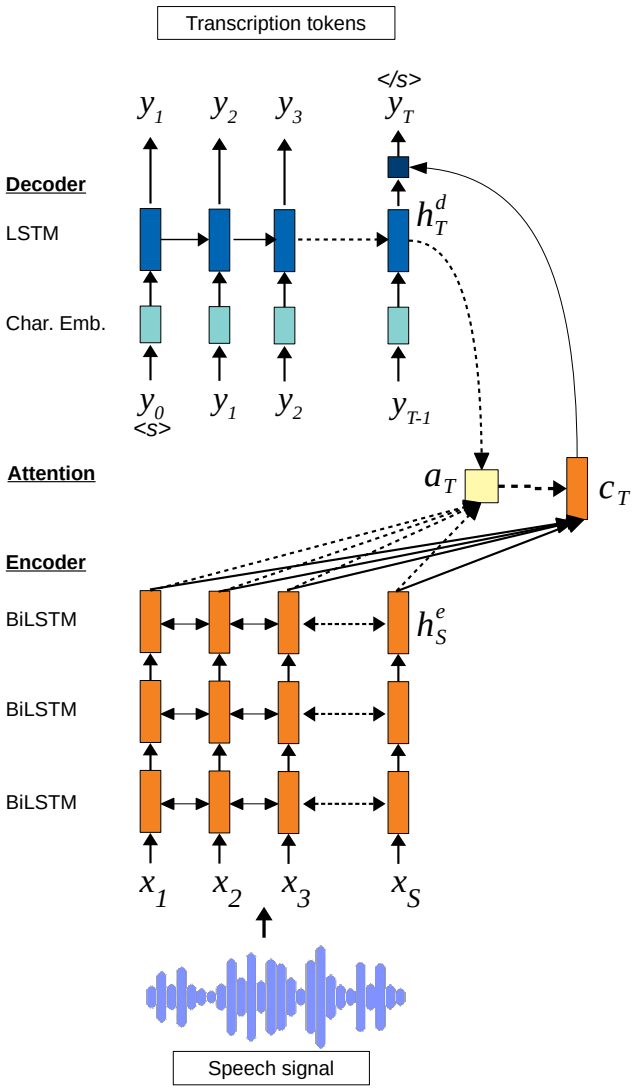


Figure 2.1. Neural ASR with encoder and decoder components with an attention module [1, 2].

2.1.1 Encoder

Encoder transforms the speech features sequence $\mathbf{X}$ with a length S into hidden representation $\mathbf{h}^e$. We can consider this component as a 'listener' to the speech. Encoder consists of stacked RNN layers. The RNN type that commonly used is bidirectional long-short term memory (BiLSTM) RNN [35, 36].

In typical deep BiLSTM structures, the output of j-th BiLSTM layer at the i-th timestep is computed by using Equation 2.2.

$$h_i^j = BiLSTM(h_{i-1}^j, h_i^{j-1}) \tag{2.2}$$

After the encoding process finishes, the encoder hidden representation $\mathbf{h}^e$ is decoded by the decoder to predict the output. Suppose that the encoder consists of J layers in total, the $\mathbf{h}^e$ that will be decoded is the $\mathbf{h}^j$ where the index $j = J$.

Since the length of a framed speech features sequence can be very long, it can cause the encoder to converge very slow and unable to achieve the optimum performance. This is because the relevant information extraction from a long sequence is difficult. This problem commonly confronted by applying hierarchical sub-sampling [1, 37, 38] to the BiLSTM layers in the encoder. The sub-sampling reduces the sequence's time resolution by a factor as it proceeds to the higher layer in the stack. It is done by concatenating some consecutive outputs in the previous layer to calculate the output of the target layer. An example of encoder that applies hierarchical sub-sampling with a factor of two for each layer can be seen in Figure 2.2. In this figure, the length of hidden representation sequence $\mathbf{h}^j$ in the encoder layer $j = 3$ is two states, which are reduced from eight input unit. When hierarchical sub-sampling is applied, for example, sub-sampling by a factor of two for each layer, the i-th output computation in j-th BiLSTM layer may follow Equation 2.3.

$$h_i^j = BiLSTM(h_{i-1}^j, [h_{2i-1}^{j-1}, h_{2i}^{j-1}]) \tag{2.3}$$

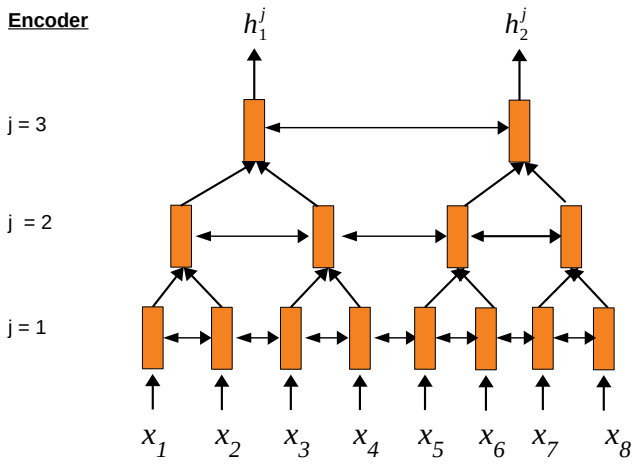


Figure 2.2. Example of encoder structure with hierarchical sub-sampling.

2.1.2 Decoder

Decoder predicts the transcription of speech as a sequence of text tokens. This component can be considered as a 'speller'. Decoder in ASR model consists of an embedding layer and unidirectional long-short term memory (LSTM) RNN layer. To predict the token sequence $\mathbf{Y}$ that has a length T, for each timestep t, decoder produces a probability distribution over the next token conditioned on previous outputs $\mathbf{Y}_{<t}$ based on the current context information c_t and current decoder hidden state h_t^d. The context information at time t (c_t) is computed by attention module [39], while decoder hidden states at time t (h_t^d) are computed by processing the previous output token using the embedding layer and LSTM layer. The probability distribution computation for the t-th output token follows Equation 2.4, where it results on a vector with the same size as the token vocabulary size.

$$P(y_t|\mathbf{X}, y_{<t}) = \mathit{Token\_Distribution}(c_t, h_t^d) \tag{2.4}$$

2.1.3 Attention

Attention module computes the contextual information from the input speech during the decoding process. The context information tells the decoder which

part of the input sequence that it has to attend to generate an output token. Context information c_t at time t is computed with the following Equation 2.5 and Equation 2.6 by referring to Figure 2.1.

$$c_t = \sum_{s=1}^{S} a_t(s) * h_s^e \tag{2.5}$$

$$a_t(s) = \frac{exp(Score(h_s^e, h_t^d))}{\sum_{s=1}^{S} exp(Score(h_s^e, h_t^d))} \tag{2.6}$$

The scoring for a context is commonly done using one of the functions in Equation 2.7 [40].

$$Score(h_s^e, h_t^d) = \begin{cases} \langle h_s^e, h_t^d \rangle, & \text{dot product} \\ h_s^{e\intercal} W_s h_t^d, & \text{bilinear} \\ V_s^\intercal \tanh(W_s[h_s^e, h_t^d]) & \text{MLP,} \end{cases} \tag{2.7}$$

$Score$ is a $(\mathbb{R}^M \times \mathbb{R}^N) \rightarrow \mathbb{R}$ function, where M is the number of encoder hidden units and N is the number of decoder hidden units.

2.2. Training Method

Sequence-to-sequence neural network model training is commonly done using a teacher-forcing training strategy [41]. This strategy trains the sequence-to-sequence model to predict an output by feeding the correct output from the previous timestep into the decoder. Figure 2.3 illustrates the decoding mechanism with teacher-forcing. It does not use the predicted output token as the decoder input in the next timestep. Teacher forcing allows the model to converge fast and keeps the model stability during training.

The training loss is computed based on the tokens probability distribution computed by the model and the correct tokens. Training loss computation for each speech utterance recognition follows Equation 2.8, where C is the number of output class or the number of tokens in the vocabulary.

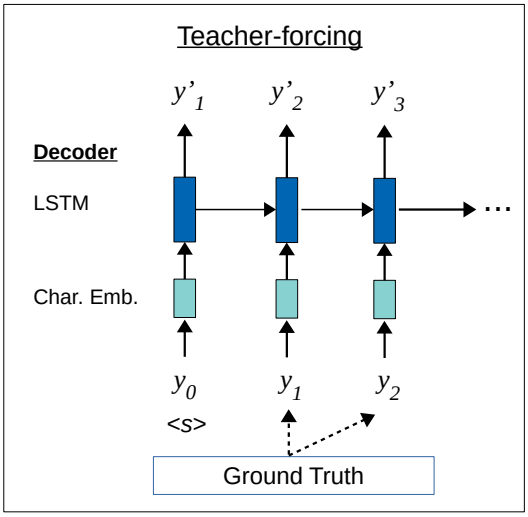


Figure 2.3. Decoding with teacher-forcing strategy for model training.

$$Loss_{ASR}(\mathbf{Y}, P(\mathbf{Y}|\mathbf{X})) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{c=1}^{C}\mathbb{1}(y_t = c) * \log P(y_t|\mathbf{X}, y_{<t})[c], \tag{2.8}$$

2.3. Inference Method

ASR inference aims to predict the most-likely token sequence that corresponds to the input speech as formulated in Equation 2.9.

$$\hat{\mathbf{Y}} = \arg\max_{\mathbf{Y}} \log P(\mathbf{Y}|\mathbf{X}) \tag{2.9}$$

Neural ASR inference starts by feeding the ASR model with the framed speech features sequence. Inside, the encoder encodes the speech features into a hidden representation, and the decoder predicts the output token by taking the previous output token and the speech context information. The decoding method for inference is not the same as the teacher-forcing decoding strategy that does the decoding based on ground truth input. Instead, the decoder takes the output token that is predicted in the prior timestep. An illustration of decoding mechanism in ASR inference is shown in Figure 2.4.

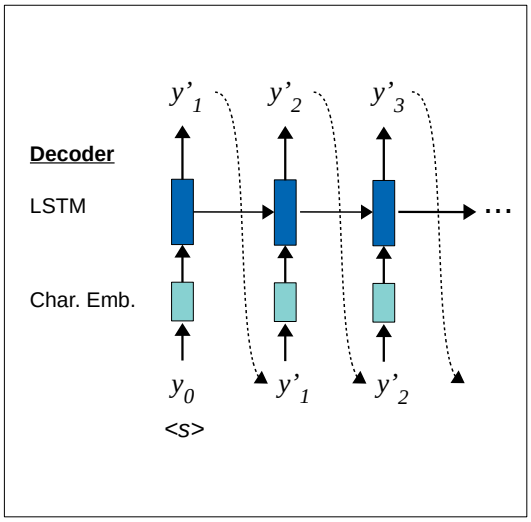


Figure 2.4. Decoding for inference. The decoder input is the token that is predicted in the previous decoding timestep.

There are two kinds of searching algorithm that are commonly implemented in neural ASR system to find the optimum output sequence during the decoding process. Those searching algorithms are greedy search and beam search.

2.3.1 Greedy Search

Decoding with a greedy sequence searching determines the output token y_t at time t by choosing the token with the highest probability based on the predicted probability distribution. Equation 2.10 formulates the output search with a greedy approach for each decoding timestep t .

$$y_t = \arg\max_{1 \leq c \leq C} P(y_t|\mathbf{X}, y_{<t})[c] \tag{2.10}$$

The greedy decoder has the advantages in speed and output stability because the actual output token in a decoding timestep is determined at the corresponding timestep. The disadvantage of this approach is that the quality of the final output sequence may not be the optimal sequence.

2.3.2 Beam Search

Decoding with beam search [42] is a heuristic approach to generate the most-likely output sequence. Unlike the greedy decoder that determines the final output token at every step, a beam search decoder keeps tracks of k token sequences for each timestep and uses those to generate several token sequence hypotheses. It works, first, by keeping k best tokens based on the predicted tokens probability distribution at timestep t=1. For each subsequent decoding step t+1, the algorithm generates all possible token sequences based on the predicted tokens probability distribution at t+1 and k token tracks that it kept before. From that, it keeps top k sequences based on the sequence score and repeats the process in the next timestep. Token sequence score is calculated as the sum of the log probability of the related sequence that it kept so far, as formulated in Equation 2.11. At the end of the decoding process, the final token sequence is chosen based on the sequence with the highest score.

$$\textit{Token\_Sequence\_Score}(y_1, ..., y_t) = \sum_{i=1}^{t} \log P(y_i|\mathbf{X}, y_{<i}) \tag{2.11}$$

Beam search decoder may predict the optimum final token sequence better than the greedy decoder. This is because the beam search decoder keeps several token sequences with a high sequence probability. The consequence of this approach is a long computation time and the instability of the output before reaching the last decoding timestep. This is because the best sequence may change during each timestep.

Chapter 3

Neural Incremental Automatic Speech Recognition

3.1. Related Work

In this work, we are interested to develop neural ISR that has a similar mechanism and performance to the standard neural ASR, but with a shorter delay. One of the approaches that we can apply to develop such ISR is by employing the standard neural ASR architecture to do an incremental recognition. Since the neural ASR performs many-to-may prediction task, we can distill its knowledge to construct an ISR system.

Knowledge distillation is an approach that can train a student model, which is a simplification of a more complex model that acts as a teacher [43, 44]. A student network is commonly constructed as a compression version that is shallower or thinner that then trains the network to mimic the original teacher network by minimizing the loss (typically L2-norm or cross-entropy) between the student and teacher output. Another approach is attention transfer, which was recently proposed by Zagoruyko and Komodakis [45] for image processing. Its basic idea is to ensure the spatial distribution of the student and teacher activations that are similar at selected layers in the network. Each layer in the student network is trained to focus on the same things as in the teacher network. Various tasks have also used attention transfer, such as video recognition [46] and emotion classification [47], but not yet for ISR task.

3.2. Proposed Approach: Neural ISR via Attention Transfer

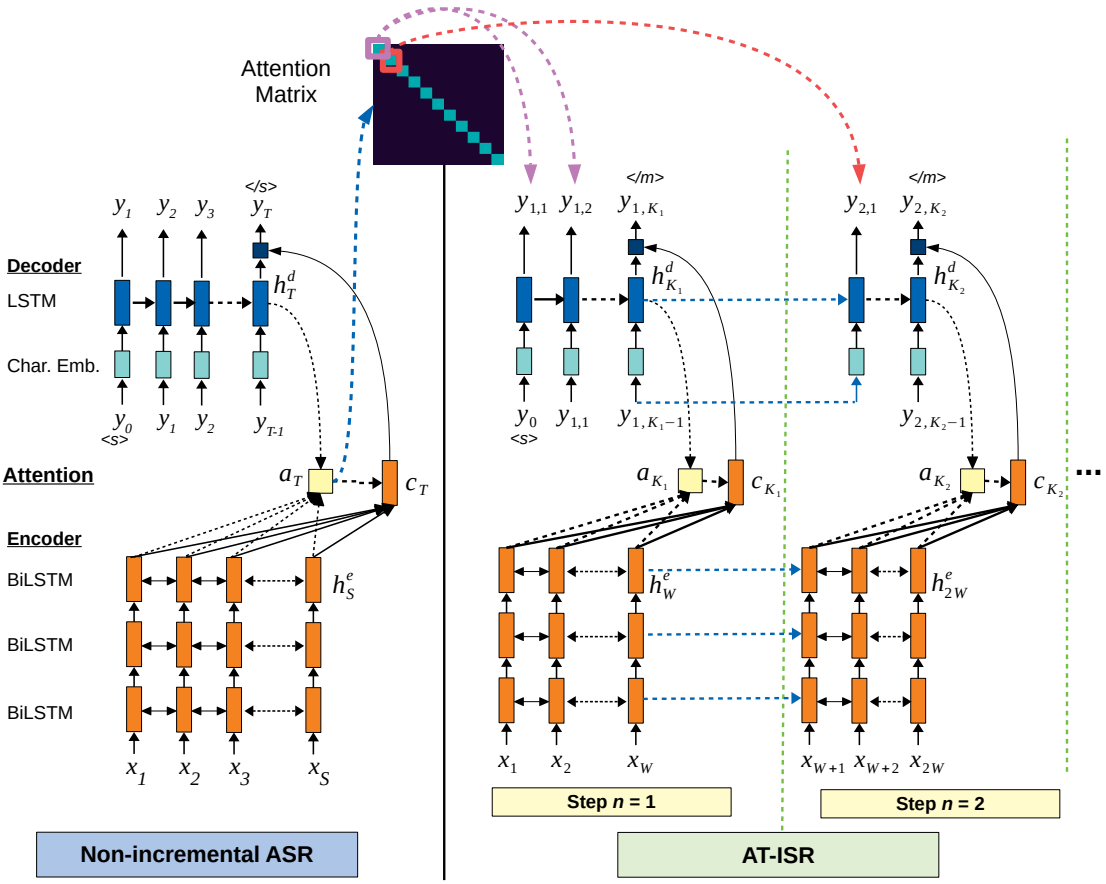


Figure 3.1. Overview of attention-transfer ISR (AT-ISR) training.

