Incorporating Meta Information for Speech Recognition of Low-resource Language

Kak Soky

Abstract

Automatic speech recognition (ASR) systems have been developed as an aid for communications, not only in human-to-machine interfaces but also in human-to-human interactions. The ASR systems have achieved impressive performance in the last decade with the advancement of deep learning techniques and computing resources. However, the performance is drastically degraded for low-resource languages because of data scarcity, especially in the current trend of end-to-end (E2E) deep neural networks (DNN) architecture, which requires a massive amount of labeled speech data for training. This study addresses the problem of improving the ASR systems for low-resource languages by incorporating meta-information or auxiliary knowledge. Here, meta-information is concerned with the speaker, domain, and language, which can be extracted together with the speech content, whereas the auxiliary knowledge is the translated information from other rich-resource languages.

We first present a large parallel speech corpus of the Extraordinary Chambers in the Courts of Cambodia (ECCC) for transcription and translation in Khmer, English, and French in Chapter 3. We address the problem of sentence segmentation in low-resource languages by conducting bilingual sentence alignment from rich-resource to low-resource language with the monotonic assumption and then enhance the alignment using the ROVER method that combines multiple machine translation (MT) outputs. We also enhance the baseline MT systems of a low-resource language by finetuning the model using the pretrained MT model of the rich-resource languages.

In Chapter 4, we address effective use of speaker information for enhancing ASR systems in the speaker-imbalanced dataset. The proposed approach jointly trains speaker recognition (SRE) and ASR in an E2E model. With a direct connection of SRE to the ASR decoder, it allows for backpropagating the ASR loss to the SRE decoder, resulting in joint optimization. Moreover, conducting speaker clustering can compensate minor speakers, which is beneficial for the speaker-sparse datasets. The proposed method improved the character error rate (CER) of the baseline model by 3.4% relative, with SRE improvement by 8.2% relative.

In Chapter 5, we present the effective finetuning of a large-scale pretrained model for low-resource language ASR with very low-resource settings. The finetuning process is composed of two-step adaptation: domain adaptation and language adaptation, using heterogeneous datasets which are matched with either domain or language. We incorporate meta-information such as domain and language in multi-task learning or adversarial learning for effective adaptation. Moreover, the fusion of domain or language identification to the ASR decoder is effective. The proposed method outperformed the naive adaptation in the CER relatively by 31.8%, 16.3%, and 9.3% for one-hour, 5-hour, and 10-hour target speech datasets, respectively.

In Chapter 6, we present an effective framework of incorporating the translation knowledge from rich-resource languages to improve the transcription of a low-resource language in multi-lingual scenarios. It assumes that the content of its back-translation is the same as the transcription of the original speech. We formulate this framework as a joint process of ASR and MT with the cross-attention mechanism of the decoder module. The proposed method improved the word error rate (WER) of Khmer and Spanish relatively by 8.9% and 1.7%, respectively.

Chapter 7 concludes the thesis and a brief look at future work.

Acknowledgment

This dissertation was accomplished at the Speech and Audio Processing Laboratory, Graduate School of Informatics, Kyoto University. It would not have been possible without support, encouragement, and contributions from many different people in different ways. I would like to express my deepest appreciation to the following people who helped me and this work.

First and foremost, I would like to express my profound gratitude and deep regards to my supervisor, Professor Tatsuya Kawahara, who has supported me throughout my Ph.D. life with his exemplary guidance, comments, feedback, and constant encouragement. He gave me the opportunity to join the Speech and Audio Processing Laboratory and it changed my life as a researcher. Since the start of my Ph.D. course, he has patiently given me a chance to discuss my research ideas, even late at night. Moreover, he taught me how to write papers and present the research works as the way of professional did. In improving the quality of the papers, he always kindly helped me proofread my research papers right before the submission deadline many times. The help and guidance given by him from time to time shall carry me a long way to complete this dissertation.

I also express my special thanks and appreciation to Dr. Sheng Li at the National Institute of Information Communications and Technology (NICT) for a lot of insightful advice and collaboration on my research. I met him for the first time at NICT, during my internship in 2017, and after that, he gave me a great opportunity to do an internship again for two months in the summer of 2020. I started to learn more deeply about automatic speech recognition during my internship, and I was able to collaborate with him since then. I further would like to thank Associate Professor Chenhui Chu for his support, comments, suggestions, and collaborations in my major publications during my Ph.D. studies. I could not accomplish this dissertation without their support.

Furthermore, I express my special thanks and appreciation to the members of my dissertation committee: Professor Sadao Kurohashi and Professor Shinsuke Mori for their time in carefully checking this dissertation and providing advice during my presentation.

I also express my thanks to the members of the Speech and Audio Processing Lab. I would like to thank Dr. Masato Mimura and Dr. Shinsuke Sakai for their time to discuss and support a lot of technical problems. I am grateful for the comments and support from Associate Professor Kazuyoshi Yoshii, Assistant Professor Koji Inoue, Ms. Mayumi Abe, Dr. Hirofumi Inaguma, Dr. Sei Ueno, Mr. Kenta Yamamoto, Mr. Hao Shi, Mr. Gao Yuan, Mr. Jumon Nozaki, and all the other members.

I wish to deeply thank the Cambodia Academy of Digital Technology (formerly NIPTICT), Cambodia, for giving me the raw resources of the Extraordinary Chambers in the Courts of Cambodia (ECCC), which was used in all of my major publications in my Ph.D. course.

Thank you to all my friends in Japan for your thoughtfulness and encouragement. Your friendship is a gift in my life during that critical time. We have shared a lot of unforgettable memories and made a life full of joy and happiness.

Last but not least, I would like to express my deepest and warmest thanks to my beloved families; especially my lovely son, daughter, wife, and my parents for their unconditional support and love that make all things possible for me in this world.

Contents

Abstract i

Acknowledgment iii

Contents viii

List of Figures x

List of Tables xii

1 Introduction 1

1.1 Background 1

1.2 Progress of ASR Technology 2

1.3 Challenges in Low-resource ASR 3

1.4 Approaches 4

1.4.1 Leveraging Multi-lingual Parallel Datasets 4

1.4.2 Exploiting Heterogeneous Datasets 5

1.4.3 Using Meta Information 5

1.5 The Thesis Outline 6

2 Literature Review 9

2.1 Automatic Speech Recognition (ASR) 9

2.1.1 Hybrid ASR System 10

2.1.2 End-to-End ASR System 12

2.1.3 Connectionist Temporal Classification 13

2.1.4 Attention-Based Encoder-Decoder Model 15

2.1.5 Joint CTC/Attention training 16
2.1.6 Transformer 16
2.2 Translation Tasks 19
2.2.1 Machine Translation (MT) 19
2.2.2 Speech Translation (ST) 21
2.3 Speech Classification 23
2.3.1 Speaker Identification 24
2.3.2 Language Identification 24
2.4 Multitask Learning in ASR System 24

3 Trilingual Corpus for Speech Recognition and Translation Studies 27
3.1 TriECCC Corpus 29
3.1.1 Khmer Language 29
3.1.2 ECCC Background 30
3.1.3 Corpus Creation and Key Statistics 31
3.2 Baseline End-to-End ASR, MT, and ST Systems 36
3.2.1 Automatic Speech Recognition (ASR) 38
3.2.2 Machine Translation (MT) 39
3.2.3 Speech Translation (ST) 39
3.3 Enhancement of MT 40
3.3.1 Methods 40
3.3.2 Experimental Evaluations 43
3.4 Monolingual ASR Corpus of Khmer ECCC 47
3.5 Conclusions 50

4 Incorporating Speaker Information for ASR in Speaker-imbalanced Corpus 51
4.1 Joint Speaker and Speech Recognition 53
4.1.1 Multi-Task Learning (MTL) 53
4.1.2 Adversarial Learning (AL) 54
4.1.3 Speech Attribute Augmentation (SAug) 55
4.2 Proposed Method 55

4.3 Experimental Evaluations 56
4.3.1 Data Setup 56
4.3.2 System Configurations 58
4.3.3 Results and Discussions 59
4.4 Conclusions 61

5 Incorporating Domain and Language Information for Adapting ASR using Heterogeneous Datasets 63
5.1 Finetuning Pretrained Model for ASR systems 65
5.2 Proposed method 66
5.2.1 Adaptation using Heterogeneous Datasets 66
5.2.2 Multi-task Learning (MTL) 66
5.2.3 Adversarial Learning (ADV) 68
5.2.4 Embedding Domain and Language ID for ASR Decoder . 69
5.2.5 Incremental Two-step Adaptation 70
5.3 Experimental Evaluations 71
5.3.1 Datasets 71
5.3.2 System Settings 71
5.4 Results 73
5.4.1 Baseline One-step Finetuning 73
5.4.2 MTL and Adversarial Learning (ADV) 76
5.4.3 Two-step Finetuning 76
5.5 Conclusions 78

6 Leveraging Simultaneous Translation for Enhancing Transcription of Low-resource Language 81
6.1 ASR Enhancement using MT Task 83
6.2 Proposed Method 83
6.2.1 Dual Encoders 85
6.2.2 Joint Decoder 85
6.2.3 Objective Function 86
6.3 Experimental Evaluations 86

6.3.1 Dataset 86
6.3.2 Model Training 87
6.3.3 System Evaluation 87
6.3.4 Application to Fisher-CallHome-Spanish 91
6.4 Conclusions 91

7 Conclusions 93
7.1 Summary of Thesis Contributions 93
7.2 Limitations and Future Research Directions 94

Bibliography 97

List of Publications 111

List of Figures

1.1 The organizing structure of this thesis 7

2.1 The flow of ASR system 9

2.2 The overview of the traditional ASR pipeline 10

2.3 The overview of the End-to-End ASR pipeline 12

2.4 Overview of the End-to-End architectures 12

2.5 Transformer-based ASR model architecture 17

2.6 The flow of ST system 21

2.7 The flow of ASR system 23

2.8 Overview of multi-task learning architectures 25

3.1 The process of creating the ECCC corpus: a: bilingual sentence alignment, b: text-to-speech alignment and segmentation 31

3.2 Source to target sentence alignment examples 33

3.3 Speaker distribution in a trilingual SLT corpus of the ECCC . . . 35

3.4 The finetuning process: a: the single parent, b: the multiple parents 42

3.5 Examples of all methods in English-Khmer MT models, the *italic text* in the “()" is the translated text into English. 46

3.6 Examples of all methods in French-Khmer MT models, the *italic text* in the “()" is the translated text into English. 47

3.7 Process of creating monolingual speech-to-text of the ECCC corpus 48

4.1 Overview of joint speaker and speech recognition methods. MTL: multi-task learning, AL: adversarial learning, SAug: speech attribute augmentation. 53

4.2 Proposed method. $\oplus$ denotes the “sum" operation. 56
4.3 Speaker distribution in the ECCC; five major speakers are the president of the chamber, the accused, and three interpreters. . . 57

5.1 The proposed method of adaptation with multi-task learning (MTL) for language adaptation. There are two options for MTL: simple MTL and MTL with ID embedding, where ID can be the domain ID in language adaptation or language ID in domain adaptation. 67
5.2 The proposed method of adaptation with adversarial learning (ADV) for domain adaptation. There are two options for ADV: simple ADV and ADV with language ID embedding. 68
5.3 The proposed method of two-step adaptation using heterogeneous datasets. Step 1: domain adaption with multilingual in-domain datasets, Step 2: language adaptation with multi-domain datasets of the same target language. Domain adaptation and language adaptation can be done in different orders. 70

6.1 Proposed method of joint ASR and MT 84
6.2 Examples of the comparison of all methods in the Khmer language, the *italic text* is the translated text into English. 89

List of Tables

3.1 CER in source to target sentence alignment 32

3.2 Statistics of data reduction by the alignment based on the English sentences 33

3.3 Statistics of each source language in the trilingual ECCC SLT corpus 36

3.4 Word error rate (WER) of the ASR models in Khmer, English, and French; '*' model is used in cascade-ST and E2E-ST 38

3.5 BLEU for translation of each language pair in a TriECCC corpus . 39

3.6 Result of ROVER method for MT outputs of Khmer 43

3.7 The best practice in the use of the pretrained model to initialize each Khmer MT model, **bold-font** module is transferred (Enc.-Dec.: Encoder-Decoder) 44

3.8 Comparison of the best performance of finetuning approach initializing the pretrained model into the encoder, decoder, or both on each Khmer MT model 44

3.9 CER (%) and WER (%) of Khmer ASR baseline on ECCC 49

4.1 Data statistics used in these experiments 57

4.2 Comparison of all systems for SRE and ASR. 60

4.3 Comparison of embedding options applied to all layers in the proposed method (Gr6) 60

4.4 Comparison of layer-wise applications of AC option of the proposed method (Gr6) 61

5.1 The heterogeneous datasets used in these experiments 72

5.2 Effect of domain and language adaptation in a single step 73

5.3 Effect of MTL or adversarial learning (ADV) and domain or language ID embedding in a single-step adaptation. (DID: domain identification, LID: language identification) 75

5.4 Results of two-step adaptation using only one-hour matched training dataset. (n, m) represents the number of hours in adaptation data of the first step (n) and the second step (m). (DID: domain identification, LID: language identification), "+ ID": ID embedding 77

5.5 Results of two-step adaptation using 5-hour and 10-hour matched training datasets. (DID: domain identification, LID: language identification) 77

6.1 WER (%) of Khmer ASR on ECCC test set; ** and * indicates statistically significant difference with $p < 0.01$ and $p < 0.05$ from baseline, respectively. 88

6.2 ASR improvement with proposed method for each group of speakers. 88

6.3 ASR improvement with the proposed method in accordance with baseline WER distribution. 88

6.4 ASR improvement with proposed method in accordance with MT BLEU distribution (English-to-Khmer). 89

6.5 WER (%) of speech transcription on Fisher-CallHome Spanish test set. 90

Chapter 1

Introduction

1.1 Background

There are over 7000 living languages across the world [1] that can be communicated in the form of speech and text. Speech-based communication is the most universal and inclusive means in our daily lives because it is an easy, quick, flexible, inexpensive, and effective way to exchange information, ideas, and feelings, and also to create and share meaning, whereas text-based communication is generally for formal and explicit exchanges. To fill the gap between these two media, the demand for automatic speech transcription has been significantly increasing, especially in events such as lectures, meetings, and court proceedings. Thus, the interest in automatic speech recognition (ASR) systems has significantly grown over the last decades.