As our proposed approach, we apply attention transfer for incremental speech recognition task by treating the standard non-incremental neural ASR as the teacher model and the ISR as the student model. Instead of using a thinner or shallower model, we design an alternative student network that retains the original architecture of the teacher model but with shorter sequences (only a few encoder and decoder states). In this way, no redesign is needed for the ISR, and some hyperparameters can be used without changing them. With attention transfer, the student network learns to mimic the same alignment between the

current input short speech segments and the transcription.

In this thesis, we define the proposed ISR with attention-transfer training as attention-transfer ISR (AT-ISR). The overview of AT-ISR construction method is shown in Figure 3.1. In the next sections, we discuss the AT-ISR in more detail.

3.2.1 Incremental Recognition Method

The proposed AT-ISR in this work performs speech recognition by recognizing the speech segment-by-segment or block-by-block within a fixed window. Fixed window recognition is beneficial for attention-based ISR, in terms of determining the input boundary that decides when the recognition can be started. Here the input boundary is always the last speech frame inside the speech window, so a complex search for it is unnecessary. It does not require a complicated approach, such as voice activity detection and word boundary detection, to start the incremental process. The output boundary is predicted along with the output production.

ISR predicts a sentence text $\mathbf{Y}$ with a length of T tokens from a full speech utterance $\mathbf{X}$ with a length of S frames in N recognition steps by processing a speech segment in each step. Each speech segment has an identical size, which we define it as window size W.

Figure 3.2 illustrates an incremental recognition with neural ISR. The recognition procedure for each recognition step $n = [1, ..., N]$, where $N = \frac{S}{W}$, is below.

1. Encode $\mathbf{X}_n = [x_{(n-1)w+1}, ..., x_{(n-1)w+w}]$, a segment of W speech frames from $\mathbf{X}$, where W<S.

2. Decode and predict $\mathbf{Y}_n = [y_{n,1}, ..., y_{n,k_n}]$, a segment of K_n text tokens from $\mathbf{Y}$, where $0 \leq K_n < T$, until an *end-of-segment* (defined as </m> symbol) token is predicted by attending the encoder states from $\mathbf{X}_n$. Token $y_{n,1}$ is a token next to the last output token in the previous step before </m> is predicted.

3. Shift the input window W frames and keep the model states.

In real-time inference, speech recognition with neural ISR can be started after the speech reaches W frames long. Therefore if the actual speech length is long,

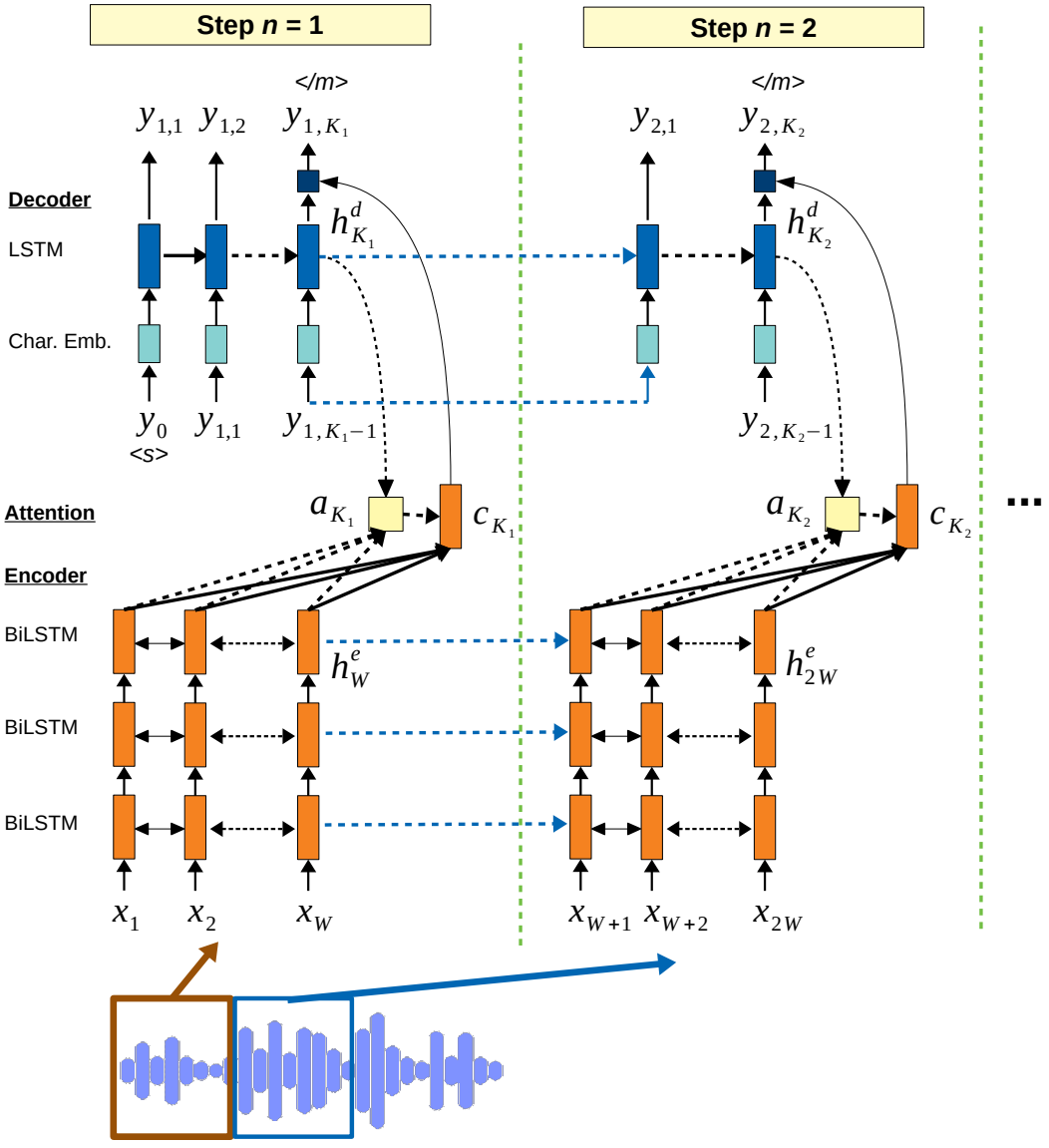


Figure 3.2. Segment-based recognition with sequence-to-sequence neural network framework.

the waiting time to get the output will be not as long as the standard neural ASR. The incremental recognition will be stopped when there is no speech segment left to be recognized. In this work, to avoid an additional delay and to keep the output stability, the proposed neural ISR performs decoding based on a greedy search.

The basic neural ISR does the recognition by predicting the text tokens that align with the whole speech frames inside the window. An alteration in the window can be applied to enrich the information in the input. It is done by enclosing the main speech segment with contextual speech frames [31]. There are two types of contextual speech frames:

- Look-back frames: Speech frames before the main speech segment.
- Look-ahead frames: Speech frames after the main speech segment.

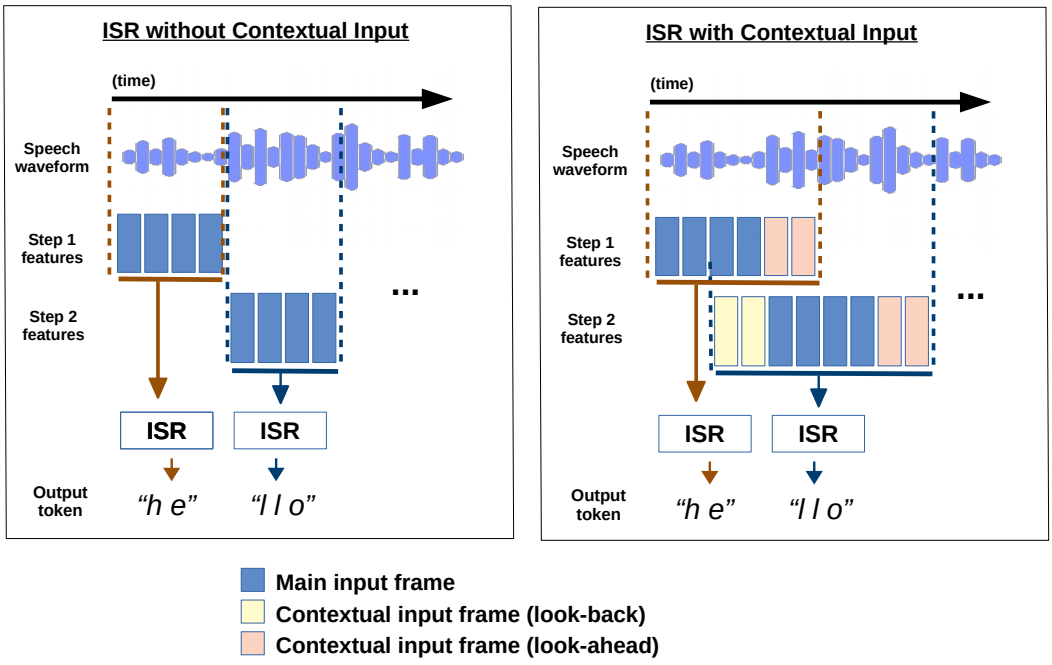


Figure 3.3. Examples of incremental speech recognition with and without contextual input segments.

A visual comparison between the incremental recognition with and without contextual inputs can be seen in Figure 3.3. Similar to the main speech segment,

the size of the contextual input segment in each incremental recognition step is consistent. When contextual speech frames are included, for each recognition step n, ISR encodes the main frames and the contextual frames together. Instead of predicting all output tokens that align with the entire segment, ISR only predicts the tokens that align with the main speech frames. The recognition delay is only affected by the window size of the main input and the number of the look-ahead frames. Look-back frames do not introduce new delays because they are the input from the previous recognition step.

3.2.2 Attention Transfer

Attention transfer is a procedure that allows a student model to learn attention-based information from a teacher model. In this work, the proposed neural ISR is trained by performing attention transfer from a standard non-incremental neural ASR. Specifically, the AT-ISR learns the attention-based alignment between speech segment and text sequence segment that is computed by the standard ASR's attention component. Attention components produce a matrix of alignment scores between encoder hidden states and decoder hidden states. This matrix is also known as the attention matrix. The alignments for AT-ISR training is extracted from attention matrix. An example of attention matrix can be seen in Figure 3.4. The attention transfer here aims to create ISR that mimics the alignments from standard ASR to do an incremental recognition.

3.2.3 Training Method

AT-ISR training consists of two phases as the following.

1. **Attention-based alignment generation.**

   The $\mathbf{X}_n$ and $\mathbf{Y}_n$ pairs are decided based on the alignment by the attention component of the non-incremental ASR with a teacher-forcing text generation. The alignment inference does not involve another system, and a one-time alignment generation is sufficient. The alignments that generated here are hard alignments, in which a text token is only aligned to a speech frame. In attention alignment, a text token may have a high alignment

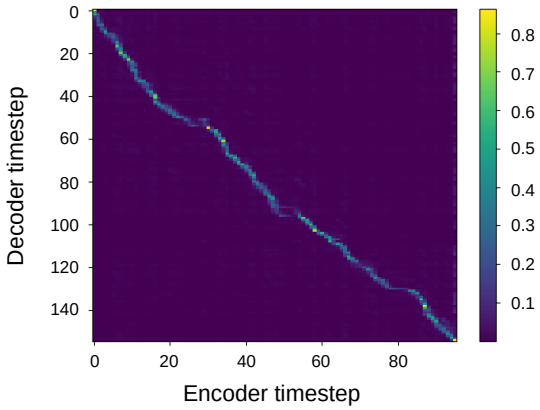


Figure 3.4. Example of attention matrix that generated by standard ASR.

score to several speech frames. From that, we only align the text token to a speech frame with the highest attention score. To be precise, a target token y_t at time t is aligned to s-th input speech frame x_s, which corresponds to encoder state h_s^e. Speech frame index s where y_t aligns to (l_t), follows the condition in Equation 3.1. In the final alignment, a speech frame without a text token that aligns with is possible.

$$l_t = \underset{l_{t-1} \leq s \leq l_{t+1}}{\arg\max} \; Score(h_s^e, h_t^d) \tag{3.1}$$

If the ASR encoder applies hierarchical sub-sampling, an encoder state will be representing the information from several speech frames. In this case, a text token will be aligned to a speech segment with a size that equals to the sub-sampling rate in ASR encoder. An example of attention alignment generation from ASR, which applies hierarchical sub-sampling with a rate of W, is shown in Figure 3.5. In Figure 3.5, a text token is only aligned to one of the speech segments that consists of W speech frames.

2. **AT-ISR model training with attention-based alignment.** The AT-ISR model training is illustrated in Figure 3.1. To enable short-segment-based prediction, AT-ISR is trained using $\mathbf{Y}_n$ that is followed by an $</m>$ token as the target of $\mathbf{X}_n$. The $\mathbf{X}_n$-$\mathbf{Y}_n$ pair is obtained from the attention-

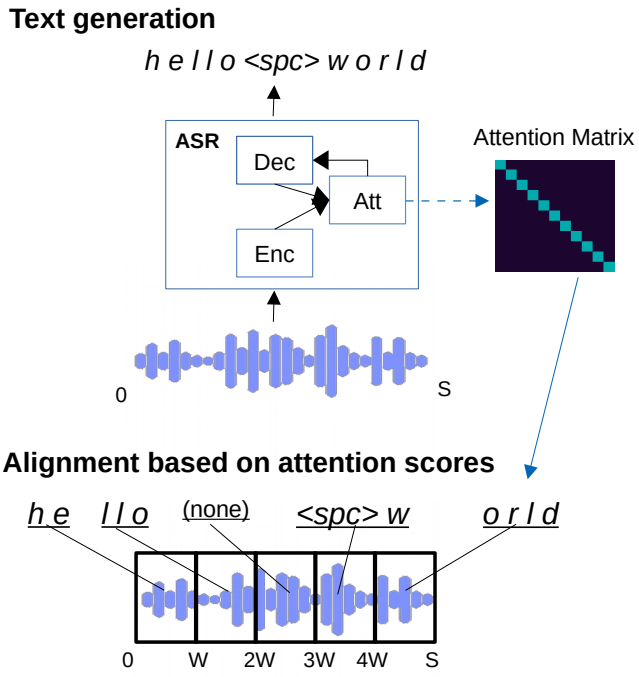


Figure 3.5. Example of attention-based alignment generation with neural ASR.

based alignment generation phase. If the contextual input is applied, the model training is done by using $\mathbf{Y}_n$ and $</m>$ symbol as the target of $\mathbf{X}_n$ that is concatenated with the contextual input.

3.2.4 Delay Management

In inference, for optimum performance, AT-ISR does the incremental recognition of speech segments with the same input window size configuration that is used during the model training. AT-ISR delay can be controlled by concatenating consecutive speech segments in alignment and adjusting the text sequence segment according to it, and then learn the concatenated alignment during model training. The shortest or basic delay is equal to the number of speech frames that an encoder state represents in the attention matrix. This is because the attention component does the scoring to encoder hidden state that represents the speech frame, not directly to the actual speech frame. An encoder hidden state may represent several speech frames, depending on the sub-sampling rate.

3.3. Experimental Setup

3.3.1 Dataset

We used *LJ Speech* [5] to find the best AT-ISR mechanism and *Wall Street Journal* (*WSJ*) corpus [6] for the well-performed configurations. *LJ Speech* dataset consists of 13.100 English speech utterances from a single speaker (24 hours). We divided the *LJ Speech* data into 12.314, 393, and 393 utterances as the train, development, and test sets consecutively. The details of *LJ Speech* are described in Appendix A.1. *WSJ* corpus consists of speech utterances that were spoken by multiple speakers. We followed the original dataset configuration to divide this data into the train (*SI-284*), development (*dev93*), and test (*eval92*) sets. Further details of the *WSJ* dataset can be seen in Appendix A.2.

All utterances in both datasets have a 16-kHz sampling rate. From each speech utterance, we extracted 80-dimension log Mel spectrogram feature, where the window length of each feature frame was 50 msec and the window shift was 12.5 msec.

3.3.2 Model Configuration

The configuration of the proposed, topline, and baseline models in the experiments are described below. All models in this experiment did not utilize an external language model to produce the output sequence.

- **Proposed Model**
  The proposed AT-ISR model consisted of encoder and decoder components with an attention mechanism. In the experiments, we represented the model output as character units, which is the basic and common output unit in end-to-end ASR systems. The encoder part consisted of a layer of feed-forward neural network and three layers of BiLSTM with a sub-sampling rate of two in each BiLSTM layer, resulted in total sub-sampling rate of eight. The first layer in encoder took a sequence of framed speech with 80 features and output 512 features, while each BiLSTM layer output 256 features. The decoder side consisted of an embedding layer, an LSTM layer with an attention mechanism, and a softmax layer. In this work,

we used an attention mechanism with MLP scoring function that utilized previously-proposed multi-scale alignment and contextual history information for scoring [48].

- **Topline Model**
  The topline was standard non-incremental speech recognition using a standard neural ASR model, which was the teacher of the AT-ISR model. The teacher ASR and AT-ISR in this experiment had an identical structure.

- **Baseline Model**
  The first baseline was short-segment-based recognition using the teacher ASR model. The standard neural ASR, which was the teacher of AT-ISR, was trained to recognize a completed speech utterance non-incrementally. In the baseline experiment, this model performed incremental recognition during the inference by feeding it a short speech segment. We experimented on two approaches to segmenting the speech. The first approach was word-level segmentation based on forced-alignment technique using HMM-based ASR [49], and another approach was fix-sized segment recognition. The baseline ISR with the first approach could be considered as isolated-word recognition. Another baseline was neural ISR that had an identical structure and the same recognition mechanism as AT-ISR, but was trained using alignments from HMM-GMM ASR (no attention transfer).

  We also compared AT-ISR to the existing neural ISR frameworks: NT and unidirectional RNN-CTC ISR. The NT model was trained using alignments extracted using HMM-GMM ASR. It consisted of the unidirectional encoder with the same size as the proposed model's encoder and the transducer with the same size as the proposed model's decoder. The RNN-CTC ISR was the one that reported in a study by Hwang and Sung (2016) [3], which applied a model of two unidirectional LSTM layers. It was designed to predict the speech transcription as a character sequence by performing beam searching with a beam width of 512.

3.3.3 Incremental Unit

Incremental unit is the speech segment unit which is recognized by ISR in an incremental step. The ASR encoder in our configuration applied three BiLSTM layers with a sub-sampling rate of two for each layer. As a result, the total sub-sampling rate was eight, so one encoder state represented the information from eight consecutive speech frames. Thus, our basic incremental unit was eight speech frames that approximately equal to 0.14 sec speech. For the rest of the part of this section, we refer the eight speech frames as one speech block.

3.3.4 Evaluation Metric

The speech recognition system's performance was evaluated based on the output quality and delay.

- **Speech Recognition Output Quality**
  The quality of speech recognition output was measured using character error rate (CER) of the predicted token sequence. CER is the number of minimum edits in the hypothesis that is required to make the hypothesis exactly matches the reference. CER equals to character-level edit distance that follows Equation 3.2.

$$Edit\_Distance = \frac{S + D + I}{N_{ref}} \times 100\% \tag{3.2}$$

  In CER calculation, S, D, and I denote the numbers of character substitutions, deletions, and insertions respectively that are required to correct the hypothesis, and N_{ref} denotes the number of characters in the reference text. The CER of the incremental model was measured by comparing the model's complete output sequence against the full reference transcription.

- **Speech Recognition Delay**
  In this thesis, speech recognition delay refers to the time that a speech recognition system needs to output the first and stable speech transcription token. An output sequence is considered as stable if the outputs from the

beginning until the current decoding step do not change, even after the prediction in the new steps.

The speech recognition delay was measured based on two factors: input-wise delay and computational delay. Input-wise delay is the delay that is caused by the waiting time for input speech. It equals to the duration of speech input that is fed into the encoder. Computational delay is the delay by running the model to do inference. It is a sum of duration of the feature extraction and encoding-decoding processes. For ISR system, its delay is the total of input-wise and computation delays for an incremental step. An illustration of ISR delay is shown in Figure 3.6.

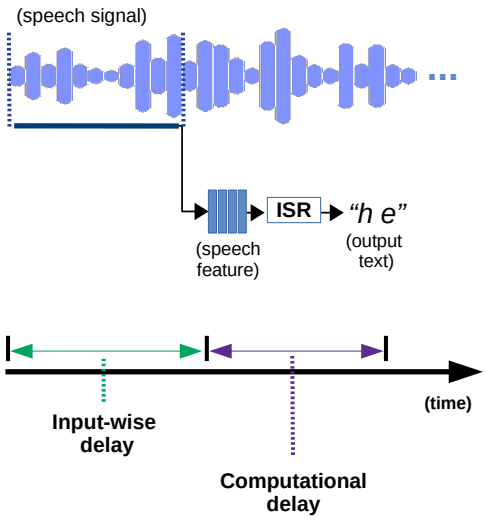


Figure 3.6. Incremental speech recognition delay.

Computational delay is highly affected by the computing resource. In this thesis, the computational delay was measured by running the speech recognition system in the following environment:

- Processor : Intel ®Core™ i7-9700K CPU @ 3.60GHz
- GPU : NVIDIA GeForce RTX 2080Ti

3.4. Results

3.4.1 Speech Recognition Performance

Table 3.1. CER (%) of topline ASR, baseline ISR, and proposed AT-ISR based on *LJ Speech* corpus. All ISRs performed incremental recognition with the basic incremental unit (1 speech block) without contextual inputs.

System		**Delay (sec.)**		**CER**	
		Input	**Computation**	**Dev.**	**Eval.**
Topline ASR					
Teacher ASR (non-incremental)		6.57 (avg.)	0.30	2.00	1.94
Baseline ISR					
Teacher ASR + isolated word recog.		0.36 (avg.)	0.02	27.07	26.91
Teacher ASR + fix-sized segment recog.		0.14	< 0.01	79.63	80.34
Proposed: AT-ISR		**Delay (sec.)**		**CER**	
Model state/step	**Dec. Initial Input**	**Input**	**Computation**	**Dev.**	**Eval.**
Reset	<m>	0.14	< 0.01	30.68	31.16
Reset	prev. actual output	0.14	< 0.01	25.17	25.58
Keep	<m>	0.14	< 0.01	23.07	23.34
Keep	prev. actual output	0.14	< 0.01	**21.73**	**21.70**
Other existing ISR					
NT		0.14	< 0.01	24.73	24.00

First, we compared the performances of the topline ASR, baseline ISR, and proposed AT-ISR with the basic incremental unit. The models performances based on *LJ Speech* dataset is shown in Table 3.1. The ISR models here did not use contextual input as their input. In this experiment, we investigated the performance of AT-ISR that kept and did not keep the model's recurrent states during incremental recognition. We also explored the two types of initial input token for ISR decoder. In the first type, ISR took a *beginning-of-segment* symbol $<m>$ as the first decoder input token for each incremental step. The second type was the decoding by taking the last character output from the previous step (before $</m>$ predicted) as the initial input token.

The result in Table 3.1 shows that the best AT-ISR outperformed the baselines. Incremental recognition using a standard non-incremental ASR model did not perform well, thus, ASR model should be adapted to do a short-speech recognition in order to perform incremental recognition. Among the proposed mod-

Table 3.2. AT-ISR CER (%) on *LJ Speech* dataset based on contextual input segments size (1 block = 8 frames ≈ 0.14 sec).

Input Segment Size (blocks)			**Delay (sec)**		**CER (%)**	
Look-back	**Main**	**Look-ahead**	**Input**	**Computation**	**Dev.**	**Eval.**
Topline ASR			6.57 (avg)	0.30	2.00	1.94
Baseline: Neural ISR (teacher architecture) trained with HMM-GMM alignment						
0	1	0	0.14	< 0.01	21.81	21.80
0	1	1	0.24	0.02	12.16	11.92
1	1	1	0.24	0.02	7.54	7.00
Proposed: AT-ISR						
0	1	0	0.14	< 0.01	21.73	21.70
0	1	1	0.24	0.02	7.27	7.99
0	1	2	0.34	0.03	4.52	4.21
0	1	4	0.54	0.05	**3.13**	**3.24**
1	1	1	0.24	0.02	5.98	5.75
2	1	1	0.24	0.02	5.83	5.55
4	1	1	0.24	0.02	5.14	4.90
Other existing ISR: NT						
0	1	0	0.14	< 0.01	24.73	24.00
0	1	1	0.24	0.02	13.95	12.65
1	1	1	0.24	0.02	10.91	10.32

els, keeping the model states and transferring the attention knowledge greatly improved the performance. The lowest CER was achieved by feeding the last character from the previous step for the first decoder input. If we compare it to the topline ASR, ISR resulted a lower performance. This was caused by the nature of short speech segment recognition, in which a short speech segment might not provide the sufficient information for the correct transcription. Based on this phenomenon, we investigated the effect of contextual input to the incremental recognition performance.

Table 3.2 shows our exploration result on AT-ISR that recognized the speech incrementally by including contextual input in the input window. AT-ISR models here kept the model states for each incremental recognition step and started the decoding by taking the last actual output from the previous step. In this table, we also compared the AT-ISR to the baseline neural ISR with the same architecture and recognition mechanism but did not trained with attention transfer.

The addition of contextual input in the input segment improved the ISR performance. When the contextual input was not used (0 look-back and 0 look-ahead block), all ISR frameworks in Table 3.2 have a similar performance. When contextual input was included in the input window, AT-ISR performance was significantly better than the other ISR approaches.

Given the same architecture and the same encoding-decoding mechanism, the ISR model that learned from attention-based alignment resulted in a better transcription than the baseline model that learned from HMM-GMM alignment. Without the contextual frames, the performance of both models was similar. However, when the input of an incremental step was a main block with a look-ahead block, AT-ISR significantly outperformed the neural ISR with HMM-GMM alignment learning. Our experiment results show that attention-based ISR is more suited to be trained using alignment that is generated by attention-based ASR. Attention-based alignment tells a summary of input frames location that the model needs to pay the highest attention, among a range of frames with high alignment score, to produce a token. It is depends on the model's capability and quality. The alignment from HMM-GMM ASR might not pair some tokens with the speech frames that a neural ASR primarily needs attend to. It might pair the tokens with speech frames that scored second or third highest attention alignment score. For this reason, when the recognition only considered the main input, the performance of both approaches did not differ significantly. But when the neural ISR based on HMM-GMM alignment allowed to take contextual input, the information in the contextual input might not as match as the AT-ISR, so it cannot perform as well as AT-ISR. AT-ISR that recognized a 0.24 sec speech, which consisted of 1 main and 1 look-ahead blocks, in each incremental step achieved a test set CER of 7.99%, which was 4.66% lower than NT with the similar input delay. AT-ISR approach might be more suitable for attention-based ISR compared to the other approaches in this experiment.

The results in Table 3.2 reveal that looking ahead from the main input segment resulted in a better performance than looking back, perhaps because, in each recognition step, the model already maintains information from the previous steps. Therefore, adding previous frames to the main segment is might not critically necessary. On the other hand, the look-ahead segment provides new in-

Table 3.3. Topline ASR and AT-ISR CER (%) on *WSJ eval92* set. (1 block = 8 speech frames ≈ 0.14 sec)

Model		**Delay (sec)**		**CER (%)**
		Input	**Computational**	
Non-incremental (Topline)				
CTC [33]		-	-	8.97
Att Enc-Dec Content [33]		-	-	11.08
Att Enc-Dec Location [33]		-	-	8.17
Joint CTC+Att (MTL) [33]		-	-	7.36
Att Enc-Dec (ours, teacher)		7.88 (avg.)	0.32	6.26
Neural ISR				
Input segment size (blocks)		**Delay (sec)**		**CER (%)**
Look-back	**Look-ahead**	**Input**	**Computational**	
Baseline: Neural ISR (teacher architecture) trained with HMM-GMM alignment				
0	1	0.24	0.02	20.15
0	4	0.54	0.05	11.95
Proposed: AT-ISR				
0	1	0.24	0.02	18.37
0	4	0.54	0.05	**7.52**
Other existing ISR				
RNN-CTC beam search ISR [3]		-	-	10.96

formation that supports a better understanding of the main segment, although it introduces a new delay. By only looking four blocks ahead, AT-ISR achieved performance with a small difference from the non-incremental ASR with significant delay reduction.

From our experiments on *LJ Speech* data, we learned that the optimum performance with reasonable latency was achieved by the following: (1) included a few ahead blocks, (2) set the last actual character of the previous step as the decoder initial input, (3) kept the recurrent states across the steps, and (4) utilized the distilled knowledge of the attention matrix in the training. With this configuration, we constructed ISR model using *WSJ* dataset.

Table 3.3 shows the experiment results based on *WSJ* data. The length of a completed utterance in this experiment was 7.88 sec on average. To do non-incremental speech recognition, our non-incremental model requires a time of

7.88 sec to wait for the input and 0.32 sec for output computation to predict the output, totaling in 8.20 sec of delay in average. On the other hand, AT-ISR only took a time of 0.54 sec to wait for the input segment and 0.05 sec for computation, and the recognition performance was close to the standard ASR performance. Similar to the previous experiment, AT-ISR performed incremental recognition better than the baseline neural ISR. We also compared our results with several published models of non-incremental ASR such as CTC, Attention Encoder-Decoder, and Joint CTC-Attention model. Our results demonstrate that AT-ISR could achieve comparable performance with other published models.