The ASR systems are useful not only in human-to-machine interfaces but also in human-to-human interactions. For instance, many people use their speech to interact with other devices such as smartphones, smart speakers, and car navigation systems through voice assistants (*e.g.*, Siri and Cortana) or their embedded applications (*e.g.*, Google Home and Amazon Alexa). These applications work smoothly and have high performance in high-resource languages such as English, Japanese, Mandarin Chinese, and other major languages. However, the low-resource languages, mainly in Asia and Africa, which have about 3 billion speakers (68% of the world population),[1] are still underserved by natural

[1] https://medium.com/neuralspace/low-resource-language-what-does-it-mean-d067ec85dea5

language processing (NLP) systems including ASR because of various challenges to build accurate state-of-the-art systems.

1.2 Progress of ASR Technology

Automatic speech recognition (ASR) is a task to decipher speech content into text, and it is often called speech-to-text. To transcribe the speech, ASR is mapping a sequence of acoustic features into the most likely sequence of tokens (character, word, word piece, and so on).

The studies of the ASR systems have been conducted for many decades [2–5]. They were investigated based on pattern matching such as dynamic programming (DP) and the effective acoustic features in the early stage. Then, statistical models of the Hidden Markov model (HMM) have been introduced using the Gaussian mixture models (GMM) to model each segment of acoustic patterns. With the advancement of computing resources, deep neural networks (DNNs) have been used to replace GMM. This replacement has drastically improved the performance of ASR systems, which are widely used for many applications. This hybrid model (GMM-HMM/DNN-HMM) incorporates an acoustic model (AM), a pronunciation lexicon that maps phones into words, and a language model (LM) to rank the likelihood of words. To achieve high performance, it is necessary to design the pronunciation lexicon, LM, and AM carefully. However, each module is optimized independently with a different criterion.

In the last decades, end-to-end (E2E) modeling [6–10] have been significantly improved the ASR systems. It solves the complex problem of sequence labeling between the input speech and output labels by integrating all models of AM, pronunciation lexicon, and LM, into a single model. These models learn the ASR tasks efficiently as the whole model is optimized based on the unified criterion.

However, the E2E modeling requires a massive amount of training data. Moreover, the ASR performance depends on the speaker, environment, domain, language, and so on [11]. For instance, the ASR performance is impacted by many factors such as paralinguistic information in speech (*e.g.*, such as disfluencies, fillers, and laughter), speaking style, speech rate, and recording conditions (e.g., number of microphones, distance from speakers to microphones, room reverberation, and the noise levels). The performance also depends on the applications such as lecture transcription, meeting transcription, telephone conversation, video captioning, voice assistants, and dialogue systems.

1.3 Challenges in Low-resource ASR

The major problem in low-resource languages is *data scarcity*. Although the tons of data is significantly increasing online, a huge amount of parallel resources of speech and text pairs is currently available for a limited number of languages. Ideally, to build an accurate ASR model, we need a training dataset of transcribed speech of more than 1000 hours [12] matched to the language, domain, and application, which is only available for a few rich-resource languages, while there are many low-resource languages left behind with many challenges:

Lack of annotated datasets: For a supervised training fashion of the ASR models, it is necessary to have the annotated datasets of speech-text pairs. The ASR model is prepared to solve only specific target domains including speaking styles, recording environments, and so on. However, creating annotated datasets requires human intervention by labeling training samples one by one, making the process usually time-consuming and very expensive given the millions of samples. Thus, it becomes infeasible to rely on only manual data creation in the long run.

Lack of unlabeled datasets: Recently, unsupervised or self-supervised training schemes have also been investigated to exploit unlabeled datasets. However, they require a much larger scale of datasets.

Lack of language processing toolkits: Language processing toolkits such as sentence tokenization [13, 14], sentence aligment [15], and speech alignment [16], are necessary, but not available for those low-resource languages.

Lack of speaker diversity: To train the universal ASR model, a large number of speakers are necessary, but the number of speakers is usually limited or imbalanced [17].

1.4 Approaches

In this section, we highlight our main approaches to address the data scarcity problems in low-resource languages. The general approach to improve ASR in low-resource languages is to train shared ASR models with multi-lingual datasets [18–20], which consist of languages related to the targeted language [21]. Another popular approach is to use untranscribed speech data for training in a semi-supervised or unsupervised way. The model is then finetuned on the target label datasets of the domain or language in low-resource settings to fit its specificities [22].

In addition to these approaches, this thesis explores new approaches as follows.

1.4.1 Leveraging Multi-lingual Parallel Datasets

For low-resource languages, translation to major languages by a human is often available, which can be used to build a parallel corpus of multi-lingual spoken language translation (SLT). However, it requires two processes: bilingual sentence alignment and speech-to-text alignment. Sentence alignment requires good language processing tools, but this assumption does not hold for most low-resource languages. Speech-to-text alignment requires timestamp information for the speech data that corresponds to each sentence of the text.

In Chapter 3, we present this corpus development, where good language processing toolkits and timestamp information are not available. Specifically, we build the SLT corpus of the Extraordinary Chambers in the Courts of Cambodia (ECCC) in three languages: English, French, and Khmer. As the ECCC is a simultaneous translation, we expect that the alignment can be conducted in a monotonic and continuous. Moreover, this corpus is multi-lingual, thus using the information of rich-resource languages such as English, which has good language processing toolkits as the source language, should be effective for bilingual sentence alignment.

Recently, multi-lingual training has shown to be effective for many tasks including ASR [23–28], MT [29–31], and ST [32, 33], especially in low-resource languages. With ECCC, we can conduct the translation and transcription for the same language output (*e.g.*, MT/ST of English-to-Khmer and ASR of Khmer). The combination of these outputs can enhance each other. We formulate this process as joint training and inference of ASR and translation tasks in Chapter 6.

1.4.2 Exploiting Heterogeneous Datasets

When the matched datasets for a specific task are limited, it is reasonable to borrow resources and knowledge from other languages and other domains. We often have access to *heterogeneous datasets*, which are partially matched each of them even in low-resource languages.

Inspired by multi-lingual training, which usually trains the systems by combining multiple languages together [28]. We finetune a large-scale pretrained model by combining the heterogeneous datasets in very low-resource settings, which include the target dataset, multi-lingual in-domain dataset, and out-of-domain in-target language dataset. This investigation is conducted in Chapter 5. It is too complex to combine multi-lingual and multi-domain of heterogeneous datasets simultaneously. Thus, we formulate separate adaptation steps, in which domain adaptation uses domain-matched multi-lingual datasets, and language adaptation uses language-matched multi-domain datasets. Each adaptation step conducts domain and language adaptations individually.

1.4.3 Using Meta Information

Speech contains a lot of *meta information* including language, speaking style, domain, accent, and gender. Speech content and meta-information can be deciphered together in parallel. Moreover, with meta-information awareness, it is often easy to recognize the speech content (*e.g.*, We will be able to catch up with the speech content more precisely when we know who is speaking or what language of the speech). We investigate if incorporating meta-information is effective in enhancing the ASR systems.

We first explore the incorporation of speaker information into ASR in Chapter 4. This is also inspired by speaker embedding, which can enhance ASR performance. We then explore the benefit of incorporating language or domain identification into the ASR system in Chapter 5. This meta-information identification is expected to improve ASR performance because it is an easy task.

1.5 The Thesis Outline

The organization of this dissertation is outlined in Figure 1.1. Chapter 3 addresses the dataset construction of a spontaneous speech corpus, which lacks language processing toolkits and timestamp information. Chapter 4 addresses the problem of speaker imbalance and presents the usefulness of speaker information embedding for ASR. Chapter 5 discusses the challenge of low-resource settings in training the E2E model and presents effective adaptation methods with meta-information incorporation. Chapter 6 addresses the use of rich-resource knowledge to enhance the transcription of low-resource language in the multilingual datasets.

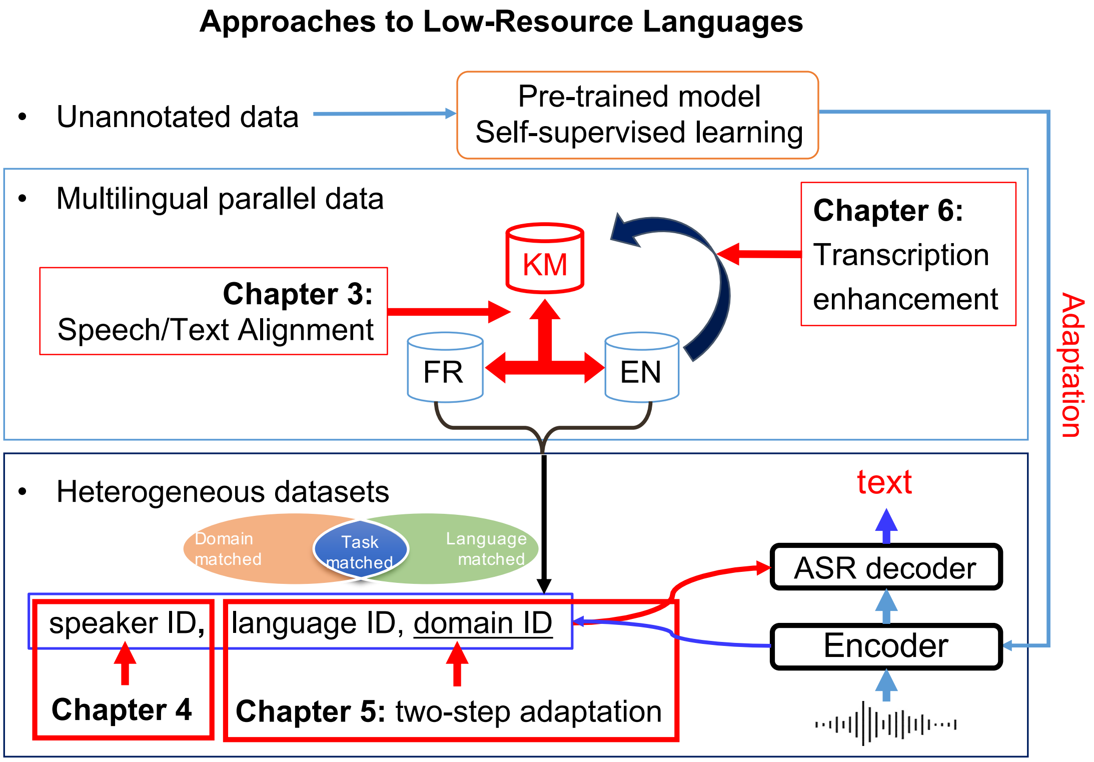


Figure 1.1: The organizing structure of this thesis

Chapter 2

Literature Review

This chapter reviews the downstream tasks which will be investigated in this dissertation. The systems mainly use speech as input including ASR, speech translation (ST), and speech classification.

2.1 Automatic Speech Recognition (ASR)

Automatic Speech Recognition (ASR), also known as Speech to Text, is the task of transcribing given audio to text as illustrated in Figure 2.1.