3.4.2 Impact of Delay to Speech Recognition Performance

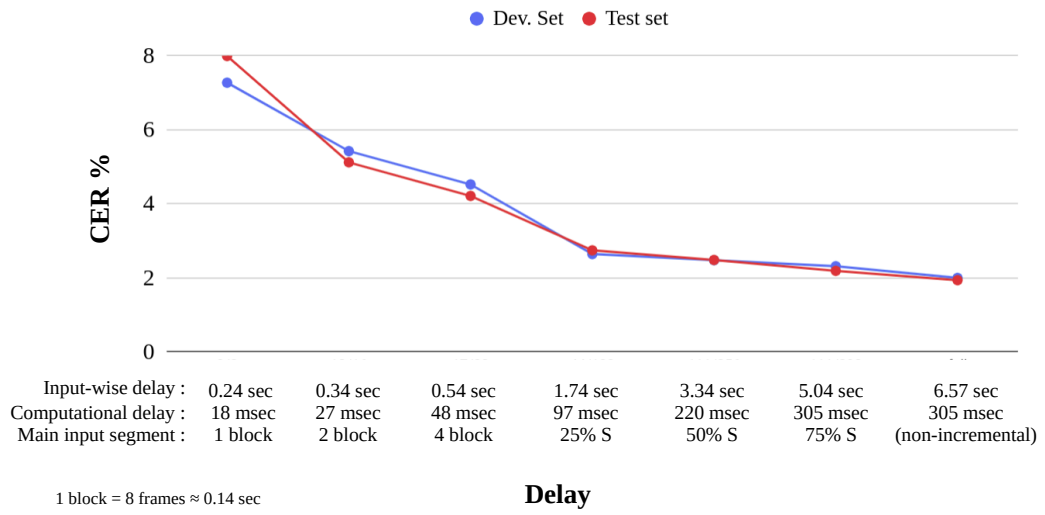


Figure 3.7. AT-ISR performance on *LJ Speech* dataset with various main input segment size (S = average frame length in *LJ Speech* set (6.57 sec))

We investigated the impact of the main input segment size. Figure 3.7 illustrates the performances of the AT-ISR models with different main input segment size. In this figure, each model were allowed to take a look-ahead speech block in an incremental step. The result shows a trade-off between time and performance. In real-time speech translation tasks, we prefer the speech recognition delay to be as short as possible with a performance that is close to the non-incremental

ASR. Therefore, we need to find a delay configuration that keeps the balance between recognition output delivery speed and recognition performance. Here in Figure 3.7, although significant improvement happened until the main input segment with a length of 25% of full utterance (1.74 sec), it did not occur again on the subsequent sizes. With this amount of delay, AT-ISR achieved a comparable performance to the one that needs to wait until the speech to end. This point of delay can be considered as balance point between time-performance. It also indicates that the AT-ISR model's performance is able to approach that of the teacher model.

3.4.3 ISR Error Analysis

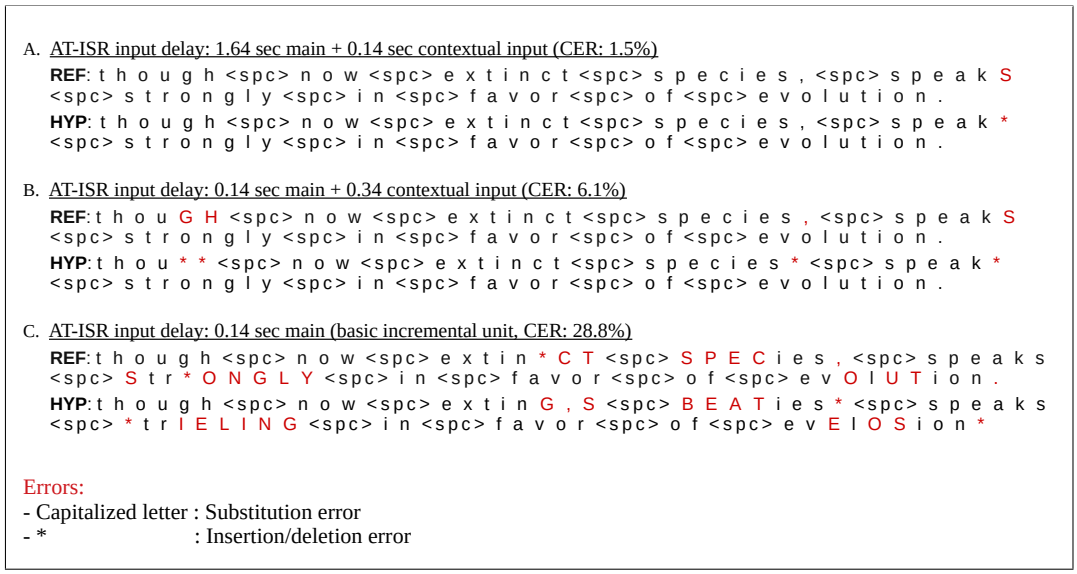


Figure 3.8. AT-ISR output examples.

From the experiment results, we can see that ISR performance increase as higher as the input delay gets. Short speech segment recognition is difficult than full speech recognition due to the information limitation in the input. Figure 3.8 shows the examples of how the input information limitation affected the AT-ISR's transcription.

AT-ISR with the shortest delay (0.14 sec) that does not consider contextual input had the highest transcription error. However, for some incorrect word, the

prononciation was relatively close to the reference word (e.g. "*extinct*" predicted as "*exting*"). The model did not predict a character sequence of a word with a completely different pronunciation as the reference, so we can see that the model could extract the appropriate phoneme-level information from the shortest input. Despite it, it could not predict the correct character sequence. This is because, in English language, multiple words or sequences of words can consist of the same or similar phonemes. In conventional HMM-GMM ASR, the assignment of phonemes into word depend on the neighboring phonemes of the target phonemes. In case of AT-ISR, when the AT-ISR input was very short and it cannot see the contextual speech frame, the phonemes information that it can obtain might not sufficient to form the character sequence of the correct word. Extending the input segment window allowed the AT-ISR to get longer phoneme sequence information, so it had a higher chance to predict the correct characters.

For some incorrect word, the word might sounded different to the reference word (e.g. "*species*" predicted as "*beaties*"). The difference in the sound of those words could occur when the speech frames of a phoneme are split and assigned into two different input segment. Therefore, the model might not extract the correct phoneme-level information in one incremental step.

Based on the output analysis, we can conclude that ISR error can occur when:

1. The length of phoneme-level information is not long enough.
2. Speech frames of a phoneme are split and assigned into different input segment, and those are not within the same incremental step.

By extending the length of the input window or by incorporating contextual input frames, we can minimize the chances of getting the conditions above and increase the ISR performance, as shown in Figure 3.8. However, although might resulted in a good performance, ISR with a high delay might not preferable for simultaneous S2ST tasks, so the choice of delay should be made carefully according to the system's goal.

3.5. Summary

In this section, we described the proposed AT-ISR framework for incremental speech recognition with attention-based sequence-to-sequence neural framework. AT-ISR learns the attention-based information from the standard neural ASR, which can be considered as a teacher model, to do incremental processing during inference. The main difference between AT-ISR and the standard neural ASR is that the AT-ISR recognizes shorter sequences than the standard architecture. Since AT-ISR applies the same structure as the teacher, no new redesign is needed for the ISR, and some hyperparameters can be used without any changes. Among the ISR approaches that we explored, the optimum performance was achieved by including the look-ahead segment in the input window, setting the last character of the last step as the decoder's first input, keeping the recurrent states across the steps, and applying attention transfer. In our experiment result, by recognizing a segment of speech with a length 0.54 sec incrementally, AT-ISR achieved a close performance to non-incremental ASR that cost a delay of 6.57 sec.

Chapter 4

Neural Incremental Automatic Speech Recognition in Speech Translation Task

4.1. Related Work

Utilization of ASR in translation task through ASR-MT integration is a challenging problem due to the error propagation and the incompatibility of training materials between both modules. Several studies addressed this challenge in speech translation tasks by adapting the ASR output to MT. One study [50, 51] modified the ASR output to resemble MT training data and resulted in the improvement in translation. The ASR output modifications included the handling of lettercase and punctuation, disfluency removal, normalization, and compound word recomposition. Another work [52] utilized a lattice-based ASR-MT interface to improve translation quality. These works, however, are based on a conventional S2ST framework that does not do the processing in a real-time manner. Wang et al. [53] previously constructed a real-time system prototype by unifying an HMM-based ASR system and an online MT system [54]. Unfortunately, study of the integration of end-to-end neural ISR and neural MT for speech translation remains limited.

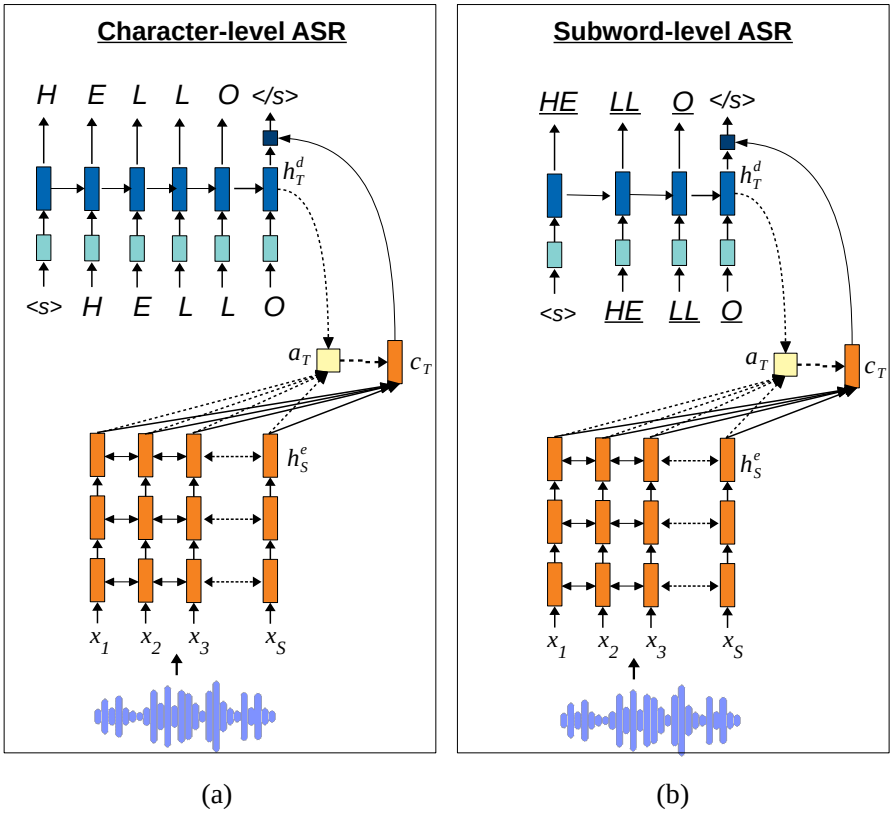
4.2. Parity of ASR and MT Tokenization in Speech Translation

To do a proper speech translation, ASR must provide tokens that can be recognized by the MT system. In common practice, when a token in the source language text is not in the MT input vocabulary, the token is assigned as an '*unknown*' symbol before it is processed by the MT, however, it can damage the translation performance [55]. To avoid this condition, beside the language uniformity, the ASR token unit should be in the same granularity as the MT input unit. For example, the character-level output from ASR might confuse a word-level MT system because the MT only learned to do translation from word-level tokens. In such a case, the former text has to be converted into tokens with the same granularity as MT input side. In a word, the uniformity of vocabulary and token unit of the ASR output side and MT input side is important to do a proper translation.

4.2.1 ASR Output Unit

The basic end-to-end ASR represents the output as character [1, 24, 56]. Recent studies also create end-to-end ASR that outputs subword units [23, 57]. Word-level end-to-end model is rarely developed because it may not able to cope with out-of-vocabulary words and result in spacious model. The descriptions of character-level ASR and subword-level ASR are the following.

- **Character**
  Figure 4.1(a) illustrates an end-to-end character-level ASR. In each decoding step, the ASR model outputs a character token. During training, the target text consists of a sequence of characters, where the character sequences of each two words are separated by a whitespace token. In this work, we symbolize the whitespace as <*spc*> token. An example of character-level tokenization of a sentence is the following:



(a) (b)

Figure 4.1. End-to-end character-level ASR (a) and subword-level ASR (b).

Word: *hello nice to meet you*
Character: *h e l l o <spc> n i c e <spc> t o <spc> m e e t <spc> y o u*

Character-level representation has some advantages in enhancing the ASR generality, so it can avoid model overfit and also out-of-vocabulary condition [56]. The number of character vocabulary is not as many as word vocabulary, so it also save the model space. However, character-level unit may fail to keep the contextual information of the word. Because of it, character-level errors in ASR output may result in words that does not have any linguistical meaning.

- **Subwords**
  Figure 4.1(b) shows speech recognition process with end-to-end subword-level ASR. The ASR predict a subword token for each decoding step. Subword unit is a text token representation that has a finer granularity than words but coarser granularity than characters. An example of subword-level tokenization from a sentence is the following:

  Word: *hello nice to meet you*
  Subword: *he ll o <spc> ni ce <spc> to <spc> me et <spc> you*

  Similar to the character unit, the subword unit enables the ASR to avoid out-of-vocabulary condition. Subwords have a coarser granularity than characters, so it can preserve the word context information better than characters. The context information preservation is better when the subword token consists of a longer character sequence, which may resemblant to a word. However, a subword vocabulary can be large depending on the amount of word context that we want to keep. Subword-level model may require a larger space than the character-level model.

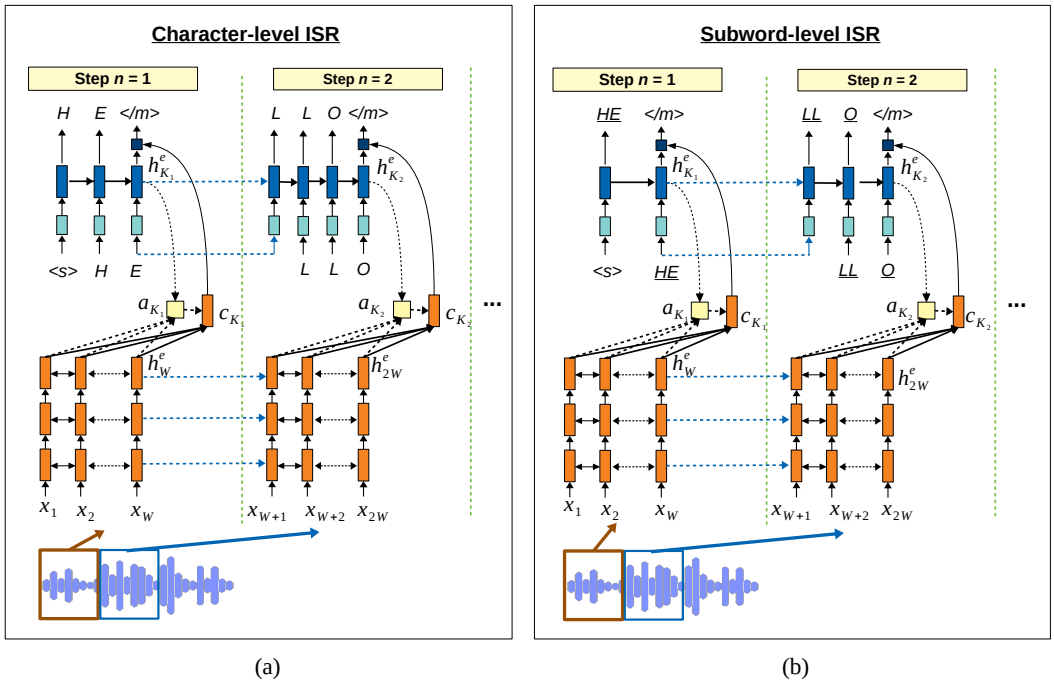
4.2.2 MT Input Unit

End-to-end MT systems generally adopt subwords as the input and output representation unit [58, 59, 60, 50]. Subword representation is utilized to avoid the out-of-vocabulary condition, which often happens in the word-level model, and to preserve the word context information.

Subword vocabulary construction and tokenization for MT are generally done by using bype-pair-encoding (BPE) segmentation algorithm [61]. In BPE mechanism, a word-to-subword segmentation model is trained using text sentences that consist of word tokens. The algorithm begins the training by initializing the subword vocabulary with character vocabulary and representing each word from training data as character sequence. In the subsequent processes, the algorithm iteratively replaces the most frequent token pair in the training data with a new token, which is a merged form of the target pair, and then add the new token to the subword vocabulary. The model construction here is done using only text

sentences without depending on language and phonemes. In inference, given a word, the segmentation model converts the word into subwords by representing it as a character sequence first and then applying the merge operation that it has learned.

4.3. Proposed approach: Subword-level ISR



(a) (b)

Figure 4.2. End-to-end character level ISR (a) and subword-level ISR (b).

In this thesis, we propose subword-level ISR system to enable the utilization of it in speech translation with standard subword-level MT towards simultaneous S2ST. A visual comparison of end-to-end character-level and subword-level ISR can be seen in Figure 4.2. MT systems are commonly developed to take a subword sequence as the input, therefore, MT should be able to do a proper translation from subword-level ISR output. In this work, we constructed the subword vocabulary for ISR and MT by using a word-to-subword segmentation model with BPE algorithm that is implemented in the SentencePiece tokenizer [62].

We propose three approaches of subword-level ISR. Two approaches perform conversion of character-level ISR output into subwords and one approach performs end-to-end subword-level ISR. The approaches that apply character-to-subword conversion are beneficial when end-to-end subword-level ISR with a matching vocabulary to the MT is unavailable. Instead of training an end-to-end ISR from scratch to match the MT vocabulary, which can be expensive, the character-to-subword model allows us to connect ISR to subword-level MT with a lesser development cost.

4.3.1 Character-based ISR with Char-to-subword Mapping using SentencePiece Tokenizer

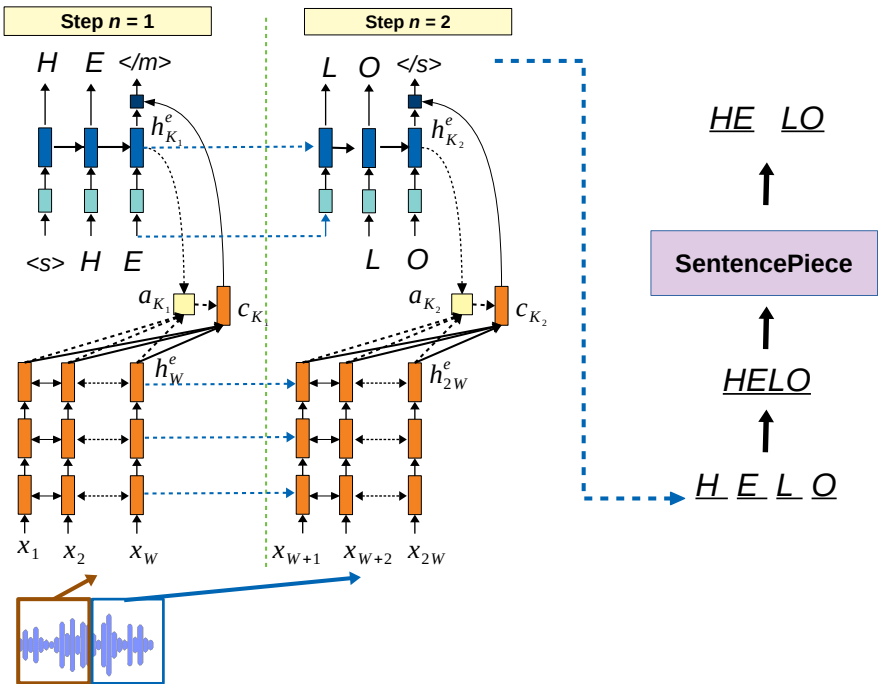


Figure 4.3. Character-based ISR with char-to-subword mapping using SentencePiece tokenizer.

Figure 4.3 shows this approach's scheme. This approach converts output token sequence from a character-level ISR into a subword sequence. When a

character sequence forms a word, this word is segmented into subwords using the word-to-subword segmentation model that is trained using the SentencePiece tokenizer system. This approach only performs characters conversion into subwords without changing the content of the sequence.

4.3.2 Character-based ISR with Char-to-subword Mapping using Encoder-Decoder Framework

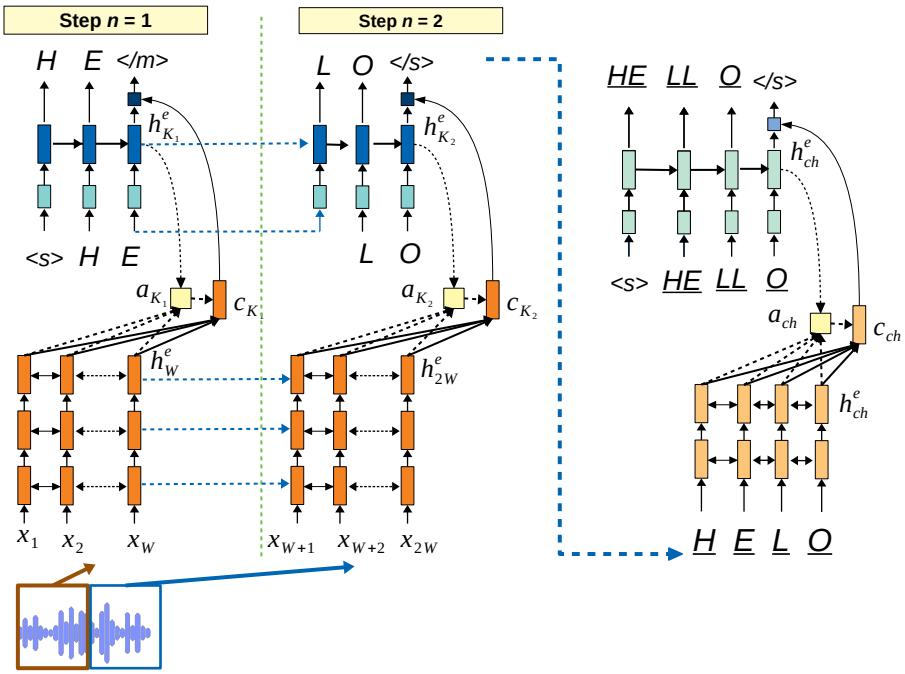


Figure 4.4. Character-based ISR with char-to-subword mapping using encoder-decoder framework.

Similar to the previous approach, we utilize an external module to convert the character sequence from character-level ISR into a subword sequence. The conversion is done using a sequence-to-sequence model that consists of an encoder-decoder with an attention mechanism. The character-to-subword mapping with encoder-decoder not only converts the characters into subwords but also performs content correction. It is achieved by training the conversion model using the ISR-generated character sequence as the input and the correct subword se-

quence as the output target. The subword sequences in the training data are segmented from training word sequences using a word-to-subword segmentation model that is trained with the SentencePiece framework. In our experiment, we applied an incremental version of this model to preserve the short latency by using attention transfer training approach. Similar to the ISR, the incremental characters-to-subword model converts a character sequence into a subword sequence in several steps, where each step takes a fixed number of characters. The scheme of this model that takes four characters in each incremental step can be seen in Figure 4.4.

4.3.3 End-to-end Subword-level ISR

Figure 4.2(b) illustrates the scheme of this approach. This ISR directly models the speech features to subword sequences without any intermediate module. The subword sequences in training data are generated by segmenting the training word sequences using a SentencePiece-trained segmentation model.

4.4. Proposed approach: Integration of ISR and MT

We applied the ISR proposed in Chapter 3 in speech translation task by uniforming the token unit and vocabulary of ISR output and MT input sides. In this work, we limit our focus on how to use ISR for translation task, so we did not explore incremental MT. Here the MT system is fixed to have subword-level input and output to keep our focus on the ASR system. The MT system here is the standard end-to-end MT that predicts a full translation sentence from a full source language sentence. The non-incremental MT here is also used to show us the optimal translation quality of the ISR-generated text.

The visualization of subword-level ISR and subword-level MT integration in speech translation can be seen in Figure 4.5. The procedure starts with incremental speech recognition by ISR. The final ISR output is a sequence of subwords, in which the ISR final output and MT input have a matching subword vocabulary. After obtaining the subword-level output, ISR output is sent to the MT model,

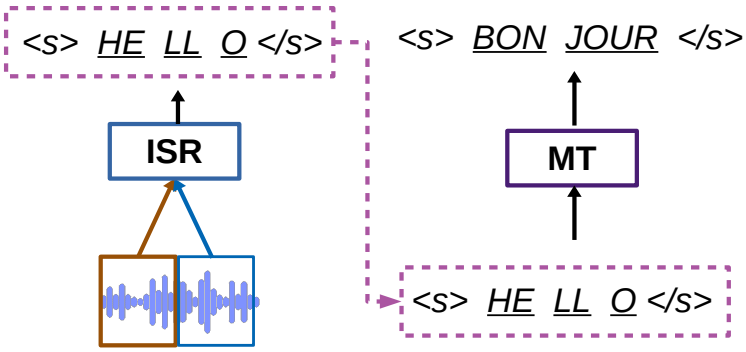


Figure 4.5. Integration of subword-level ISR and MT with subword-level input for English-French translation task.

and the MT model translates the ISR text into the target language.

In the experiment, we aim to see how ISR affects translation performance. The translation tasks that we investigated were English to French, for our main analysis, and English to Japanese. We firstly aim to investigate the ISR for less complicated translation task. Specifically, it is the translation between languages that have the same word syntax order. Translation between languages with different syntaxis word order has been a challenging issue in speech translation fields. In this work, as our main focus is on the ISR, we performed the exploration mainly for the English-French speech translation task, in which both languages have the same syntaxis word order. Based on our best ISR approach, we also examined the translation quality for the English-Japanese task, whose languages have different syntaxis word order.

4.5. Experimental Setup

4.5.1 Dataset

We used corpora related to TED talks to evaluate the ISR in speech recognition and translation tasks. Simultaneous speech recognition or translation systems are most useful for source speech that has a long duration, such as lecture talks. TED talks data were taken from the recordings of lecture talks that were presented in

TED talks. The lectures cover various domains and were spoken by speakers from various speaking styles. TED talks data have been used for speech recognition corpus and translation corpus construction, so we can use it to create and evaluate speech translation system where a parallel dataset of speech, transcription, and translation text is necessary. Since the speech originated from actual talks, the transcription and translation texts were written in spoken language style, which style is slightly different to the written language. From these reasons, we see that the TED talks data are suitable to build our ISR and MT systems with a less-restricted content domain and matching language style.

We trained the ISR model using the *TED-LIUM release 1* corpus [7] that consists of 118 hours of speech data recorded from TED talks. The details of this corpus can be seen in Appendix A.3. The acoustic features for the ISR input consist of 80 dimensions of Mel-spectrogram with a 50 msec window length and 12.5 msec shift.

The MT model was trained with the English-French translation dataset from the *IWSLT 2017* shared task [63], which consists of English transcriptions and French translation texts from TED talks. We used the default in-domain training set to train the model and used the *dev2010* set as the development data. We evaluated the translation quality from the original and ISR-generated texts using the data in the *tst2010* set. The MT model in our experiment applied subword units as the input and output representation. Both input and output vocabularies consisted of 16,000 subwords. All subword vocabularies were constructed using the BPE algorithm in the SentencePiece tokenizer based on the training data of the respective languages. The English subword tokenizer here was also utilized to tokenize the text data for the reference during subword-level ISR training.

For the English-Japanese translation task, the MT model was trained using the combined datasets of *IWSLT 2017* dataset, *ASPEC*[64], and *JESC* [65]. The MT input vocabulary was the same as the English-French MT model, and the output unit was Japanese subword that was a sequence of Japanese characters (mixed logographic and syllabic), in which the tokenization was done based on MeCab tokenizer [66].

To minimize the dissimilarity between the ISR and MT training materials, we removed the punctuation and normalized the numbers in the MT training

corpus. The Unicode symbols in the English texts were also normalized into basic Latin alphabet letters due to the conditions of the *TED-LIUM release 1* corpus, which did not contain punctuation, numbers, or Unicode letters. The *TED-LIUM release 1* transcriptions contain speech fillers, unlike the MT dataset. Therefore, we removed the fillers in the ISR output before passing it into MT.

4.5.2 ISR Model Configuration

We conducted the explorations on character-level and subword-level neural ISR. The character-level and subword-level neural ISR models, as well as non-incremental ASR, were constructed using the same model configuration described in Section 3.3.2, where the decoder embedding layer was customized according to the output unit.

The sequence-to-sequence characters-to-subword model consisted of an encoder-decoder structure. The encoder side consisted of a feed-forward layer (256 units) and a BiLSTM layer (256 units). The decoder consisted of an embedding layer, an LSTM layer (256 units), and a softmax layer. The incremental model performed the recognition on every eight characters from the ISR output as the main input, with the addition of eight look-ahead characters. This configuration was based on the average word length in the training data.

We evaluated the ISR by comparing it to the non-incremental ASR as the topline and isolated-word recognition by the non-incremental ASR model as the baseline. There are three alignment generation approaches that we explored to train the neural ISR model. Those approaches were forced-alignment by HMM-GMM ASR that used phoneme-level acoustic model, forced-alignment by HMM-GMM ASR that used character-level acoustic model, and attention transfer from a standard non-incremental ASR (AT-ISR). Forced-alignment is a method to compute the token-level speech-to-text alignment position, in which it notes the start time and end time of a token. For each HMM-GMM ASR system, forced-alignment produces word-level alignment and token-level alignment of the same unit as its acoustic model's output. From the HMM-GMM ASR with a phoneme-level acoustic model, we cannot infer a precise character- or subword- level alignment. Therefore, the character or subword sequence was aligned to the end timing of the corresponding word [31]. From the HMM-GMM ASR with a character-level

acoustic model, we constructed the character-level and subword-level alignment based on the start time and end time of a character in speech. ISR that learned from attention-transfer or AT-ISR is our proposed neural ISR in this thesis, which was described in Chapter 3.

4.5.3 MT Model Configuration

The MT model in this work was constructed by applying an encoder-decoder structure with an attention mechanism. The MT encoder consisted of an embedding layer (256 units), a feed-forward layer (512 units), and two BiLSTM layers (256 units). The decoder side consisted of an embedding layer (512 units), two LSTM layers (512 units), and a softmax layer.

4.5.4 Incremental Unit

The basic incremental unit of the neural ISR in this experiment was the same as the one described in Section 3.3.3, in which a speech block consists of eight speech frames that approximately equal to 0.14 sec. The neural ISR main input size ranged from a speech block to several blocks that equal to a full speech length. We set the neural ISR to include the two or four look-ahead contextual speech segment as part of its input to keep the performance. The input size here was decided based on the experiment results described in Section 3.4.

4.5.5 Evaluation Metric

The evaluation metrics that we utilized to measure the speech recognition and translation performances are the following:

- **Speech Recognition**
  - **CER**: Character error rate. CER was calculated using Equation 3.2.
  - **WER**: Word error rate. WER is the minimum word-level edits to make the ASR hypothesis same as the reference text. WER calculation was based on Equation 3.2 with the word-level tokens.