Let $\mathbf{X}$ = $(x_1, ..., x_T)$ denotes an input speech sequence of lengths T. Let $\mathbf{y}$ = $(y_1, ..., y_L)$ denotes a target label sequence of lengths L, where $y_l \in \{1, ..., K\}$ and K is the number of target labels. The ASR model generally tries to find the most plausible word sequence $\hat{\mathbf{Y}}_{\text{src}}$ given an input speech $\mathbf{X}$ by mapping the input $\mathbf{x}$ to the target label y_l at time t.

$$\hat{\mathbf{Y}}_{\text{src}} = \arg\max_{\mathbf{Y}_{\text{src}}} P_{\text{ASR}}(\mathbf{Y}_{\text{src}}|\mathbf{X}), \tag{2.1}$$

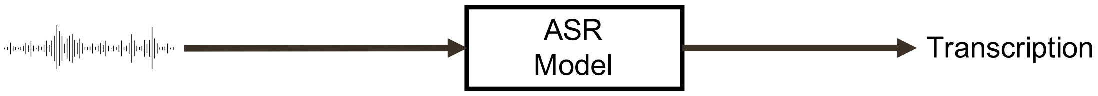


Figure 2.1: The flow of ASR system

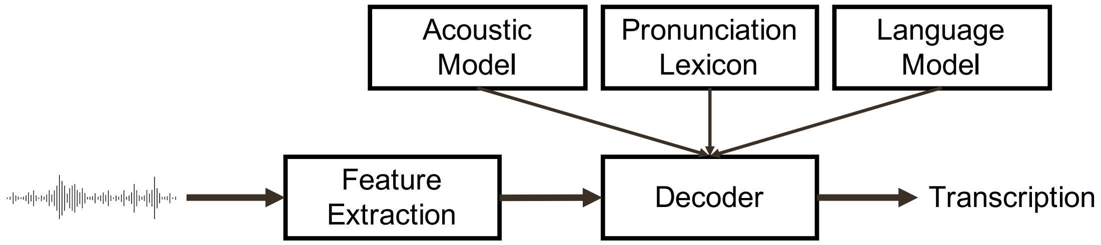


Figure 2.2: The overview of the traditional ASR pipeline

2.1.1 Hybrid ASR System

The ASR task has been an active research topic since the 1980s. The conventional approach of ASR is a hybrid system that is composed of three independent components [34], namely, acoustic model (AM), language model (LM), and pronunciation model (PM) as presented in Figure 2.2. All of which are independently trained, and often manually designed with different datasets. The AM takes acoustic features and predicts a set of subword units, typically context-dependent or context-independent phones. Next, a hand-designed lexicon (the PM) maps a sequence of phones produced by the acoustic model to words. Finally, the LM assigns probabilities to word sequences.

Let us denote the optimal word sequence $\hat{\mathbf{W}}$ from the vocabulary list and the input sequence of the acoustic features $\mathbf{X}$. The main objective is to identify the optimal word sequence, thus the Equation (2.1) can be rewritten as:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}} \, P(\mathbf{W}|\mathbf{X}), \tag{2.2}$$

In the fundamental principle, a word sequence ($\mathbf{W}$) is determined with a minimal posterior probability $P(\mathbf{W}|\mathbf{X})$. However, it is difficult to calculate the $P(\mathbf{W}|\mathbf{X})$ directly, thus the Bayes' rule can be applied and Equation (2.2) can be reformulated as:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}} \, \frac{P(\mathbf{X}|\mathbf{W})P(\mathbf{W})}{P(\mathbf{X})}, \tag{2.3}$$

Usually, $P(\mathbf{X})$ does not affect the choice of the $\hat{(}\mathbf{W})$, we thus remove the $P(\mathbf{X})$. Hence, we can redefine the Equation (2.3) as:

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmax}}\ P(\mathbf{X}|\mathbf{W})P(\mathbf{W}), \tag{2.4}$$

Where $\underset{\mathbf{W}}{\mathbf{argmax}}$ is the search space function of the vocabulary, $P(\mathbf{X}|\mathbf{W})$ is the acoustic model, and $P(\mathbf{W})$ is given by the language model.

In the conventional ASR systems, there are two revolutions in hybrid acoustic modeling. The first one started after applying the Hidden Markov Model (HMM) to the AM. The HMM has several states including self-transitions. The transition probabilities between states are defined and observation probabilities are computed for input features in each state. The observation probability density is usually represented by Gaussian mixture models (GMM). This approach is usually called GMM-HMM. It played an important role in speech recognition and became the mainstream acoustic model. For a sequence of input features, the resulting probability is calculated by multiplying transition probabilities and observation probabilities.

In GMM-HMM, each phone is modeled with an HMM. However, articulation depends on the neighbor phones before and after the phone which means that sounds change according to the surrounding contexts. This context-dependent (CD) HMM is usually called a triphone. with questions about the left and right context, the triphone modeling can be clustered. Then, the states of HMMs can be shared to reduce the parameter space of CD triphones. Thus, the emission probability is defined for the tied triphone states. The GMM-HMM systems use GMMs as observation functions for HMMs.

There are several methods to apply neural networks to speech recognition. In 2009, the second revolution of hybrid ASR systems started after Deng et al. [35] proposed the use of deep learning in speech recognition. The neural network became a research upsurge of speech technology, which turned from the ANN (artificial neural network) to the deep neural network (DNN). The DNN-HMM hybrid architecture replaces the GMM with the DNN. It became the main acoustic model by showing strong recognition capability around 2010 [36]. In this framework, the HMMs capture the temporal dynamics of the speech signal and the DNN estimates the observation probabilities given the acoustic

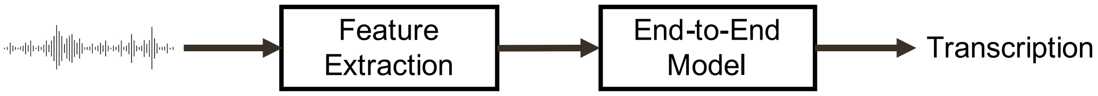
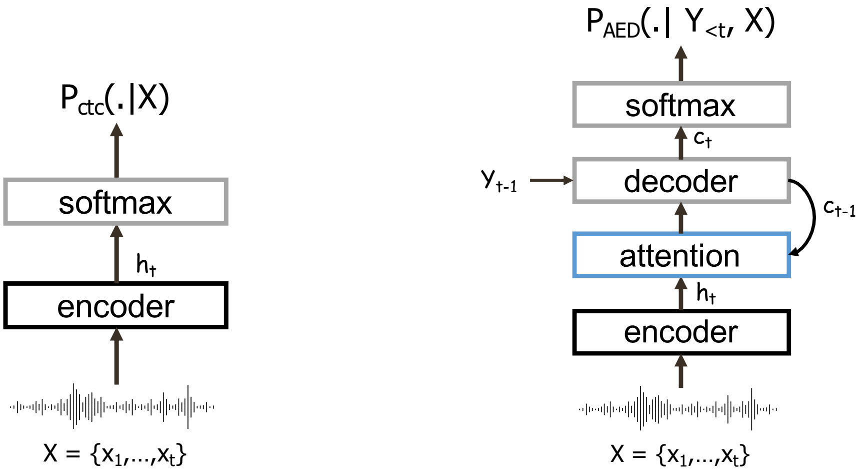


Figure 2.3: The overview of the End-to-End ASR pipeline



a: CTC

b: AED

Figure 2.4: Overview of the End-to-End architectures

features. Each output node of the DNN corresponds to the tied triphone state.

In training, the DNN can be regarded as a feature extractor by feeding the speech feature (MFCC or filterbank) into DNN and transforming it into the posterior probabilities. Since the HMM requires the likelihood instead of the posterior probability during the decoding process, it is necessary to convert the posterior probability to the likelihood by dividing it with a prior probability of each tied-state estimated from the training set. The prior probability of each tied-state can be calculated by counting the number of frames based on the Viterbi algorithm.

2.1.2 End-to-End ASR System

In the previous approach, building a speech recognition system is a complicated process involving training separate models for AM, LM, and PM, which requires a lot of professional knowledge. Moreover, the error of each model may not behave well with errors in another component and leads to a bad effect on the overall performance of the ASR system. Thus, various attempts have been made in recent years to reduce the complexity of ASR, with the scheme of directly mapping speech to a target label sequence. The first successful attempt at End-to-End (E2E) modeling was presented as early as 2006 by Alex Graves et al. [6], namely connectionist temporal classification or CTC in short. E2E speech recognition greatly simplifies the complexity of traditional speech recognition. This means that there is no need to train AM, LM, and PM separately. The neural network can automatically learn language or pronunciation information in a single model as shown in Figure 2.3. Now there are four main approaches for end-to-end speech recognition: CTC [5,6], transducer model [7], attention-based encoder decoder model [8,37,38], and Transformer-based model [9].

2.1.3 Connectionist Temporal Classification

Connectionist temporal classification (CTC) was proposed by Graves et al [6]. It is a kind of objective function for labeling a sequence problem in RNN-based model training. The core concept of CTC is an alignment-free one-to-one mapping that maps an audio frame to a relatively high-level representation. While in early research, the acoustic model training using CTC as the loss function is an end-to-end training, which does not need to align the data in advance, but only needs an input sequence and an output sequence to be trained. The CTC-based model is typically a decoder-free architecture that stacks a linear projection layer on the top of the encoder to generate a probability distribution P_{CTC} as in Firgure 2.4a. This model is particularly attractive for its fast decoding due to the non-autoregressive prediction.

In the ASR system, the length of input features $\mathbf{X}$ is generally longer than that of the output sequence $\mathbf{Y}_{\text{ASR}}$. To bridge this gap, the CTC-based model introduces a special token called "blank" (ϕ) for no predicted token at this frame. In the prediction step, it allows repetitions of the same label, possibly interleaved with ϕ tokens.

In this model, these outputs define the probabilities of all possible ways of aligning all possible label sequences with the input sequence. The total probability of one label can be calculated by summing the probabilities of its different alignments.

$P(\mathbf{Y}_{\text{ASR}}|\mathbf{X})$ is marginalized using the probabilities of all possible alignment in $\Omega(\mathbf{Y}_{\text{ASR}})$ as:

$$p(\mathbf{Y}_{\text{ASR}}|\mathbf{X}) = \sum_{\boldsymbol{\pi} \in \Omega(\mathbf{Y}_{\text{ASR}})} p(\pi|\mathbf{x}) = \sum_{\boldsymbol{\pi} \in \Omega(\mathbf{Y}_{\text{ASR}})} \prod_{t=1}^{T} p(\pi_t|\mathbf{x}_t) \tag{2.5}$$

where $\boldsymbol{\pi} = (\pi_1, ..., \pi_T)$ is the output sequences of the target label $\pi_t \in \{1, ..., K\} \cup \{\phi\}$ and the posterior probabilities $p\left(\pi_t|\mathbf{x}_t\right)$ are modeled with a recurrent neural network $N_w : \mathbb{R}^{m \times T} \mapsto \mathbb{R}^{n \times T}$ such as LSTM which maps an input acoustic sequence $\mathbf{X}$ into a m-dimensional continuous value.

The CTC loss and its gradient with respect to the network parameters are efficiently computed with the forward-backward algorithm. Usually CTC-based model learns a monotonic alignment. It is advantageous for speech recognition because the output label sequence is monotonic in speech recognition. However, they do not explicitly learn the internal relationship between different time frames since they assume that the probability of each label is independent of others as in equation (2.5).

The CTC loss is defined based on the minimum log-likelihood criterion.

$$L_{CTC}(\mathbf{X}, \mathbf{Y}_{\text{ASR}}) = -\log P(\mathbf{Y}_{\text{ASR}}|\mathbf{X}) \tag{2.6}$$

In inference, we remove all repeating labels and blank labels from the paths in $\Omega^{-1}(\boldsymbol{\pi}) = \mathbf{y}$. For example, we can recognize $\Omega^{-1}(\phi aa\phi\phi a\phi bb) = aab$.

The time indices of non-blank tokens in $\boldsymbol{\pi}$ are used as the reference token boundaries. When repeated non-blank labels exist, the leftmost index corresponding to the same non-blank token is used as a reference token boundary. For instance, given a CTC path $\boldsymbol{\pi} = (\phi aa\phi\phi a\phi bb)$ corresponding to a reference transcription "a a b", we convert it to $(\phi, a, \phi, \phi, a, \phi, b, \phi)$ and then extract the time indices of the non-blank tokens alignment $= (2, 5, 7)$. In this dissertation, the CTC loss was used in some of the Chapters, especially in Chapter 5.

2.1.4 Attention-Based Encoder-Decoder Model

Attention-based encoder-decoder (AED) models are sequence-to-sequence (seq2seq) modeling that can learn soft alignments between a variable-length input and a target sequence [8,38,39]. This architecture consists of two distinct sub-networks as in Firgure 2.4b: an encoder, which consists of multiple recurrent neural network (RNN) layers that map the acoustic feature sequence to a distributed representation of lengths T, and a decoder, which consists of one or more RNN layers that predict the output sub-word sequence of length I. The length I is usually shorter than the input length T.

Generally, the decoder is tightly connected with the encoder output via an attention mechanism. An attention layer acts as the interface between the encoder and the decoder: it selects frames in the encoder representation $\mathbf{h}_{ASR} = (\mathbf{h}_1, ..., \mathbf{h}_T)$ that the decoder should attend in order to predict the next sub-word unit. In the decoder network, the hidden state activation of the RNN-based decoder at the i-th time step is computed as:

$$\mathbf{s}_i = \text{Recurrency}\left(\mathbf{s}_{i-1}, \mathbf{g}_i, y_i\right) \tag{2.7}$$

where $\mathbf{g}_i$ and y_{i-1} denote the "glimpse" at the i-th target label and the predicted symbol at the previous step. The glimpse $\mathbf{g}_i$ is a weighted sum of the encoder output sequence as:

$$\mathbf{g}_i = \sum_t \alpha_{i,t}\mathbf{h}_t \tag{2.8}$$

where $\alpha_{i,t}$ is an attention weight of $\mathbf{h}_t$. In this work, we use a content-based attention mechanism formulated as follows:

$$\begin{aligned} e_{i,t} &= \mathbf{w}^T \text{tanh}(\mathbf{W}\mathbf{s}_{i-1} + \mathbf{V}\mathbf{h}_t + \mathbf{U}f_{i,t} + \mathbf{b}) && (2.9) \\ \mathbf{f}_i &= \mathbf{F} * \boldsymbol{\alpha}_{i-1} && (2.10) \\ \alpha_{i,t} &= \exp(e_{i,t}) / \sum_{t'=1}^{T} \exp(e_{i,t'}) && (2.11) \end{aligned}$$

where $*$ denotes a 1-dimensional convolution. Using $\mathbf{g}_i$ and $\mathbf{s}_{i-1}$, the decoder predicts the next symbol $\mathbf{y}_i$ as:

$$\mathbf{y}_i \sim \text{Generate}\left(\mathbf{s}_{i-1}, \mathbf{g}_i\right) \tag{2.12}$$

where the Generate function is implemented as:

$$\mathbf{R}\tanh\left(\mathbf{P}\mathbf{s}_{i-1}+\mathbf{Q}\mathbf{g}_{i}\right) \tag{2.13}$$

The objective function for training the attention models is cross entropy. The loss is calculated using negative log-likelihood between the predicted symbol sequences and the target oracle label sequences.

$$\begin{aligned}\mathcal{L}_{AED} &= -\log P_{AED}(\mathbf{Y}_{\text{ASR}}|\mathbf{X}) &(2.14)\\ &= \sum_{i=1}^{I}\log P_{AED}(\mathbf{y}_{\text{i}}|\mathbf{Y},\mathbf{X}) &(2.15)\end{aligned}$$

2.1.5 Joint CTC/Attention training

When training the AED model, we use the cross entropy between the ground-truth labels and the predicted labels ($\mathcal{L}_{AED}$). In the ASR task, the attention between the acoustic features and the target label has monotonicity (left-to-right), but the structure of attention itself does not have the constraint, which sometimes causes the label repetition. The monotonic constraint of a CTC loss complements AED models to encourage monotonicity in the input-output alignment [40–42]. Therefore, the CTC loss L_{CTC} is typically used as an auxiliary regularization by sharing the encoder sub-network. To enhance the monotonicity, The total objective function L_{total} of multi-task learning with CTC loss is defined as:

$$\mathcal{L}_{ASR} = (1-\lambda_{ctc})\mathcal{L}_{AED} + \lambda_{ctc}\mathcal{L}_{CTC} \tag{2.16}$$

where λ_{ctc} is a tunable hyperparameter ($0 \leq \lambda_{ctc} \leq 1$) for the CTC loss weight.

2.1.6 Transformer

Transformer [9] is an end-to-end model that is relying entirely on self-attention without using RNNs. It was initially proposed for machine translation, and later it is shown to be also effective in speech processing tasks [39,43–45].

The transformer model consists of distinct encoder and decoder sub-networks as in Figure 2.5. The encoder is stacking of multiple identical layers (originally,

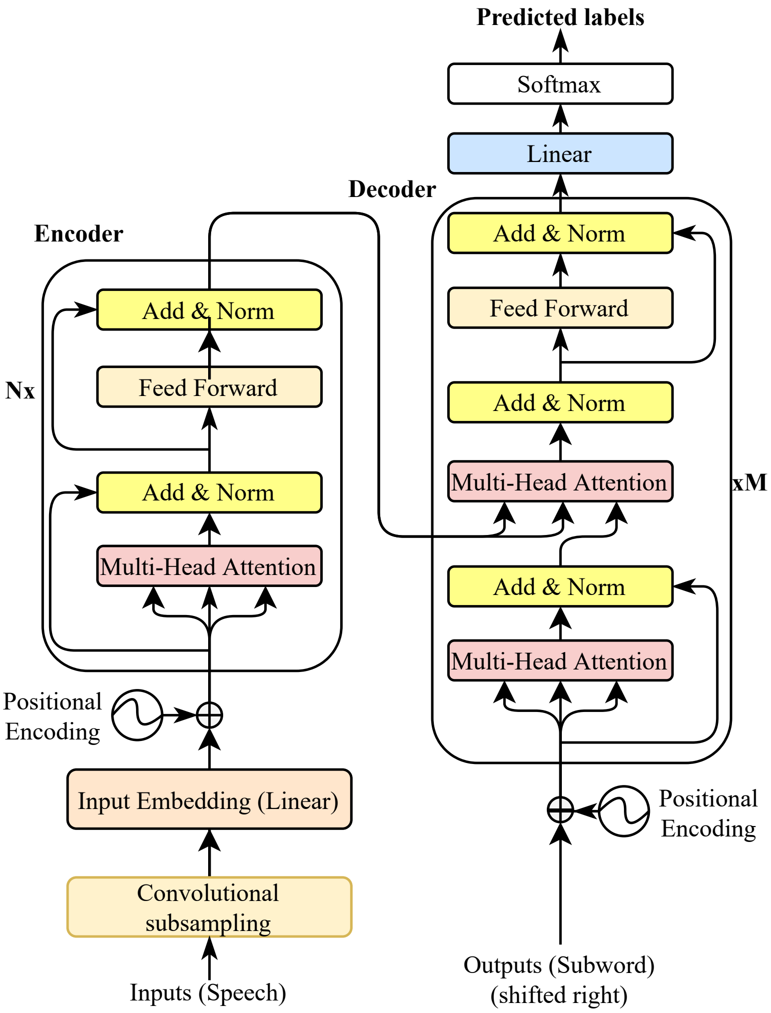


Figure 2.5: Transformer-based ASR model architecture

$N = 6$). Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. The decoder is also composed of a stack of multiple identical layers (originally, $M = 6$). In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack.

The multi-head attention (MHA) of the transformer is based on scaled dot-product attention. The scaled dot-product attention learns three weight matrices to calculate the attention; the $d_{q,k}$-dimensional query weights $W_Q \in \mathbb{R}^{d_{model} \times d_{q,k}}$, the $d_{q,k}$-dimensional key weights $W_K \in \mathbb{R}^{d_{model} \times d_{q,k}}$, and the d_v-dimensional value weights $W_V \in \mathbb{R}^{d_{model} \times d_v}$. We produce the query vector $Q = W_Q X_Q$, the key vector $K = W_K X_{K,V}$, and the query vector $K = W_V X_{K,V}$ using the input $X_{K,V}$ of the key and value, and X_Q of the query. The output of each Transformer layer is calculated using the multihead attention mechanism as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, ..., head_h)W^O, \tag{2.17}$$

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{2.18}$$

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.19}$$

where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_{q,k}}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_{q,k}}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $W^O \in \mathbb{R}^{hd_v \times d_{model}}$, h is the number of heads, and d_{model} is the model dimension, $\frac{1}{\sqrt{d_k}}$ is a scaled factor to alleviate the gradient vanishing problem of the softmax function. Note that in the multi-head attention, $d_{q,k} = d_v = d_{model}/h$. On each of these projected versions of queries, keys and values, the basic attention function is performed in parallel, yielding d_v-dimensional output values. These are concatenated and projected again, resulting in the final values.

In addition to attention sub-layers, each layer contains a fully connected network (feed-forward network, FFN), which is applied to each position separately and identically. This FFN module has several variants. FFN has two linear transformations with a ReLU activation in the original work. Thus, the output sequence of each encoder layer H_{enc}^n is given by:

$$\mathcal{A}_{enc}^n = \text{LayerNorm}(H_{enc}^{n-1}), \tag{2.20}$$

$$\mathcal{B}_{enc}^n = H_{enc}^{n-1} + \text{Multihead}(\mathcal{A}_{enc}^n, \mathcal{A}_{enc}^n, \mathcal{A}_{enc}^n), \tag{2.21}$$

$$\mathcal{C}_{enc}^n = \text{LayerNorm}(\mathcal{B}_{enc}^n), \tag{2.22}$$

$$H_{enc}^n = \mathcal{B}_{enc}^n + FFN(\mathcal{C}_{enc}^n), \tag{2.23}$$

$$\text{where FFN}(x) = \text{ReLU}(W_1 x + b_1)W_2 + b_2, \tag{2.24}$$

where $H_{enc}^0 = H + P$, using a sinusoidal positional encoding P.

On the other hand, the output of each decoder layer is calculated using both outputs of the previous decoder layer Z_{dec}^{m-1} and the encoder output H_{enc}^N. Note that we define $Z^0_{dec} = \text{Embedding}(Y) + P$.

$$\mathcal{A}^m_{dec} = \text{LayerNorm}(Z^{m-1}_{dec}), \tag{2.25}$$

$$\mathcal{B}^m_{dec} = Z^{m-1}_{dec} + \text{Multihead}(\mathcal{A}^m_{dec}, \mathcal{A}^m_{dec}, \mathcal{A}^m_{dec}), \tag{2.26}$$

$$\mathcal{C}^m_{dec} = \text{LayerNorm}(\mathcal{B}^m_{dec}), \tag{2.27}$$

$$\mathcal{D}^m_{dec} = \mathcal{B}^m_{dec} + \text{Multihead}(\mathcal{C}^m_{dec}, H^N_{enc}, H^N_{enc}), \tag{2.28}$$

$$\mathcal{E}^m_{dec} = \text{LayerNorm}(\mathcal{D}^m_{dec}), \tag{2.29}$$

$$Z^m_{dec} = \mathcal{D}^m_{dec} + FFN(\mathcal{E}^m_{dec}), \tag{2.30}$$

$$\text{where FFN}(x) = \text{ReLU}(W_1 x + b_1)W_2 + b_2, \tag{2.31}$$

We perform label prediction at each decoding step using the output of the final decoder layer Z^M_{dec} as:

$$\hat{Y} = \text{Softmax}(\text{Linear}(Z^M_{dec})), \tag{2.32}$$

The Transformer has a lot of benefits compared to the encoder-decoder model, which is composed of RNNs. First, the MHA can leverage parallelization and computational complexity. Secondly, it solves the problem of long-range dependencies using self-attention. Thus, the Transformer architecture is mainly used for most of this dissertation.

2.2 Translation Tasks

The translation is a task to transform a text or speech in a source language into text in other target languages. The former is called machine translation (MT), and the latter is called speech translation (ST).

2.2.1 Machine Translation (MT)

Machine translation is the task of automatically translating text from a source text of one language $\mathbf{Y_{src}}$ to the target text of one or more languages $\mathbf{\hat{Y}_{tgt}}$. Modern machine translation goes beyond simple word-to-word translation to communicate the full meaning of the original language text in the target language.

It analyzes all text elements and recognizes how the words influence one another. This task can be formulated to find the most plausible translation as

$$\hat{Y}_{tgt} = \underset{\mathbf{Y_{tgt}}}{\text{argmax}} P_{MT}(Y_{tgt}|Y_{src}), \tag{2.33}$$

where $\mathbf{P_{MT}}$ is the output probability of the MT system. The model training is conducted using a pair of ground-truth source and target text sequences.

Machine Translation (MT) has evolved through three major paradigms thus far. Namely, they are Rule-Based Machine Translation (RBMT), Statistical Machine Translation (SMT), and finally Neural Machine Translation (NMT).