- **UCR**: Uncovered-word rate. UCR is the rate of uncovered-words that are produced by the speech recognition system. An uncovered-word is a word that does not exist in the training data because of one or several character-level mistakes in the word. It can be a word that does not have any linguistical meaning. The calculation of UCR followed Equation 4.1, where UC_w is the number of the uncovered-word in the hypothesis and N_{hyp_w} is the number of the word in the hypothesis

$$UCR = \frac{UC_w}{N_{hyp_w}} \times 100\% \tag{4.1}$$

- **Translation**

  - **BLEU**: Bilingual evaluation understudy [67]. BLEU score measures the position-independent n-gram word matches between the hypothesis and the reference. BLEU score was calculated using Equation 4.2, where M_{wn} is the number n-gram word matches between hypothesis and reference, and $N_{hyp_{wn}}$ is the number of n-gram words was hypothesis. In the experiment, we evaluated translation output based on 1-gram BLEU and 4-gram BLEU.

$$BLEU = \frac{M_{wn}}{N_{hyp_{wn}}} \times 100\% \tag{4.2}$$

  - **METEOR**: Metric for evaluation of translation with explicit ordering [68]. METEOR calculates the parameterized harmonic mean of precision and recall of unigram matches. The unigram matching includes the exact, stem, synonym, and paraphrase matches between the words in the hypothesis and reference. METEOR score calculation was done by following Equation 4.3. Here Pen is a penalty to take the unigram matches into longer matches account, C_w is the number of word adjacent chunks between the hypothesis and reference. P_w is the precision and R_w is the recall of unigram word matches. The α_{METEOR} and β are free parameters that are tuned to achieve maximum correlation

with human judgment.

$$METEOR = (1 - Pen) \cdot F_{mean} \tag{4.3}$$

$$Pen = \gamma \cdot (\frac{C_w}{M_{w1}}))^{\beta} \tag{4.4}$$

$$F_{mean} = \frac{P_w \cdot R_w}{\alpha_{METEOR} \cdot P_w + (1 - \alpha_{METEOR}) \cdot R_w} \tag{4.5}$$

4.6. Results

The performance comparison between the end-to-end character-level and subword-level standard ASR and ISR is shown in Table 4.1. The models were evaluated based on *TED-LIUM release 1* set. Here UCR was measured by comparing the hypothesis to the ISR training data word vocabulary. The UCR of the correct transcription in the evaluation set was 1.55%. In this and the further experiments, the ISR delay was evaluated only based on input-wise delay because the computational delay was very short.

Our results show that AT-ISR had the best performance among other ISR approaches. The neural ISR that was trained using forced-alignment with HMM-GMM ASR with a phoneme-level acoustic model cannot perform as well as the other neural ISR approaches. This is because it could not infer the precise alignment of character or subword units, so all units within a word were aligned into a speech segment where that word ends. It implies that some token alignments might be delayed by several speech segments. As a result, if the speech segment window cannot include all necessary segments, the neural ISR was difficult to learn to predict the correct tokens. On the other hand, the AT-ISR and neural ISR, which was trained with alignments from HMM-GMM ASR with a character-level acoustic, learned from more precise alignment. These models could immediately recognize the tokens from a speech segment without delaying it to the next speech segment. Nevertheless, AT-ISR resulted in better performance.

Table 4.1. End-to-end ISR performance on TED-LIUM release 1 eval. set. (*e2e*: End-to-end; 1 block = 8 frames ≈ 0.14 sec)

Output Unit (e2e)	**CER (%)**	**WER (%)**	**UCR (%)**
Topline: Teacher Non-incr. ASR *delay* = 7.58 sec (avg.)			
Characters	15.21	27.37	2.65
Subwords	**13.35**	**24.00**	**0.54**
Baseline: Isolated word recognition *delay* = 0.25 sec (avg.)			
Characters	68.89	80.50	5.60
Subwords	68.38	72.39	0.03
Neural ISR + HMM-GMM alignment (phoneme acoustic model) *delay* = 0.84 sec ([4 main + 4 ahead] blocks)			
Characters	27.89	43.10	1.75
Subwords	28.43	39.77	**0.37**
Neural ISR + HMM-GMM alignment (character acoustic model) *delay* = 0.84 sec ([4 main + 4 ahead] blocks)			
Characters	18.08	34.31	5.99
Subwords	22.13	34.33	0.35
Proposed: AT-ISR			
delay = 0.54 sec ([1 main + 4 ahead] blocks)			
Characters	21.04	41.12	10.22
Subwords	21.28	36.80	0.54
delay = 0.84 sec ([4 main + 4 ahead] blocks)			
Characters	16.62	31.06	4.59
Subwords	**15.19**	**28.26**	0.81

4.6.1 Impact of ISR Output Unit to Speech Recognition Performance

Table 4.1 shows that subword-level model outperformed the character-level model. Although the CERs of character- and subword-level models were close, there were differences in the WER and UCR. In Table 4.1, for example, the CER, WER, and UCR differences between character-level and subword-level non-incremental ASRs were 1.86%, 3.39%, and 2.48% respectively, where the subword-level ASR performance was superior. The subword-level model remarkably resulted in better UCR than the character-level model. Subword sequence is more reliable for

Table 4.2. One-tailed T-test on ISR system. AT-ISR and ISR + HMM-GMM alignment-based models have a delay of 0.84 sec. ($\alpha = 5\%$; *ch* = character-level model; *sw* = subword-level model).

Test Pair		***p*-value**		
Model 1	**Model 2**	**CER**	**WER**	**UCR**
Topline ASR (sw)	Topline ASR (ch)	0.008	0.001	1.6e-20
AT-ISR (sw)	AT-ISR (ch)	0.410	0.010	1.2e-25
AT-ISR (sw)	Baseline (sw)	5.5e-144	4.5e-161	4.1e-10
AT-ISR (sw)	ISR + HMM-GMM alignment-phoneme (sw)	0.003	2.6e-33	0.260
AT-ISR (sw)	ISR + HMM-GMM alignment-character (sw)	1.9e-5	1.2e-11	0.080

forming correct words because it retains a longer context than a character to represent a part of a word. On the other hand, character-level ASR may result in more low-level errors than subwords. As a result, when the characters are concatenated into a word, the chance of forming an uncovered-word is higher than the concatenation from subwords.

The performance difference between character-level and subword-level AT-ISR, our best neural ISR approach, also evaluated through a one-tailed T-test with a significance level (α) of 5%. Table 4.2 shows the test results. It shows that the CER of character-level and subword-level AT-ISR was statistically not different, but the WER and UCR were. AT-ISR also showed statistically significant improvement from other neural ISRs that we explored. In the rest part of this work, we only focus on AT-ISR as our neural ISR.

From our discussion in Chapter 3, we saw that a trade-off between delay and performance in speech recognition does occur. The effect of AT-ISR delay and output unit on the recognition WER can be seen in Figure 4.6. All AT-ISR models in this figure included two look-ahead blocks in addition to the main blocks. Here we made the size of look-ahead blocks shorter than those in Table 4.1 to limit the delay.

In our investigation, we found that character-level AT-ISR performance improvement did not happen significantly between the following delays: 25%, 50%, and 100% utterance lengths. Here when the recognition delay equals 2.04 sec or 25% of utterance length, it also starts to result in comparable WER to the non-incremental ASR's. This result shows that this model is able to retain the balance between recognition delay and performance when the delay is 2.04 sec or

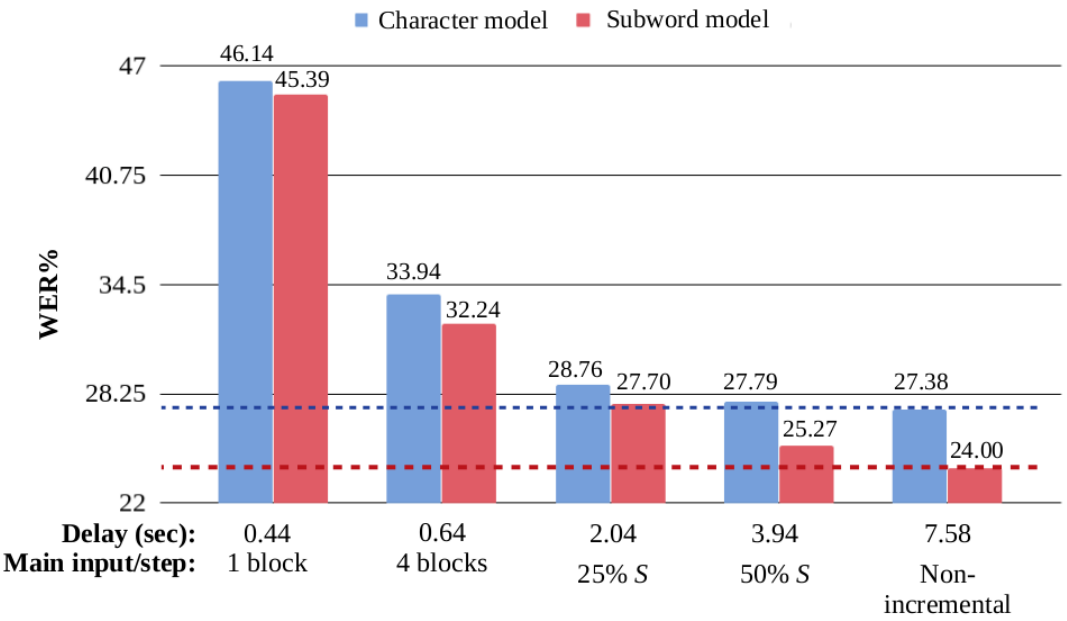


Figure 4.6. WER comparison of end-to-end character- and subword-level AT-ISR based on delay. (1 block = 8 frames ≈ 0.14 sec; S = average speech utterance length in *TED-LIUM release 1* set (7.58 sec))

25% of utterance length.

Interestingly, the subword-level models outperformed the character-level models in general, but the character-level AT-ISR achieved a closer performance to the teacher model with a short delay than the subword-level model. From Figure 4.6, when the AT-ISR delay was 25% of the average full-utterance length, the WER difference between character-level student and teacher models was 1.38%. With an identical delay, subword-level AT-ISR WER was 3.7% higher than the teacher. In the subsequent delays that we explored, the subword-level also did not show a clear balance point between the speed and performance, unlike the character-level AT-ISR.

Character-level AT-ISR is better at mimicking the teacher because the necessary information to predict a character token can be satisfied by a shorter speech segment than for predicting a subword token. Figure 4.7 shows the examples of attention alignment matrix that were generated using character-level and subword-level non-incremental ASR models. In this figure, a subword token

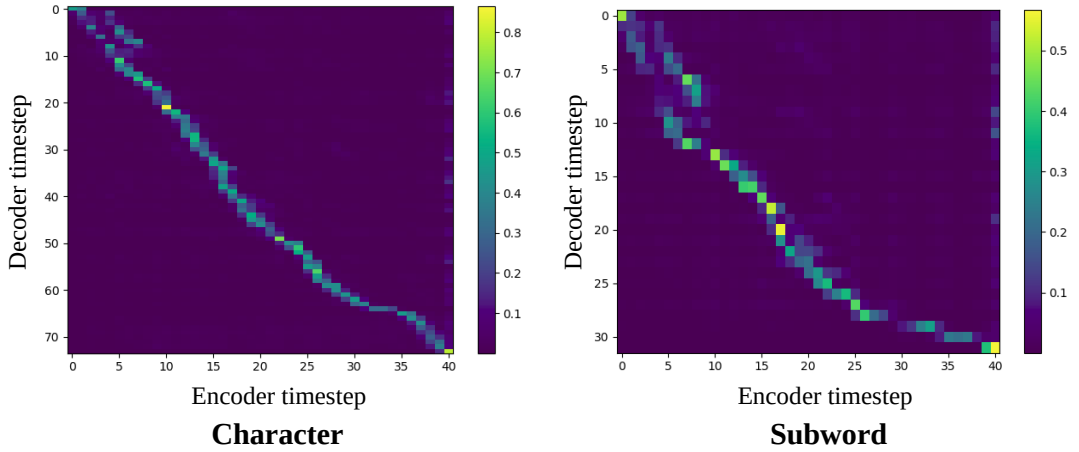


Figure 4.7. Examples of attention matrix by non-incremental ASR from evaluation set. From attention alignment, a text token is aligned to a speech segment, which corresponds to encoder state, with monotonically highest alignment scores.

in subword-level alignment scored a high score to several encoder states, which correspond to a speech segment longer than a character. This is because a subword token consists of several characters. Therefore, the subword-level ISR's performance cannot approach the teacher's level when the input window does not include or fails to reach other speech segments with a high attention score.

Since a subword consists of several characters, the subword-level ISR requires a longer speech context than the character-level ISR. Theoretically, when the input segment is very short, the character-level ISR should be able to result in a better performance than the subword-level ISR. In our experiment, however, the subword-level ISR outperformed the character-level ISR in every delay that we tried. This is because the incremental recognition here included look-ahead blocks in the input segment. In our data, a subword token consisted of seven characters on average, and one speech block was aligned to two characters on average. Our shortest delay used one main block with two look-ahead blocks, which contain the information of six characters on average. For the subword-level model, it might result in a similar amount of information as the character-level recognition. When the recognition delay was below 50% of the average utterance length, the performance difference of the character- and subword-level ISRs was around 1%. So within that delay, the quality of both models is similar, although

the subword-level ISR was slightly better.

4.6.2 Speech Translation Performance

We utilized the AT-ISR in speech translation task. Table 4.3 shows the the speech recognition performance and its translation quality on *tst2010* set for English-French translation task. In this table, '*ch-sw*' denotes character-level AT-ISR that applied character-to-subword conversion before passing its output to the MT.

Table 4.3. Speech recognition and English-French translation performance on *tst2010* set. (1 blocks $\approx$ 0.14 sec; *ch* = characters; *sw* = subwords; *spm* = SentencePiece; *seq2seq* = sequence-to-sequence; d = delay)

ASR Output	**Speech Recognition**			**Translation**		
	CER	**WER**	**UCR**	**BLEU1**	**BLEU4**	**METEOR**
Correct transcription	0.0	0.0	1.36	59.4	31.6	52.0
Topline: Non-incremental ASR (d= 7.58 sec (avg.))						
ch-sw (spm)	15.11	26.75	2.67	47.1	21.1	39.4
ch-sw (seq2seq)	15.42	27.06	1.46	46.9	21.0	39.4
sw	**12.39**	**22.43**	**0.50**	**50.0**	**23.1**	**42.2**
Proposed: AT-ISR						
d= 0.54 sec (1 main + 4 ahead blocks)						
ch-sw (spm)	21.56	41.39	10.07	38.0	13.5	29.8
ch-sw (seq2seq)	23.15	44.3	1.12	40.4	15.5	31.7
sw	**21.52**	**36.56**	**0.60**	**42.6**	**16.3**	**33.4**
d= 0.84 sec (4 main + 4 ahead blocks)						
ch-sw (spm)	19.18	33.09	4.45	44.0	17.9	34.8
ch-sw (seq2seq)	20.25	33.56	1.44	44.3	18.2	35.0
sw	**15.71**	**28.17**	**0.86**	**47.2**	**20.6**	**39.1**

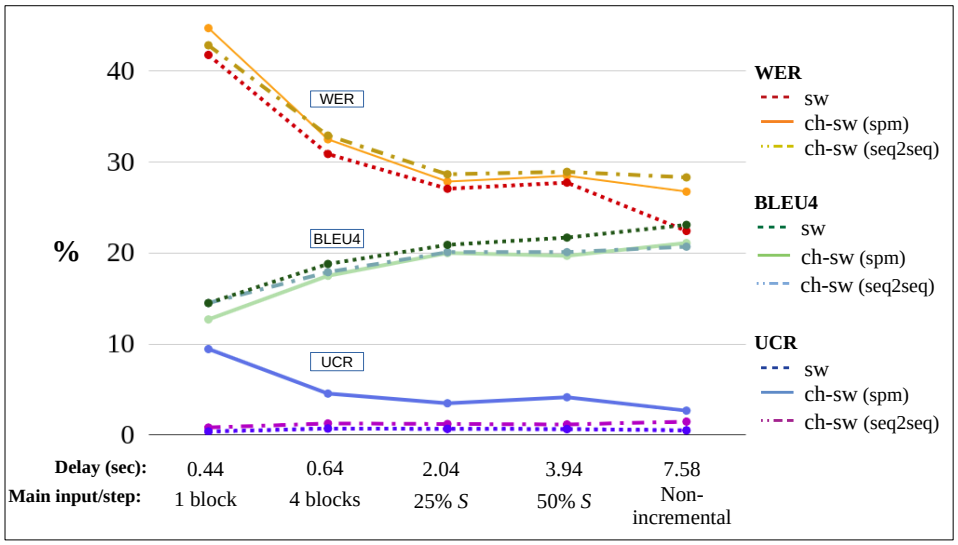
Since the speech recognition output contained errors, the translation quality was lower than the translation from the correct transcription. A low translation quality from the ISR output was caused by the nature of incremental recognition, in which the model was forced to produce outputs based on a short input segment, so the recognition result might be not correct.