RBMT was the first organized attempt to use computers for a translation task around the 1950s. It involves the creation of a bilingual dictionary and a set of grammar rules for each language to refer to during translation. In practice, RBMT was underwhelming, failing to produce fluent translations. Also, the initial cost in terms of funding and time to create these systems was very large.

SMT was pioneered by IBM in the 1990s [46]. It involves the statistical analysis of parallel corpora to derive an approximated translation model. It has been rather successful and was the dominant approach in MT until the emergence of NMT in the last few years. Due to the inclusion of a monolingual language model which quantifies the likelihood of a translation, SMT produced more fluent translations than RBMT. Moreover, it did not require complicated linguistic rules which were expensive and time-consuming. However, it required lots of manual feature engineering to create representative statistical models.

NMT is a radically different approach to solving the problem of machine translation that uses deep neural networks and artificial intelligence to train neural models [47–49]. Unlike the conventional approach of SMT, which consists of many small sub-components that are tuned separately, the NMT approach jointly trains all parts of the NMT model in a single network (end-to-end) to maximize the translation performance. In which a large neural network reads a sentence and outputs a correct translation. NMT has quickly become the dominant approach to machine translation with a major transition from SMT to NMT in just a few years. NMT typically produces much higher quality

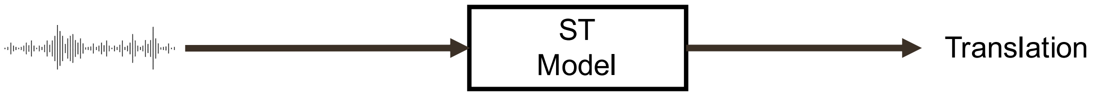


Figure 2.6: The flow of ST system

translations than SMT approaches, with better fluency and adequacy.

The translation difficulties vary significantly depending on the text style. For example, translation in news domains is conducted in a written form, while translation in conversational domains is conducted in a spoken form. The latter could include paralinguistic information, which is not included in the text. This dissertation mainly applies the Transformer-based architecture of NMT using the conversation domains in the court.

2.2.2 Speech Translation (ST)

Speech translation (ST) is a task to transform speech in a source language to text in one or more target languages. In traditional approaches, the overall ST task is decomposed into a chain of ASR and MT sub-tasks as

$$\begin{aligned}
\hat{Y}_{tgt} &= \underset{\mathbf{Y_{tgt}}}{\text{argmax}} P_{ST}(Y_{tgt}|X), && (2.34)\\
&\approx \underset{\mathbf{Y_{tgt}}}{\text{argmax}} \sum_{Y_{src}} P_{MT}(Y_{tgt}|Y_{src}) P_{ASR}(Y_{src}|X), && (2.35)\\
&\approx \underset{\mathbf{Y_{tgt}}}{\text{argmax}} P_{MT}(Y_{tgt}|\hat{Y}_{src}), && (2.36)
\end{aligned}$$

where $\mathbf{P_{ST}}$ is an output probability of the ST system, and $\mathbf{\hat{Y}_{tgt}}$ is the most plausible target translation. Because it is intractable to find $\mathbf{\hat{Y}_{tgt}}$ in the whole search space in both source and target languages, the most plausible ASR output $\mathbf{\hat{Y}_{src}}$ is generally used as an input to the MT module by introducing a heuristic search method, e.g., beam search.

Speech translation (ST) systems can be categorized into loosely-coupled cascade approaches and tightly integrated end-to-end(E2E) approaches according to the treatment of the ASR sub-task.

A cascade ST system is a loosely coupled model that decomposes the overall ST task into multiple sub-tasks, each of which is relatively easy to deal with [50]. It consists of separate ASR, MT, and optional text normalization sub-modules. While it is not easy to collect ST corpora having a triplet of (source speech, transcript, translation), the modularity in the cascade approach enables training each sub-module with separate corpora. This is one of the most important advantages over the E2E approach and the reason why cascade systems outperform E2E systems when the amount of training data is unrestricted. However, the cascade system typically suffers from error propagation from the ASR system because there exist mismatches between the ASR output and the MT output. The first mismatch is due to ASR errors such as misspelling by homophones because the ASR system is not perfect. The second mismatch is due to (1) a loss of punctuation and case information and (2) the existence of noisy classes such as disfluencies and sound events (e.g., applause, laughter, music, etc.). The third mismatch is due to the text style. ASR systems are trained in a spoken domain such as lectures and conversations while MT systems are trained in a written domain such as news.

To overcome the error propagation problem in the cascade approach, end-to-end ST (E2E-ST) has been intensively studied recently [51–53]. E2E-ST systems are typically implemented with E2E models such as Transformer. The main advantages of E2E modeling can be summarized as follows. (1) mitigation of error propagation from ASR systems, (2) low-latency inference, and (3) endangered language documentation.

E2E-ST is important especially when the source language does not have orthography. It is difficult to build ASR systems for such a language. However, collecting the training data for the direct ST task is more difficult than that for ASR and MT tasks. Moreover, E2E training is more complex than ASR and MT sub-tasks and thus makes optimization difficult. So there was a large gap in translation performance between cascade and E2E approaches.

There were techniques to improve the performance of E2E models. Firstly, the MTL with the use of ASR and MT auxiliary sub-tasks [52–54], in which the speech encoder is the ASR sub-task and the text decoder is the MT sub-task.

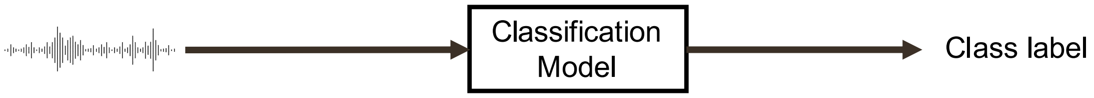


Figure 2.7: The flow of ASR system

However, an additional ASR decoder, a CTC layer, or both are stacked on the shared speech encoder, while an additional text encoder for source transcripts is also added. The translation decoder is shared for both E2E-ST and MT sub-tasks. The total training objective $\mathcal{L}_{\text{total}}$ can be formulated as a linear interpolation of an E2E-ST loss $\mathcal{L}_{\text{ST}}$, an ASR loss $\mathcal{L}_{\text{ASR}}$, and an MT loss $\mathcal{L}_{\text{MT}}$ as follows

$$\mathcal{L}_{\text{total}} = (1 - \lambda_{\text{ASR}} - \lambda_{\text{MT}})\mathcal{L}_{\text{ST}}(Y_{\text{tgt}}|X) + \lambda_{\text{ASR}}\mathcal{L}_{\text{ASR}}(Y_{\text{src}}|X) + \lambda_{\text{MT}}\mathcal{L}_{\text{MT}}(Y_{\text{tgt}}|Y_{\text{src}}), \tag{2.37}$$

where $\lambda_{ASR}(0 \leq \lambda_{ASR} < 1)$ and $\lambda_{MT}(0 \leq \lambda_{MT} < 1)$ are ASR and MT loss weights, respectively.

The second method is using pretraining models for ASR and MT tasks, in which the parameters of the E2E-ST encoder can be initialized with those of the pretrained ASR encoder, and the parameters of the E2E-ST decoder can be initialized with those of the pretrained MT decoder. Using a pretrained model can also encourage the model to converge faster than training it from scratch and reduce the overall training time. In this case, pretraining is more data-efficient than multi-task learning.

In this dissertation, we applied the second method, which uses the ASR encoder and MT decoder to initialize the encoder and decoder of ST, respectively.

2.3 Speech Classification

Speech classification is the task of automatically assigning a label or class to a given utterance or audio segment as in Figure 2.7. It can be used for recognizing which command a user is giving or the emotion of a statement, as well as identifying a speaker, language identification, and so on.

Speech classification task generally requires less capacity with the lower complexity of the tasks compared to the structured prediction problems of speech recognition and speech translation.

2.3.1 Speaker Identification

Speaker identification is classifying the audio of the person speaking in the speech signal. A set of speakers are usually predefined. The result is usually the decision for a certain speaker or rejection in the case of an unknown speaker.

A set of speech signals for each of the speakers to be identified is needed for training, which means that the system usually can recognize only the seen speakers in training. This algorithm is also language-independent, thus the speaker will be identified irrespective of the language used.

In the ASR system, speaker identification can be used to enhance the recognition performance with speaker embedding [55], which means that providing the speaker information is helpful for recognizing the transcription of the speech signal.

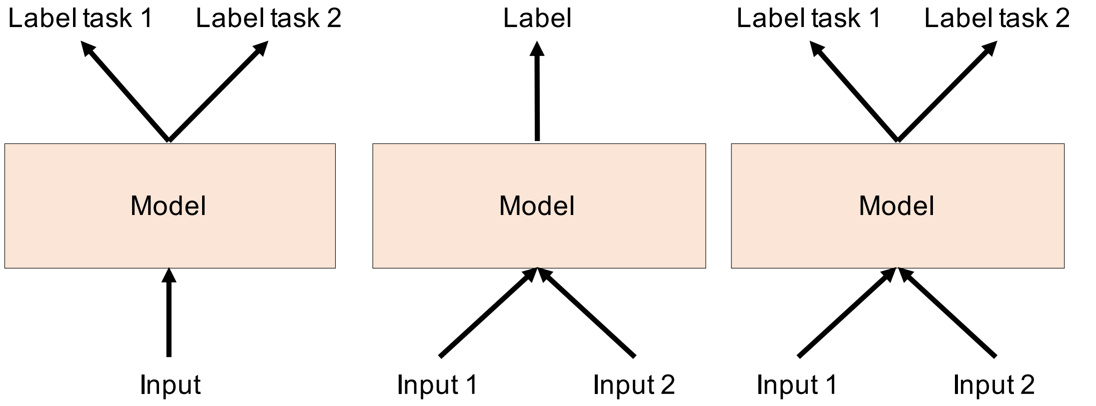
2.3.2 Language Identification

Spoken language identification (Lang ID) also known as language recognition, is the task of recognizing the language of the spoken utterance automatically. It typically serves as the prepossessing of ASR, determining which ASR model will be activated according to the language.

In the ASR system, language identification can be used to enhance the recognition performance in multilingual ASR. The commonly used method is to recognize the language first, and then recognize the transcription of the speech signal.

2.4 Multitask Learning in ASR System

Multi-task learning (MTL) was initially introduced in 1997 [56], and the early versions of MTL were called hint [57]. The idea is to train a neural network jointly for several different tasks but related to each other. In that, the network



a: Multiple outputs b: Multiple inputs c: Multiple inputs and outputs

Figure 2.8: Overview of multi-task learning architectures

learns one task together with one or more auxiliary tasks. The auxiliary tasks aim at helping the model to converge faster and better which can benefit the main task. At its core, MTL is an approach to parameter estimation for statistical models. Even though we use multiple tasks during training, we will produce only one model, which updates the model parameters of the multiple tasks in parallel by backpropagating the error through the hidden layers of the network. All MTL systems generally share two common characteristics: (1) all tasks are trained on the same input or output features, and (2) all tasks share the same internal representation. In MTL, each task contributes to the loss function with a definition as:

$$\epsilon_{\text{MTL}} = \epsilon_{\text{Main}} + \sum_{n=1}^{N} \alpha_n * \epsilon_{\text{Aux}_n}, \tag{2.38}$$

where ϵ_{MTL} is the cost function to be optimized, α_n is a nonnegative weight and N is the total number of auxiliary tasks. If the α_n is close to 1, it means that the auxiliary task will be as impacting as the main task, whereas α_n is close to 0, the auxiliary tasks are less influence on training.

Regarding neural approaches as illustrated in Figure 2.8, multi-task models are usually comprised of three architectures: (1) one-to-many; (2) many-to-one; (3) many-to-many. MTL models will always have some hidden layers shared among tasks entirely or partially.

With regards to domains in which we have very limited data (i.e. low-resource environments), multi-task parameter estimation promises gains in performance. In the common scenario where the engineers have access to only a small dataset, the best way we could improve performance would be by collecting more data. However, data collection is time-consuming and costly. Thus, creating new tasks is the promise of MTL in low-resource domains, in which we do not need to collect more data.

For ASR, MTL has found its way into other speech technologies including speech synthesis, speech emotion recognition, speaker identification, and language identification. In this dissertation, we mainly use MTL to enhance the ASR performance of the low-resource language in low-resource settings.