Among the three AT-ISR subword-level approaches that we proposed, end-to-end subword-level AT-ISR resulted in the best speech recognition and translation

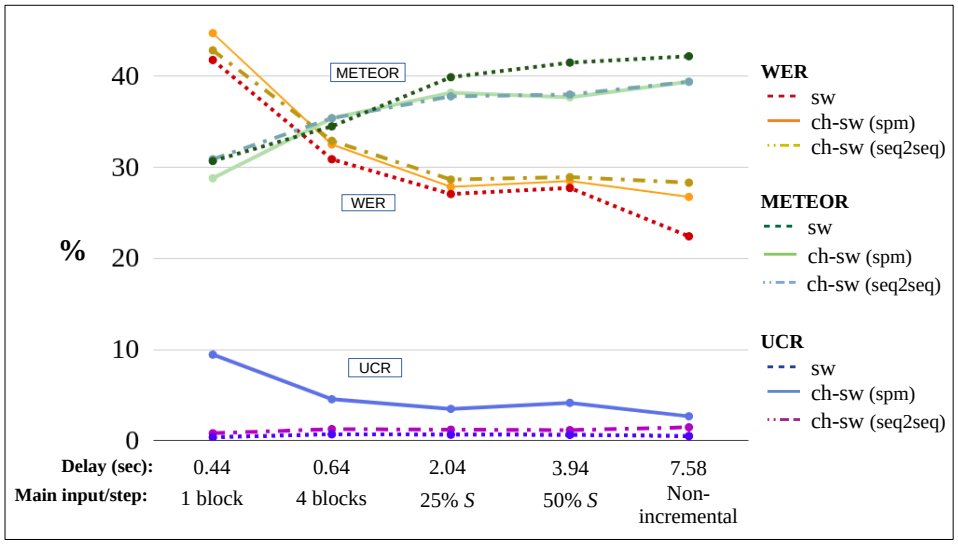
performance. If the original AT-ISR output unit was character, characters mapping into subwords using sequence-to-sequence model resulted in higher CER and WER than the mapping with SentencePiece tokenizer. Nevertheless, character-to-subword mapping with a sequence-to-sequence framework had a better performance in terms of UCR and translation quality than the other characters conversion method. The translation quality improvement by the sequence-to-sequence characters-to-subwords model might be resulted from its low UCR, since this model could correct the content of the character sequence. Therefore, when the end-to-end subword-level ISR is unavailable, output conversion with a sequence-to-sequence model is suitable for speech translation task.

The effect of AT-ISR delay and output conversion approach in the translation task can be seen in Figure 4.8. The AT-ISR delay affected the speech recognition and translation performances, in which a higher delay resulted in a better performance. Here a lower WER resulted in higher BLEU and METEOR scores. The end-to-end subword-level AT-ISR resulted in a better translation result than other approaches. Interestingly, when the AT-ISR delay was 50% of total utterance length, the WER of all models were close, but not the BLEU and METEOR scores. At the identical delays, AT-ISRs that performed character output conversion into subword had higher UCR than the end-to-end subword-level AT-ISR. Therefore, it shows that the translation quality not only depends on the WER but also on the number of uncovered-word. Similar WERs with different UCRs do occur, for example, when the location or the number of errors is similar, but the content of the mistaken words are different. Lower UCR might result in better performance because language translation is basically based on information of words that have meaning. Therefore, when the MT receives a subword sequence that did not have linguistical meaning, it caused a translation error.

In the English-Frech translation task, we can see that end-to-end subword-level AT-ISR has better speech recognition and translation performances than the other approaches. We also investigated how the end-to-end subword-level AT-ISR affects the translation quality for the English-Japanese task. Table 4.4 shows English-Japanese translation quality based on AT-ISR output text. In this table, we limited the ISR input unit to one and four main input segments with four look-ahead segments to keep the latency and performance. The translation result



(a)



(b)

Figure 4.8. AT-ISR speech recognition performance with English-French translation 4-gram BLEU (a) and METEOR (b) scores on *tst2010* set. (1 block $\approx$ 0.14 sec; S = average speech utterance length in *TED-LIUM release 1* set (7.58 sec))

Table 4.4. End-to-end subword-level speech recognition and English-Japanese translation performance on *tst2010* set. (1 blocks ≈ 0.14 sec; d = delay; m: main input block; *la*: look-ahead input block)

ASR Output	**Speech Recognition**		**Translation**		
	WER	**UCR**	**BLEU1**	**BLEU4**	**METEOR**
Correct transcription	0.0	1.36	38.0	12.2	15.1
Non-incremental ASR (d: 7.58 sec)	22.43	0.50	36.6	10.7	14.3
AT-ISR					
d: 0.54 sec (1 m + 4 la)	36.56	0.60	35.5	9.4	13.3
d: 0.84 sec (4 m + 4 la)	28.17	0.86	36.8	10.3	13.9

was evaluated based on tokenized Japanese logographic and syllabic characters. English-Japanese translation is a complex task because of the difference in their word syntax order, so our translation model could not perform as well as the English-French translation. Nevertheless, the English-Japanese translation from AT-ISR text resulted in a translation quality that was close to the translation from non-incremental ASR.

4.7. Summary

In this section, we described approaches to enable the utilization of neural ISR in speech translation task. We proposed neural ISR with matching output vocabulary as MT input vocabulary, specifically, subword-level ISR that has identical vocabulary as subword-level MT, the common MT approach. Based on our experiment of speech recognition task, ISR quality and performance closeness to the teacher depends on the granularity of the output unit. When the output unit has a coarse granularity, such as subword, it might result in higher recognition performance than a model with finer output unit granularity, such as character. On the other side, ISR with a fine-granulated output unit is able to achieve a teacher-like performance within a shorter delay than ISR with a coarse-granulated output unit.

Our experiment shows that translation quality based on ISR output not only depends on WER but is also affected by the number of words in ISR text that are not covered in the training data. Character-level ISR output does not match with MT vocabulary, so it cannot be connected directly to MT. Character sequence

conversion into subwords that can be recognized by MT can be done by the SentencePiece tokenizer or sequence-to-sequence model. Sequence-to-sequence character-to-subword conversion is able to reduce uncovered-words in the original character sequence, so the translation quality might be better than character conversion with SentencePiece tokenizer. Based on our experiment, the end-to-end subword-level ISR achieves the best translation quality with the lowest WER and UCR compared to other investigated approaches.

Chapter 5

Conclusions and Future Directions

5.1. Conclusions

In this thesis, we constructed neural ISR that is able to do a low delay speech recognition. The construction of it was done by employing sources from the standard neural ASR system to achieve a similar mechanism and performance as the standard system, but with the shorter delay. We proposed attention-transfer ISR (AT-ISR) that applies an identical structure to the standard non-incremental neural ASR and learns its attention-based knowledge to perform a short speech-segment-based recognition. Our proposed framework successfully reduced the recognition delay of the standard recognition approach, and it was achieved by less complicated training procedures than the previous neural ISR framework. By adapting the AT-ISR output, we successfully performed speech translation with AT-ISR. We analyzed the effect of low-delay recognition performance on the translation performance.

We began our study with the investigation for the encoding and decoding mechanisms and the model states handling in AT-ISR model. Among the approaches that we explored, the optimum performance was achieved by, for each incremental recognition step, (1) providing the encoder with a look-ahead contextual input segment, (2) using the last actual output token from the prior incremental step as the first input token for decoder, and (3) keeping the model

states across the recognition steps. Based on the experiments on single-speaker speech from *LJ Speech* corpus, our AT-ISR with a delay of 0.54 sec achieved CER 3.13%, which was the closest score to the standard ASR with a CER 1.94% but with an average delay of 6.57 sec. Our experiments with multi-speaker speech from *WSJ* dataset also resulted in a similar result. Using *WSJ* data, an AT-ISR model with a delay of 0.54 sec achieved CER 7.52%, which was close to the standard ASR that had CER 6.26% but with a recognition delay of 7.88 sec.

We investigated the impact of AT-ISR recognition delay to the speech recognition quality. In our experiments, we saw a trade-off between delay and recognition performance, in which a longer delay resulted in a better performance. A long delay, however, is not preferable for real-time recognition or simultaneous S2ST tasks, so we must keep a balance between time and performance. Based on our investigation using *LJ Speech* and *TED-LIUM release 1* datasets, a balance point between delay and performance can be found. At the balance point, the incremental recognition performance is comparable to the non-incremental recognition, and a significant performance improvement does not occur in the subsequent delays. For both datasets, the character-level AT-ISR balance point was when the delay was 25% of a full speech utterance, where the average length of a full speech utterance was around 7 sec. The performance similarity between the AT-ISR and the teacher model also depends on the granularity of the AT-ISR output unit.

We enabled the utilization of ISR in speech translation by investigating several approaches to adapt the ISR output and examining the effect of ISR on the translation performance. Among the explored approach, end-to-end subword-level ISR with a matching vocabulary as the MT subword input vocabulary resulted in the best performance. Based on our experiment results, ISR output with a lower WER and CER resulted in a better translation performance. Our investigation showed that translation quality also depends on the number of uncovered-words that the ISR hypothesis contains. Hypotheses with the similar WER and CER but the different UCR resulted in a dissimilar translation performance, where the hypothesis with a lower UCR resulted in better translation quality.

5.2. Future Directions

There is a lot of room for improving neural ISR. The currently proposed ISR approach performs fix-sized segment-based recognition, and the system delay depends on the incremental unit that the model learns from. We cannot change the incremental unit during inference, for example into a lower size, without sacrificing the model's performance. If we don't have prior information about the acceptable delay, we might need to train several models to find the best model with the lowest delay. However, training different models for different incremental unit configurations can be expensive. Therefore, we will further investigate the mechanisms for flexible incremental unit processing in neural ISR.

This thesis provides attempts in neural ISR construction and the integration of it into MT to achieve simultaneous S2ST in the future. The current MT, which we performed analysis with, was a non-incremental MT that could not be used in the simultaneous S2ST system. Therefore, as our next task, we would like to do the exploration of ISR and incremental MT (IMT). It will be also an interesting task to train ISR and IMT jointly so we can improve the ISR based on the feedback from translation result, and vice versa.

Acknowledgements

I would like to express my gratitude to Professor Satoshi Nakamura for welcoming me to his lab and providing me the best research environment I have ever been to. He introduced me to the world of scientific research in Japan through a summer internship back in 2017 and recommended me for MEXT-IPGP scholarship program. I would not imagine myself in this position without his support. His great insight and leadership inspire me on how I strive to become a good researcher.

I want to thank Research Associate Professor Sakriani Sakti for tirelessly supervising and teaching me various things during my time at NAIST. From her, I learned the knowledge of speech recognition, research method, academic writing and presentation, and other things that made my skills today. Her working style and enthusiasm always inspires me, and I am grateful for her continuous support.

I'm grateful for Professor Taro Watanabe as a member of my master thesis committee, for reviewing this thesis and giving me insightful comments and questions during my research presentation.

I also want to express my gratitude for each member in Augmented Human Communication Lab, my first family in Japan. I would like to thank Associate Professor Katsuhito Sudoh, Research Associate Professor Keiji Yasuda, Assistant Professor Koichiro Yoshino, and Assistant Professor Hiroki Tanaka for the support, discussion, and constructive feedback during my research progress report. My special thank to Ms. Manami Matsuda, who always gave me research and daily supports since my internship time in NAIST, Japan. Her advises helps me able to go through my life in Japan.

Finally, I want to thank my family and friends in Indonesia for giving me emotional supports. Their encouragements always support me during hard times and make me never give up on achieving my dreams.

References

[1] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Proceedings of ICASSP*, pages 4960–4964, Shanghai, China, 2016.

[2] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. Listening while speaking: Speech chain by deep learning. In *Proceedings of ASRU*, pages 301–308, Okinawa, Japan, 2017.

[3] Kyuyeon Hwang and Wonyong Sung. Character-level incremental speech recognition with recurrent neural networks. In *Proceedings of ICASSP*, pages 5335–5339, Shanghai, China, 2016.

[4] Navdeep Jaitly, Quoc V. Le, Oriol Vinyals, Ilya Sutskever, David Sussillo, and Samy Bengio. An online sequence-to-sequence model using partial conditioning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 5067–5075. Curran Associates, Inc., 2016.

[5] Keith Ito. The LJ speech dataset. https://keithito.com/LJ-Speech-Dataset/, 2017.

[6] Douglas B. Paul and Janet M. Baker. The design for the Wall Street Journal-based CSR corpus. In *Proceedings of HLT*, pages 357–362, 1992.

[7] Anthony Rousseau, Paul Deléglise, and Yannick Estève. TED-LIUM: an automatic speech recognition dedicated corpus. In *Proceedings of LREC*, pages 125–129, Istanbul, Turkey, 2012.

[8] Satoshi Nakamura. Overcoming the language barrier with speech translation technology. *Science & Technology Trends - Quarterly Review No.31*, April 2009.

[9] Evgeny Matusov, Arne Mauser, and Hermann Ney. Automatic sentence segmentation and punctuation prediction for spoken language translation. In *Proceedings of IWSLT*, pages 158–165, Kyoto, Japan, 2006.

[10] Mael Pouget, Olha Nahorna, Thomas Hueber, , and Gérard Bailly. Adaptive latency for part-of-speech tagging in incremental text-to-speech synthesis. In *Proceedings of INTERSPEECH*, pages 2846–2850, San Francisco, CA, USA, 2016.

[11] Takatomo Kano, Sakriani Sakti, and Satoshi Nakamura. Structured-based curriculum learning for end-to-end English-Japanese speech translation. In *Proceedings of INTERSPEECH*, pages 2630–2634, Stockholm, Sweden, 2017.

[12] Laura Cross Vila, Carlos Escolano, José A. R. Fonollosa, and Marta Ruiz Costa-jussà. End-to-end speech translation with the transformer. In *Proceedings of IberSPEECH*, pages 60–63, Barcelona, Spain, 2018.

[13] Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. End-to-end speech translation with knowledge distillation. In *Proceedings of INTERSPEECH*, pages 1128–1132, Graz, Austria, 2019.

[14] Chistian Fügen, Alex Waibel, and Muntsin Kolss. Simultaneous translation of lectures and speeches. *Machine Translation*, 21:209–252, 2007.

[15] Takashi Mieno, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Speed or accuracy? A study in evaluation of simultaneous speech translation. In *Proceedings of INTERSPEECH*, pages 2267–2271, Dresden, Germany, 2015.

[16] Satoshi Nakamura, Katsuhito Sudoh, and Sakriani Sakti. Towards machine speech-to-speech translation. *Interpreting Technologies*, (17):81–87, 2019.

[17] Sane Yagi. Studying style in simultaneous interpretation. *Meta*, 45, January 2000.

[18] Henri C. Barik. *A Study of Simultaneous Interpretation*. 1969.

[19] Marianne Lederer. *Simultaneous Interpretation – Units of Meaning and other Features*, pages 323–332. January 1978.

[20] Bhuvana Ramabhadran, Jing Huang, and Michael Picheny. Towards automatic transcription of large spoken archives - English ASR for the MALACH project. In *Proceedings. of ICASSP*, pages I–216 – I–219, Hong Kong, China, 2003.

[21] Jiahong Yuan, Mark Liberman, and Christoper Cieri. Towards an integrated understanding of speaking rate in conversation. In *Proceedings of INTERSPEECH*, pages 541–544, Pittsburgh, PA, USA, 2006.

[22] Daniel Gile. Methodological aspects of interpretation (and translation) research. *Target*, 3:153–174, 01 1991.

[23] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. In *Proceedings of ICASSP*, pages 4774–4778, Calgary, Canada, 2018.

[24] Tsubasa Ochiai, Shinji Watanabe, Takaaki Hori, and John R Hershey. Multichannel end-to-end speech recognition. In *Proceedings of ICML*, pages 2632–2641, Sydney, Australia, 2017.

[25] Takaaki Hori, Jaejin Cho, and Shinji Watanabe. End-to-end speech recognition with word-based RNN language models. In *Proceedings of SLT*, pages 389–396, Greece, Athens, 2018.

[26] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[27] Biing-Hwang Juang and Lawrence R. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[28] Mark Gales, Steve Young, et al. The application of hidden Markov models in speech recognition. *Foundations and Trends® in Signal Processing*, 1(3):195–304, 2008.

[29] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of ICML*, pages 1764–1772, Beijing, China, 2014.

[30] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of ICML*, pages 369–376, Pittsburgh, Pennsylvania, USA, 2006.

[31] Tara N. Sainath, Chung-Cheng Chiu, Rohit Prabhavalkar, Anjuli Kannan, Yonghui Wu, Patrick Nguyen, and ZhiJeng Chen. Improving the performance of online neural transducer models. In *Proceedings of ICASSP*, pages 5864–5868, Calgary, Canada, 2018.

[32] Shinji Watanabe, Marc Delcroix, Florian Metze, and John R. Hershey. *New Era for Robust Speech Recognition: Exploiting Deep Learning*. Springer International Publishing, 2017.

[33] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint CTC-attention based end-to-end speech recognition using multitask learning. In *Proceedings of ICASSP*, pages 4835–4839, New Orleans, USA, 2017.

[34] Jan K. Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Proceedings of NIPS*, pages 577–585, Montreal, Canada, 2015.

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

[36] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional LSTM. In *Proceedings of ASRU*, pages 273–278, Olomouc, Czech Republic, 2013.

[37] Alex Graves. Supervised sequence labelling. In *Supervised Sequence Labelling With Recurrent Neural Networks*, volume 385, pages 5–13. Springer, Berlin, Heidelberg, 2012.

[38] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *Proceedings of ICASSP*, pages 4945–4949, Shanghai, China, 2016.

[39] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[40] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*, pages 1412–1421, Lisbon, Portugal, 2015.

[41] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

[42] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv*, abs/1211.3711, 2012.

[43] Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 535–541, Philadelphia, USA, 2006.

[44] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[45] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *Proceedings of International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[46] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan Kankanhalli. Attention transfer from web images for video recognition. In *Proceedings of the ACM on Multimedia Conference (MM)*, pages 1–9, Mountain View, USA, 2017.

[47] Jianfei Yu, Lu Marujo, Jing Jiang, Pradeep Karuturi, and William Brendel. Improving multi-label emotion classification via sentiment classification with dual attention transfer network. In *Proceedings of EMNLP*, pages 1097—-1102, Brussels, Belgium, 2018.

[48] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. Multi-scale alignment and contextual history for attention mechanism in sequence-to-sequence model. In *Proceedings of SLT*, pages 648–655, Athens, Greece, 2018.

[49] Fabio Brugnara, Daniele Falavigna, and Maurizio Omologo. Automatic segmentation and labeling of speech based on hidden Markov models. *Speech Communication*, 12(4):357–370, 08 1993.

[50] Tamali Banerjee and Pushpak Bhattacharyya. Meaningless yet meaningful: Morphology grounded subword-level NMT. In *Proceedings of the Second Workshop on Subword/Character LEvel Models*, pages 55–60, New Orleans, 2018.

[51] Daniel Dechelotte, Holger Schwenk, Giles Adda, and Jean-Luc Gauvain. Improved machine translation of speech-to-text outputs. In *Proceedings of INTERSPEECH*, pages 2441–2444, Antwerp, Belgium, 01 2007.

[52] Evgeny Matusov and Hermann Ney. Lattice-based ASR-MT interface for speech translation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):721–732, May 2011.

[53] Xiaolin Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. A prototype automatic simultaneous interpretation system. In *Proceedings of COLING*, pages 30–34, Osaka, Japan, 2016.

[54] Xiaolin Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. An efficient and effective online sentence segmenter for simultaneous interpre-

tation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT)*, pages 139–148, Osaka, Japan, 2016.

[55] Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China, 2015.

[56] Sei Ueno, Hirofumi Inaguma, Masato Mimura, and Tatsuya Kawahara. Acoustic-to-word attention-based model complemented with character-level CTC-based model. In *Proceedings of ICCASP*, pages 5804–5808, Calgary, Canada, 2018.

[57] Hainan Xu, Shuoyang Ding, and Shinji Watanabe. Improving end-to-end speech recognition with pronunciation-assisted sub-word modeling. In *Proceedings of ICCASP*, pages 7110–7114, Brighton, United Kingdom, 2019.

[58] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv*, abs/1609.08144, 2016.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008, Long Beach, CA, USA, 2017.

[60] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of ACL (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia, 2018.

[61] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of ACL (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, 2016.

[62] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of EMNLP*, pages 66–71, Brussels, Belgium, 2018.

[63] Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. Overview of the IWSLT 2017 evaluation campaign. In *Proceeding of IWSLT*, pages 2–14, Tokyo, Japan, 2017.

[64] Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. ASPEC: Asian scientific paper excerpt corpus. In *Proceedings of LREC*, pages 2204–2208, Portorož, Slovenia, 2016.

[65] Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. JESC: Japanese-English subtitle corpus. In *Proceedings of LREC*, Miyazaki, Japan, 2018.

[66] Taku. Kudo. Mecab : Yet another part-of-speech and morphological analyzer. *http://mecab.sourceforge.jp*, 2006.

[67] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002.

[68] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of ACL*, pages 65–72, Ann Arbor, Michigan, 2005.

List of Publications

Refereed

- Sashi Novitasari, Quoc Truong Do, Sakriani Sakti, Dessi Lestari, and Satoshi Nakamura. "Multi-Modal Multi-Task Deep Learning For Speaker And Emotion Recognition Of TV-Series Data." Oriental COCOSDA Conference (O-COCOSDA), 2018. Miyazaki, Japan. May 2018.

- Sashi Novitasari, Quoc Truong Do, Sakriani Sakti, Dessi Lestari, and Satoshi Nakamura. "Construction of English-French Multimodal Affective Conversational Corpus from TV Dramas." International Conference on Language Resources and Evaluation (LREC),2017. Miyazaki, Japan. May, 2018.

- Sashi Novitasari, Dessi Lestari, Sakriani Sakti, and Ayu Purwarianti. "Rude-Words Detection for Indonesian Speech Using Support Vector Machine." International Conference on Asian Language Processing (IALP), 2018. Bandung, Indonesia. November, 2018.

- Sashi Novitasari, Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. "Sequence-to-Sequence Learning via Attention Transfer for Incremental Speech Recognition." International Speech Communication Association Conference (INTERSPEECH), 2019. Graz, Austria. September, 2019.

- Sashi Novitasari, Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. "Cross-Lingual Machine Speech Chain for Javanese, Sundanese, Balinese, and Bataks Speech Recognition and Synthesis." Spoken Language Technologies for Under-resourced Languages Conference (SLTU), 2020. May, 2020.

Non-refereed

- Sashi Novitasari, Quoc Truong Do, Sakriani Sakti, Dessi Lestari, and Satoshi Nakamura. "Speaker and Emotion Recognition of TV-Series Data Using Multimodal and Multitask Deep Learning." The 25th Conference on Natural Language Processing (NLP), 2019. Nagoya, Japan.The 25th Conference on Natural Language Processing (NLP), 2019
- Sashi Novitasari, Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. "Neural Incremental Speech Recognition Through Attention Transfer." The 26th Conference on Natural Language Processing (NLP), 2020. Mito, Japan. March, 2020.

Appendices

Appendix A

Data Details

A.1. LJ Speech Dataset

LJ Speech [5] dataset consists of audio clips of a single female speaker reading passages from seven non-fiction books in English language. The audio clips only contain clean "read" speech that does not contain environmental noise and speech hesitation. The passages that were used for the speech recording were texts in the public domain, which published between 1884 and 1964. The audio recording was done in 2016 until 2017. The speech utterances were segmented automatically based on silences. The texts were matched manually and were confirmed through a quality assurance process. The statics of the *LJ Speech* dataset are described in the following Table A.1.

Table A.1. *LJ Speech* data statistics [5].

Total duration	23h 55m 17s
Number of utterances	13,100
Number of speaker	1
Number of words	225,715
Number of distinct words	13,821

A.2. Wall Street Journal Dataset

WSJ speech dataset [6] is a corpus of speech collected to facilitate the development of continuous speech recognition system that recognizes a large vocabulary English speech independently to the speaker. The recorded speech utterances are clean and "read" speeches from newspaper text paragraphs that read by multiple speakers. In the recorded speech, the punctuations from the original text passages were read verbally (e.g., "*period*", "*hyphen*") and were transcribed in the speech transcription as it is. The statistics of *WSJ* sets that were used in this thesis are described in Table A.2.

Table A.2. *WSJ* data statistics [6].

Characteristic	***SI-284***	***dev93***	***eval92***
Total duration	81h	1.1h	0.7h
Number of utterances	37,318	503	333
Number of unique speaker	284	10	8

A.3. TED-LIUM Release 1 Dataset

TED-LIUM release 1 [7] dataset is a speech corpus that collected from English-language TED talks from various domains of talk. The audio data were collected from the recordings of TED talks and the transcriptions were made based on the corresponding talk's closed caption. As the speech data were recorded from talks, the style of speech is not a read speech, so speech hesitations and speech fillers sometimes occur. In the transcription, the speech hesitations and fillers were also transcribed and mapped into specific filler words. Sounds from the audience of the talk, such as applause and laughter sounds, are also recorded in the audio.

The statics of *TED-LIUM release 1* are shown in Table A.3 for the textual data and Table A.4 for the audio data.

Table A.3. *TED-LIUM release 1* corpus textual data statistics [7].

Characteristic	**Train**	**Eval**
Number of talks	774	19
Number of segments	56.8 K	2 K
Number of words	2.56 M	47 K

Table A.4. *TED-LIUM release 1* corpus audio data statistics [7].

Characteristic	**Train**	**Eval**
Total duration	118h 4m 48s	4h 12m 55s
- Male	81h 53m 7s	3h 13m 57s
- Female	36h 11m 41s	58m 58s
Mean talk duration	9m 9s	13m 18s
Number of unique speaker	666	19

Appendix B

ISR Architecture in Related Works

B.1. Unidirectional RNN with CTC

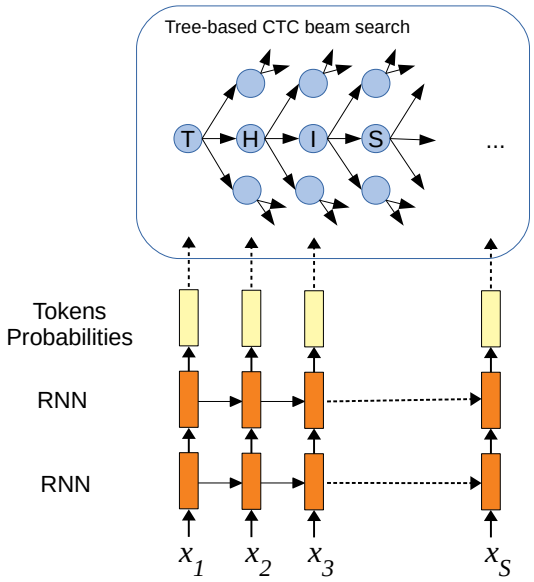


Figure B.1. Unidirectional RNN ISR with CTC-based beam output search [3].

Figure B.1 show the architecture of an ISR that consists of stacked unidirectional RNN layers and trained with CTC objective function [3]. The RNN layers

process the speech sequence unidirectionally and sequentially from the speech frame with earlier timestamp to the later timestamp. Each RNN step outputs a vector of tokens probabilities. To generate the transcription, the model search for the optimal token sequence using a beam search approach based on the CTC state transition.

B.2. Neural Transducer

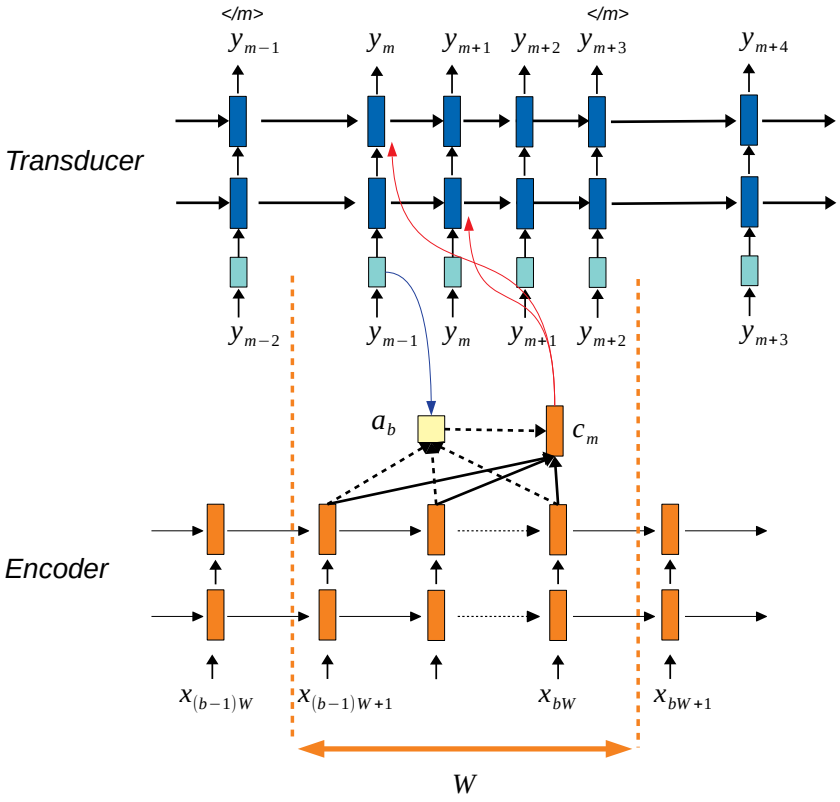


Figure B.2. Neural transducer [4].

Figure B.2 shows the architecture of ISR with NT framework[4]. NT consists of three components: unidirectional encoder, transducer, and attention module. It encodes a speech segment based on a fix-sized window W. The transducer

decodes the encoded speech by taking the previous output token as input. The decoding of a speech segment stops when an end-of-block symbol is predicted.