Chapter 3

Trilingual Corpus for Speech Recognition and Translation Studies

There are several spontaneous speech translation corpora available in a single source language, such as Must-C [33], Fisher-CallHome Spanish [58], and in multiple source languages, including Europarl-ST [59] and Multilingual TEDx [60]. Among them, only Europarl-ST is simultaneous speech translation. However, they only have a few languages available such as Western languages, mostly rich-resource languages. Moreover, the common problems of end-to-end speech processing systems are the requirement for a huge amount of parallel resources of speech and text, which is currently available for a limited number of language pairs. Annotating speech-text datasets process is usually time-consuming and very expensive, that is infeasible to rely on only manual data creation in the long run. Thus, innovative data collection methodologies are required for low-resource languages.

In this Chapter, we address two main challenges for constructing the speech-text pair corpus for speech transcription and translation studies: text sentence alignment and speech-text alignment. We extract the parallel speech of approximately 200 hours of raw audio from ECCC and its corresponding documents in Khmer, English, and French. This corpus will be not only usable for a pure ASR, MT, and ST, but also for a wide range of advanced tasks including multilingual ASR, MT, ST, cross-lingual, multi-source translation [61–67], or joint training [68,69].[1]

Sentence alignment of the source and target language is crucial in spoken language translation (SLT) corpus creation. Better language processing tools are required to improve quality alignment and time efficiency. However, this assumption does not hold for most low-resource languages, which usually have worse performance or lack of toolkit to support those languages. Additionally, the written style of Khmer occasionally uses spaces only to make the text more natural for reading; however, there are no sentence boundaries or punctuation marks to separate the text sentences. To overcome these challenging characteristics, we propose aligning the bilingual sentences in a monotonic process that only requires sentence segmentation of the source-language text. In contrast, only word tokenization is needed for the target-language text. This is suitable for a simultaneous translation dataset like the ECCC or Europarl. Secondly, we apply the Recognizer Output Voting Error Reduction (ROVER) method [70], a voting mechanism of multiple automatic speech recognition outputs, to improve the quality of the bilingual sentence alignment to Khmer by voting the alignment outputs of English-Khmer and French-Khmer.

Another challenge is text-to-speech alignment. Most other corpora have timestamp information for the audio data, but it is unavailable for the original ECCC dataset. Therefore, we generate timestamps for the speech data that corresponds to each sentence of the text. Ultimately, we created a large parallel TriECCC, which respectively has about 146, 148, and 125 hours in length of speech in Khmer, English, and French, approximately 62K utterances in each language pair of six directions. In this corpus, 60% of speech is the original speech of Khmer speakers, 18% of speech is the original speech of English speakers, and 22% of speech is the original speech of French speakers. Moreover, there is a wide range of speakers, including witnesses, defendants, lawyers, judges, and officers.

We first evaluate the baseline model of ASR, MT, and both cascaded-ST and E2E-ST on Transformer architecture [9] using the TriECCC. Among them, Khmer

[1] The data copyright belongs to NICT, Kyoto Univ. speech lab. and CADT, formerly NIPTICT.

language systems show worse performance than the other language pairs. In this work, we focus on improving the Khmer MT from/to English and French. We first investigate the system combination using the ROVER method for combining MT outputs. We then finetune the MTs of Khmer language pairs using the pretrained models of English-French MTs to initialize the encoder or decoder of each Khmer MT model. Experimental results show that the finetuning process improves the BLEU scores on Khmer-to-French, French-to-Khmer, Khmer-to-English, and English-to-Khmer MT systems.

3.1 TriECCC Corpus

3.1.1 Khmer Language

Khmer or Cambodian is the official language of Cambodia. Around 90% of the Cambodian population speak this language in Cambodia, and some speakers live in other countries. Khmer language (Cambodian) is one of the under-resourced Southeast Asian languages for natural language processing (NLP). It has an SVO (Subject, Verb, and Object) syntax structure. Syntactically it is pretty similar to Chinese and English, and also it is similar to Japanese, Chinese, and Myanmar in word composition. Each Khmer word is composed of single or multiple syllables, usually not separated by white spaces. Although spaces are used for separating phrases for easy reading, it is not strictly necessary. In addition, these spaces are rarely used in short sentences, and there is no exact rule on how they are used. There are three main word groups in modern Khmer: (1) original Khmer words, (2) Sanskrit and Pali, which have been influenced by the royal and religious registers, through Hinduism and Buddhism, and (3) loanwords from French and English, i.e., many words were borrowed and have become a part of the colloquial language, as well as medical and technical terms. There is also a smattering of Chinese and neighboring countries' loanwords in colloquial speech. Unlike Thai, Vietnamese, and Lao, Khmer is non-tonal. And it has a high percentage of disyllabic words which are derived from monosyllabic bases by prefixation and suffixation [71].

3.1.2 ECCC Background

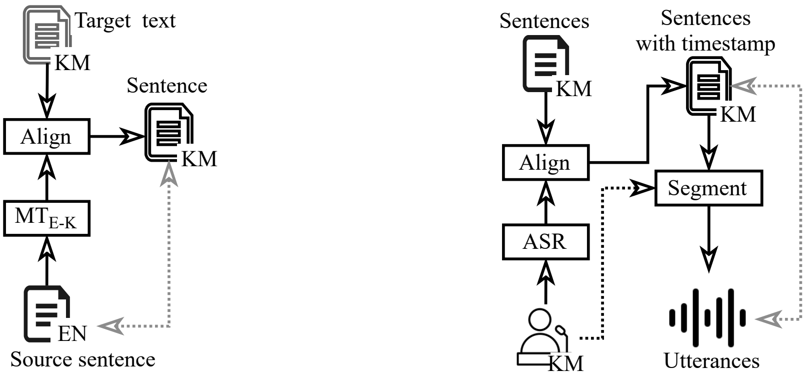
The ECCC is a court established to prosecute the senior leaders who committed crimes during the Khmer Rouge regime in Cambodia from 1975 to 1979, known as Democratic Kampuchea. The trials have been subsequently divided into four cases that began on February 17, 2009. These trials are still in progress, and only a small part has been released to the public. Therefore, we chose only the first case, which spanned from February 17 to November 27, 2009, as the resources of that case are available.

The trial had two kinds of hearing: public and non-public. Each hearing was simultaneously conducted in three languages: Khmer, English, and French. This means that the videos were recorded in the courtroom in the language of the main speaker. Concurrently, the human translators translated that speech to the other two languages. Each video, therefore, has three different languages. Thus far, the recordings have been carefully transcribed by native transcribers. Each transcription covers a single day of the trial, which corresponds to four or five audio sessions. Each recording session has a length of 5 to 150 minutes and involves a wide range of speakers: witnesses, defendants, judges, clerks or officers, co-prosecutors, experts, defense counsels, civil parties, and interpreters. As a result, we have collected 222 recording sessions that correspond to 60 documents. Each transcription has many pages in A4 size, ranging from 5 to 200. Finally, the public hearing videos were uploaded to a YouTube channel[2], and the proceedings are published in a digital format at the ECCC's official website[3].

The ECCC dataset has been built as a bilingual Khmer-English corpus for MT, which has only text data [72]. In this work, we focus on building a trilingual SLT corpus of Khmer, English, and French, which has six SLT directions. Moreover, we will also consider building a monolingual speech-to-text systems of Khmer.

[2] https://www.youtube.com/user/krtribunal/

[3] https://www.eccc.gov.kh/



a: Sentence alignment

b: Text-speech alignment

Figure 3.1: The process of creating the ECCC corpus: a: bilingual sentence alignment, b: text-to-speech alignment and segmentation

3.1.3 Corpus Creation and Key Statistics

The raw resources presented in Section 3.1.2 are useful for ASR, MT, and ST systems. However, it is not possible to directly use them for those tasks, particularly because this dataset lacks timestamps. The sentence alignment is a critical component of corpus creation. As English has better language processing tools, we used it as the source language for the alignment purpose.

Source to target sentence alignment

To align sentences, sentence segmentation is required in both source and target text, as presented in [73–75]. In these works, sentences were aligned based on the alignment score of each sentence. With this scoring, the alignment can be in the form of zero-to-one, one-to-zero, one-to-one, one-to-many, many-to-one, and many-to-many. However, only one-to-one is usable in the translation task. Thus, many of the original resources can be removed. Some languages such as Khmer, however, do not have any sentence tokenization tools such as Moses [13] and Punkt [14]. On the other hand, the simultaneous translation is processed in a monotonic and continuous alignment. With this characteristic, only the source language requires sentence segmentation.

We followed Fig. 3.1a to align the bilingual source and target texts. We first conducted sentence segmentation of English using Moses. The sentences were

Table 3.1: CER in source to target sentence alignment

Sentence alignment	CER (%)
Bilingual English (EN)-to-Khmer (KM)	13.2
ROVER ({EN, French (FR)}-to-KM)	**12.7**

re-split based on some conjunction words to ensure less than 200 characters (without spaces). We then translated those sentences to the target languages, Khmer and French, using the translation API in Google Sheets. For the ground truth of Khmer and French, we merged all text into a single line. However, the Khmer language is written without word boundaries. Thus, the Khmer word segmentation tool [76] was used to segment both the translated and ground-truth text.

Second, the alignment between translated and ground truth was conducted using dynamic programming (DP) in a monotonic manner. Sentence boundary tokens were inserted following the sentence boundaries of the translated text. In this alignment, the calculation was based on the word-level Levenshtein distance. As a result, only one-to-zero and one-to-one alignments are obtained. At this point, we removed the one-to-zero-aligned sentences from the source language.

Fig. 3.1a shows that the alignment requires the MT to translate from source to target language. The alignment between English-French is acceptable because of the high translation quality of English-French. However, the translation quality of English/French-to-Khmer is limited; thus, the alignment still needs improvement. To address this problem, we applied the ROVER method, which will be described in Subsection 3.3.1, to combine the aligned Khmer text of English-Khmer and French-Khmer translations. With this voting result, we improved the performance by 0.5% of character error rate (CER) as shown in Table 3.1 and the example is given in Fig. 3.2. As a result, we obtained 82, 078 sentences in English aligned with 78, 981 sentences in French and 80, 417 sentences in Khmer, which means that only 4% and 2% in French and Khmer were discarded, respectively as presented in the second column of Table 3.2.

Raw resource	[EN]: interrogators and the cadres from Prey Sar would be called to attend such a political session in general but there was another political session conducted separately. [KM] រួម ទាំង កង សួរ ចម្លើយ រួម ទាំង ខាង ព្រៃ ស គឺ ជា វគ្គ នយោបាយ រួម ទូទៅ ។ តែ ចំពោះ អ្នក សួរ ចម្លើយ គឺ នយោបាយ ខុស គ្នា នេះ ជា នយោបាយ ជំនាន់ នោះ ។
Reference	[EM]: 1. interrogators and the cadres from Prey Sar would be called to attend such a political session in general 2. but there was another political session conducted separately. [KM] 1. រួម ទាំង កង សួរ ចម្លើយ រួម ទាំង ខាង ព្រៃ ស គឺ ជា វគ្គ នយោបាយ រួម ទូទៅ ។ 2. តែ ចំពោះ អ្នក សួរ ចម្លើយ គឺ នយោបាយ ខុស គ្នា នេះ ជា នយោបាយ ជំនាន់ នោះ ។
Monolingual (EN-to-KM)	[KM] 1. រួម ទាំង កង សួរ ចម្លើយ រួម ទាំង ខាង ព្រៃ ស គឺ ជា វគ្គ នយោបាយ រួម 2. ទូទៅ ។ តែ ចំពោះ អ្នក សួរ ចម្លើយ គឺ នយោបាយ ខុស គ្នា នេះ ជា នយោបាយ ជំនាន់ នោះ ។
ROVER	[KM] 1. រួម ទាំង កង សួរ ចម្លើយ រួម ទាំង ខាង ព្រៃ ស គឺ ជា វគ្គ នយោបាយ រួម ទូទៅ ។ 2. តែ ចំពោះ អ្នក សួរ ចម្លើយ គឺ នយោបាយ ខុស គ្នា នេះ ជា នយោបាយ ជំនាន់ នោះ ។

Figure 3.2: Source to target sentence alignment examples

Table 3.2: Statistics of data reduction by the alignment based on the English sentences

Source	Text sentence	Speech utterance	Target language speech utterance	utterance
EN	**82,078**	$79,857(-3\%)$	KM: $78,063(-5\%)$	FR: $78,016(-5\%)$
FR	$78,981(-4\%)$	$75,616(-4\%)$	KM: $73,967(-6\%)$	EN: $75,461(-4\%)$
KM	$80,417(-2\%)$	$65,679(-18\%)$	EN: $65,391(-19\%)$	FR: $64,203(-20\%)$

Text to speech alignment

Fig. 3.1b shows the process of the text to speech alignment. We first trained an acoustic model that supported Vosk[4] using the Basic Expressions Travel Corpus [77] that was used in [28]. Vosk enables us to diarize the speech to generate the transcription with its corresponding timestamp.

Then, we conducted sentence alignment between the segmented sentences and the pseudo labels of ASR diarization output. The starting and ending timestamps of each sentence are aligned with a short audio data segment. At this stage, the alignment algorithm in Subsection 3.1.3 was used to generate the ground-truth text with the corresponding timestamp.

[4] https://alphacephei.com/vosk/

In this step, the performance of the ASR system affects the alignment result, which means that better ASR performance will generate better alignment output. In this case, the text-to-speech alignment of English and French is well done. As presented in the third column of Table 3.2, it reduced only 3% and 4% of English and French utterances, respectively. However, it reduced about 18% Khmer utterances by the text-to-speech alignment. The reason for this large reduction is the Khmer ASR model performance [28] was insufficient for transcribing some parts of the speech in this dataset. This is related to the domain and speaking-style mismatch, as the model was trained on the traveling domain and reading style.

Data cleaning

For a usable corpus, we first cleaned the text data. We focused on the transcribed text that corresponds to speech data using the following process: removing unrelated parts that do not correspond to speech such as page headings, descriptions of the activity, and feelings that are usually marked by “[ ]". For English and French, the text normalization was conducted using Moses. Subsequently, the punctuation marks were removed and the text was changed to lowercase. For Khmer, we deleted the non-standard characters, punctuation marks, and other Latin symbols. We also normalized the text by correcting the spelling and following the order of the Khmer characters or diacritics, as presented in [78]. The numbers and abbreviations were also replaced by their standard spoken equivalent in all languages.

Second, we cleaned the speech corpus to ensure that the length of each audio segment was usable in ASR or ST. A usable length is in a range from $3s$ to $30s$. Each sentence of the transcription had to be less than 300 characters in length because each source sentence in English was limited to less than 200 characters before alignment. Sentences and audio segments that did not meet these criteria were deleted from the corpus.

With the cleaning process, only a small portion $(1-2\%)$ of the original segmented speech utterances in the third column was reduced to the fourth

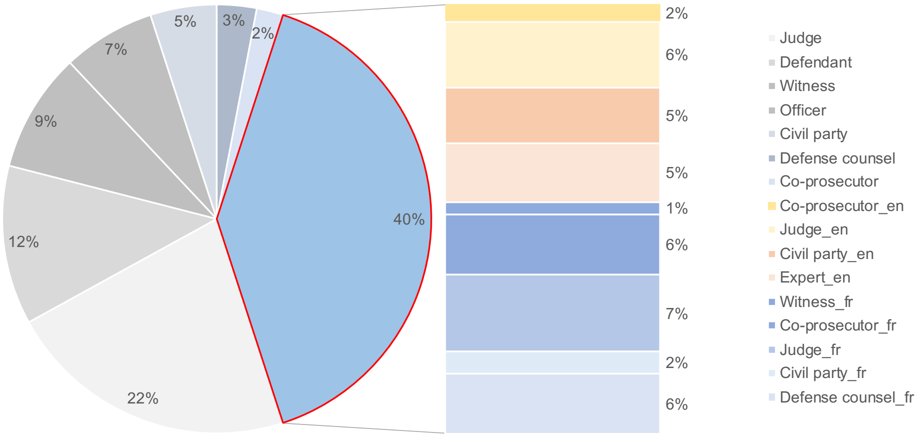


Figure 3.3: Speaker distribution in a trilingual SLT corpus of the ECCC

and fifth columns of Table 3.2. There are two main reasons for this reduction of utterances: i) mismatch between source speech and target text, and ii) long speech utterances which were not transcribed and segmented in the process of Section 3.1.3.

Trilingual corpus statistics

The graph in Figure 3.3 shows the speaker distribution for each speaker group. Overall, 60%, 18%, and 22% of speech is the original speech of Khmer, English, and French speakers, respectively. For Khmer source speech, 60% of speech is the original speech of Khmer speakers, including judges, defendants, witnesses, officers, co-prosecutors, defense counsels, and civil parties. The largest percentage is the speech of the judges, which makes up 22% of the corpus, followed by 12% from the defendant, 9% from the witnesses, and 17% in total from other speakers. The remaining 40% of the speech is that of interpreters who interpreted the speech from native English and French speakers such as co-prosecutors, judges, civil parties, experts, witnesses, and defense counsels. For English source speech, 18% of speech is the original speech of English speakers, while the other 82% is the speech of interpreters who interpreted from the native of French and Khmer. Similarly, French has 22% of speech of native speakers, whereas another 78% is interpreted by English and Khmer native speakers.

Table 3.3: Statistics of each source language in the trilingual ECCC SLT corpus

Source	#utterances	#words	vocabulary	#hour (train/dev/test)
KM	62K	1.6M	9K	132/7/7
EN		1.2M	14K	134/7/7
FR		1.3M	20K	113/6/6

Table 3.2 gives only the statistics of bilingual SLT, which cannot be used in some tasks such as multi-source translation or parallel joint training. Thus we selected the subset of the trilingual SLT corpus as shown in Table 3.3. This table gives the statistics of the SLT corpus of six-direction between Khmer, English, and French languages. It is approximately 146, 148, and 125 hours of speech in Khmer, English, and French, respectively, about 62K utterances of six directions. In terms of text, it is approximately 1.6M, 1.2M, and 1.3 words and the vocabulary sizes are 9K, 14K, and 20K in Khmer, English, and French, respectively. Finally, each language pair was split into training, development, and test sets, which are used in all experiments in this work.

3.2 Baseline End-to-End ASR, MT, and ST Systems

The Transformer [9] is a recently state-of-the-art model applied in many fields, including applications of speech and language processing such as ASR, MT, and ST, also involved in this work. This architecture mainly stacks data input, encoder, decoder, and output building blocks. The data input building block uses embedding and position-encoding layers to transform an encoder's input sequences or features. On the other side, the output building block uses linear and softmax layers to generate the sequence of output tokens. Mainly, the encoder and decoder modules are core components that use the self-attention mechanism to calculate the attention score of each input sequence. Scaled dot product attention is then used to compute a weighted sum of values for a queries ($\mathbf{Q}$) matrix of the three inputs: $\mathbf{Q}$, keys ($\mathbf{K}$) and values ($\mathbf{V}$) as defined:

$$\mathrm{Attention}(Q, K, V) = \mathrm{Softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3.1}$$

The encoder module is comprised of stacking the multi-head attention (MHA) and fully connected feed-forward network, coupled with layer normalization and residual connection. The attention module splits its **Q**, **K**, and **V** parameters N-ways and passes each split independently through a separate head. And all heads will be then combined to produce a final attention score using a concatenation operation:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, .., \text{head}_\text{h})W^O, && (3.2) \\ \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), && (3.3) \end{aligned}$$

and the fully connected feed-forward network consists of two linear transformations with Rectified Linear Unit (ReLU) activation in between:

$$\begin{aligned} x &= \text{FeedForward}(x), && (3.4) \\ \text{FeedForward}(x) &= \max(0, xW_1 + b_1)W_2 + b_2, && (3.5) \end{aligned}$$

The decoder has similar architecture as the encoder, which stacks multi-head attention with feed-forward networks in each layer. However, there are two multi-head attention sub-layers: i) a decoder self-attention, in which each position attends to all previous positions including the current position, and ii) encoder-decoder attention, in which each position of the decoder attends to all positions in the last encoder layer.

Even though there has been a lot of interest recently in applying Transformer in speech and language processing to archive promising results in both quality and efficiency, the task is limited to languages with large enough resources. This means that so far in low-resource language, the performance is limited due to data scarcity, resource quality, domain variability, and so on. Additionally, speaker-variability, speaking styles, and audio-recorded environment [11] also affect the ASR performance. In contrast, the text style of written and spoken forms and text-speech mismatch and error propagation [79] generally influence to the MT and ST performance.

We conducted ASR, MT, and ST experiments using a Transformer-based [9] architecture implemented in ESPnet [80]. In all experiments, the network is comprised of six encoder layers and six decoder layers. The dimension of the

Table 3.4: Word error rate (WER) of the ASR models in Khmer, English, and French; '*' model is used in cascade-ST and E2E-ST

Transformer ASR Model	**WER**		
	KM	**EN**	**FR**
w/o augmentation	23.6	6.9	14.5
w/ speed perturbation (SP)	22.2	6.6	14.0
w/ SpecAugment (SA)	21.8	6.4	13.8
w/ SP + SA *	**21.4**	**6.0**	**12.6**

feed-forward network was set to $2,048$, and the dropout was set to 0.1. The model used 4-head self-attention of 256 dimensions. We trained each model on a single 12-GB GPU Titan X (Pascal) with the aforementioned configurations.

3.2.1 Automatic Speech Recognition (ASR)

In the ASR system, we trained the model using 80-dimensional log-melscale filterbank (lmfb) coefficients and 3-dimensional pitch features. This network was started with downsampling by a 2-layer time-axis convolutional layer with 256 channels, stride size 2, and kernel size 3. The model was jointly trained with connectionist temporal classification (CTC) (weight $\alpha = 0.3$) for 45 epochs with a batch size of 64. The Noam optimizer was used with 25K warmup steps and an initial learning rate of 5.

The transcription was stripped of all punctuation marks. We used 5K byte-pair encoding (BPE) tokens [81] as the vocabularies for each language. Speech perturbation [82] and SpecAugment [83] were applied as data augmentation. All system performances are evaluated in WER and shown in Table 4.2. The table shows that English ASR performs better compared to other languages. Its WER is 6.0% followed by French with 12.6% and the performance of the Khmer language is worst with 21.4%. The Khmer speech is the most challenging in this corpus because the original Khmer speech was spoken by the older people who were the victims of the Khmer Rouge regime. Most of them are illiterate in the Khmer language. They sometimes cannot pronounce words correctly, and exhibit disfluency and emotions in their speech during the trial [84]. On the other hand,

Table 3.5: BLEU for translation of each language pair in a TriECCC corpus

Source	**Target**	**BLEU**		
		MT	**Cascade-ST**	**E2E-ST**
KM →	EN	16.63	15.14	13.81
	FR	11.53	10.66	9.39
EN →	KM	14.44	14.15	14.14
	FR	25.01	24.32	20.83
FR →	KM	10.54	9.82	10.26
	EN	27.37	25.17	23.64

78% speech in English and 82% speech in French were spoken by middle-aged interpreters and other well-prepared speakers, including judges, co-prosecutors, and civil parties.

3.2.2 Machine Translation (MT)

For MT, we trained another Transformer-based model for 100 epochs with a batch size of 96. However, the model tends to converge within 50 epochs. The Noam optimizer was used with 8K warmup steps and an initial learning rate of 1. All punctuation marks were stripped and converted to lowercase English and French in each language pair. We applied 15K BPE tokens of trilingual vocabularies, which resulted in 5K per language. The translation performances are reported using BLEU [85], as shown in Table 3.5.

The translation between English and French performs much better than that between Khmer and English/French. This is because of the disfluency of Khmer transcription, which was transcribed from the disfluent speech of the original Khmer speakers. Moreover, the translations between Khmer and English perform better than between Khmer and French. This is reasonable because English was directly used as the source language for the bilingual sentence alignment to Khmer and French, which were indirectly aligned.

3.2.3 Speech Translation (ST)

The E2E-ST front-end configuration is similar to the ASR system. The speed perturbation and SpecAugment were applied as the speech data augmentation.

The 15K BPE tokens of trilingual vocabulary were used as they were for MT. Note that the trilingual vocabulary was used for all translation models because it is useful for transfer learning purposes on both ST and MT in this work. In ST systems, we trained only 60 epochs with a batch size of 64. The ASR and MT pretrained models, which were presented in the previous Sections 3.2.1 and 3.2.2, were respectively used to initialize the E2E-ST encoder and decoder. With this initialization, the E2E-ST can achieve reasonable performance, as described in [86]. For cascade-ST, we first transcribed the speech using the ASR system, and then this output text was fed to the MT model to translate into the target language. The results are reported in Table 3.5.

The table shows the performance of both E2E-ST and cascade-ST. Overall, the cascade-ST system has a slightly lower BLEU score compared to the MT system, but it is better than E2E-ST in most cases. Generally, the ST performance has a big problem in the non-monotonic alignment of speech-text or text-speech, which is why their performances were worse than the normal MT models. Moreover, the speech condition is also an influential factor on ST performance, for instance, the translation to Khmer by the E2E-ST system is comparable to or better than cascade-ST models. This is because the English and French ASR performances are better than the Khmer ASR performance.

3.3 Enhancement of MT

3.3.1 Methods

In order to enhance ASR and MT of low-resource languages, many approaches have been investigated including multilingual training, system combination, transfer learning, and knowledge distillation. In this work, we focus on improving the MT of the Khmer language from/to English and French by using a system combination of ROVER and cross-lingual transfer learning methods.

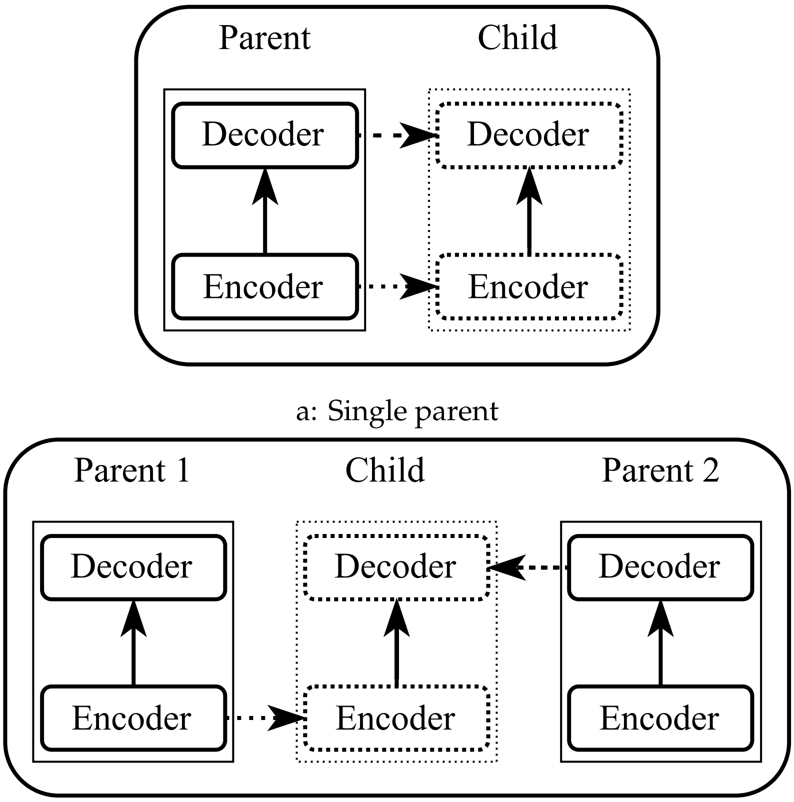
ROVER method

ROVER is one of the most commonly used methods to combine the hypotheses of multiple ASR outputs in the system combination. Originally, ROVER performs two-step procedures composed of word alignment and voting mechanisms. Word alignment combines the multiple outputs using dynamic programming to a minimal-cost word transition network (WTN). Then, the voting mechanism selects the best output word sequence based on the frequency of occurrence and word-level confidence score. This method has been shown to significantly reduce the WER [70]. However, the voting result will be poor if the confidence score of each output system is not reliable. Moreover, the voting result will not outperform the individual system if multiple systems do not have complementary errors [87].

In this work, we combine only two translation systems that produce different hypotheses of the same target language from different language source inputs of the same content. Specifically, we used the ROVER method to combine the translation output of English-to-Khmer and French-to-Khmer MT systems to enhance the hypothesis of the Khmer language.

Transfer learning

The transfer learning methods have been successfully applied to applications in speech and language processing, including speech recognition, document classification, and sentiment analysis [88], MT [89] and various downstream tasks [90, 91]. In the MT task, using the pretrained model of high-resource language pair (e.g., Spanish-to-English) is effective in assisting a low-resource language pair (e.g., Catalan-to-English) [89]. With this approach, a parent model is trained on a high-resource language pair, and then the trained parameters are used to initialize a child model, which is trained on the desired low-resource language pair. On the other hand, a multiple parents finetuning process [92], which has two parents to transfer to a child model in two steps, is beneficial when multiple languages are involved. For instance, to improve a child model (e.g., German-to-Czech), we can first use a parent (e.g., German-to-English) to



a: Single parent

b: multiple parents

Figure 3.4: The finetuning process: a: the single parent, b: the multiple parents

initialize to encoder parameters, and another parent (e.g., English-to-Czech) to initialize the parameters of the decoder of the child model. We can transfer some or all parameters from the parent to a child model at the initializing stage. However, the effectiveness of finetuning might be different when transfer learning is conducted in different parameters or layers, especially with a complex model with multiple modules such as Transformer.

In this work, we use English-to-French and French-to-English MT systems as the pretrained models because we aim to leverage the well-trained model of the high-resource language pairs. Moreover, as presented in Table 3.5, the English-French models show much better translation quality than Khmer from/to English and French.

We investigate initialization from both single and multiple parents as shown in Fig 3.4. We first investigate the use of a single parent (Fig 3.4a) to initialize the

Table 3.6: Result of ROVER method for MT outputs of Khmer

Source	Target	Baseline	ROVER
EN	KM	14.44	**14.79**
FR		10.54	

encoder, decoder, and both encoder and decoder modules (e.g., the English-to-French model is used to initialize the English-to-Khmer and Khmer-to-French models). Secondly, we conduct the initialization from multiple parents as in Fig 3.4b (e.g., the encoder part is finetuned from the English-to-French model, and the decoder part is finetuned from the French-to-English model or vice versa).

For the finetuning from the pretrained model, we used the same configuration of the original MT as presented in Subsection 3.2.2. However, we trained each model only 30 epochs, and the models were well converged. In each finetuning process, the initializing was applied to all layers or some of the specific layers in the encoder-decoder modules of the Transformer, but initializing to all layers of both the encoder and decoder shows the best performance.

3.3.2 Experimental Evaluations

Voting the Khmer translation using ROVER method

Table 3.6 shows that the ROVER method improved the translation to Khmer. Specifically, it outperforms the BLEU score of English-to-Khmer and French-to-Khmer systems by 0.35 and 4.25, respectively. This is because the ROVER method increases the variety of output by combining the two hypotheses.

Finetuning the Khmer translation using pretrained model

Table 3.7 presents the best practice of initializing with single and multiple parents by transferring the parameters to the encoder, decoder only, or both encoder and decoder modules. In the single-parent case, the English-to-French model is used to initialize the English-to-Khmer and Khmer-to-French models, while the French-to-English model is used to initialize the French-to-Khmer and Khmer-to-

Table 3.7: The best practice in the use of the pretrained model to initialize each Khmer MT model, **bold-font** module is transferred (Enc.-Dec.: Encoder-Decoder)

Source	**Target**	**pretrained parent models used for initialization**			
		Encoder	**Decoder**	**Enc.-Dec.**	**Multiple parents**
KM	EN	**FR**-EN	FR-**EN**	**FR-EN**	**EN**-FR and FR-**EN**
	FR	**EN**-FR	EN-**FR**	**EN-FR**	**FR**-EN and EN-**FR**
EN	KM	**EN**-FR	EN-**FR**	**EN-FR**	**EN**-FR and FR-**EN**
FR		**FR**-EN	FR-**EN**	**FR-EN**	**FR**-EN and EN-**FR**

Table 3.8: Comparison of the best performance of finetuning approach initializing the pretrained model into the encoder, decoder, or both on each Khmer MT model

Source	**Target**	**Performance of each initial option (BLEU↑)**				
		Baseline	**Encoder**	**Decoder**	**Enc.-Dec.**	**Multiple**
KM	EN	16.63	17.56	17.04	**18.16**	17.64
	FR	11.53	12.69	12.45	**13.77**	13.61
EN	KM	14.44	15.37	15.11	15.54	**15.61**
FR		10.54	11.81	11.32	**12.13**	11.85

English models. In the case of multiple parents, the English-to-French model is used for the encoder module and the French-to-English model is used for the decoder side or vice versa.

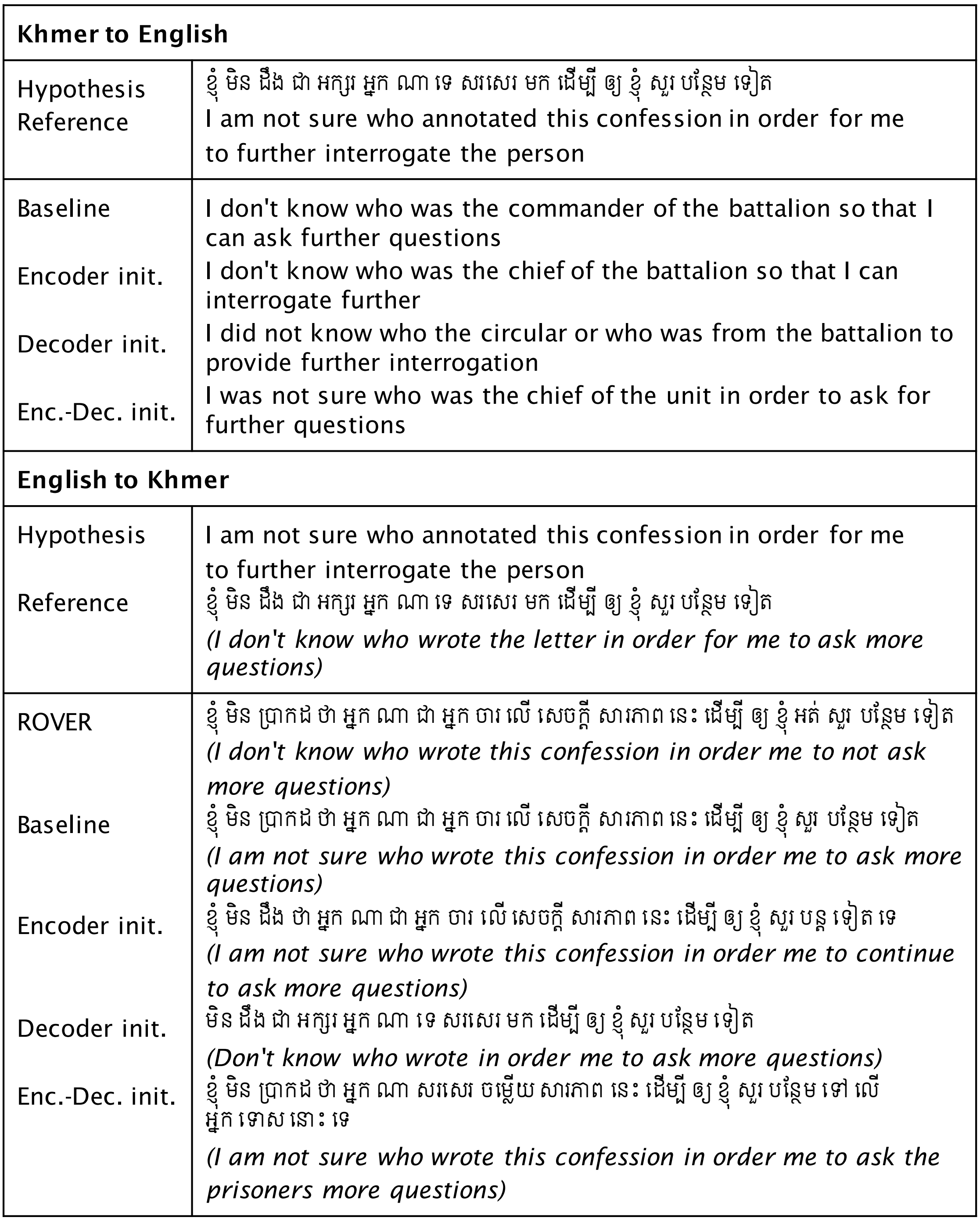
Table 3.8 compares the performance of the finetuning approach, which uses the pretrained model to initialize all layers of the encoder, decoder, or encoder-decoder modules, compared with the baseline performance. Generally, initializing with the pretrained model is effective for boosting the MT performance in both directions of Khmer MT systems. Transferring the knowledge to the encoder only is usually better than the decoder, but initializing both the encoder and decoder shows the best improvement in all systems. Additionally, the single-parent finetuning shows better performances in most models, except for the English-to-Khmer.

In terms of Khmer as a source language, using the pretrained model of the same target language gives the best performance. Specifically, the pretrained model of French-to-English improved the Khmer-to-English MT, whereas the English-to-French model improved the Khmer-to-French MT. This is because the pretrained model helps to generalize the alignment from Khmer to the target languages. Especially, the decoder part can be enhanced from the translation knowledge of the pretrained model in the same target language.

On the other hand, when Khmer is a target language, the French-to-English model was the best pretrained model to enhance the performance of the MT performance. In this case, the use of this pretrained model to initialize both the encoder and decoder increased the performance with the single-parent finetuning process. Whereas using the English-to-French pretrained model to initialize the encoder and initializing the decoder with the French-to-English model shows the best performance in the English-to-Khmer MT model. There are two main reasons for this improvement, i) using the same source language between the new and pretrained models is helpful in the alignment process, ii) the pretrained model of French-to-English has better performance than English-to-French.

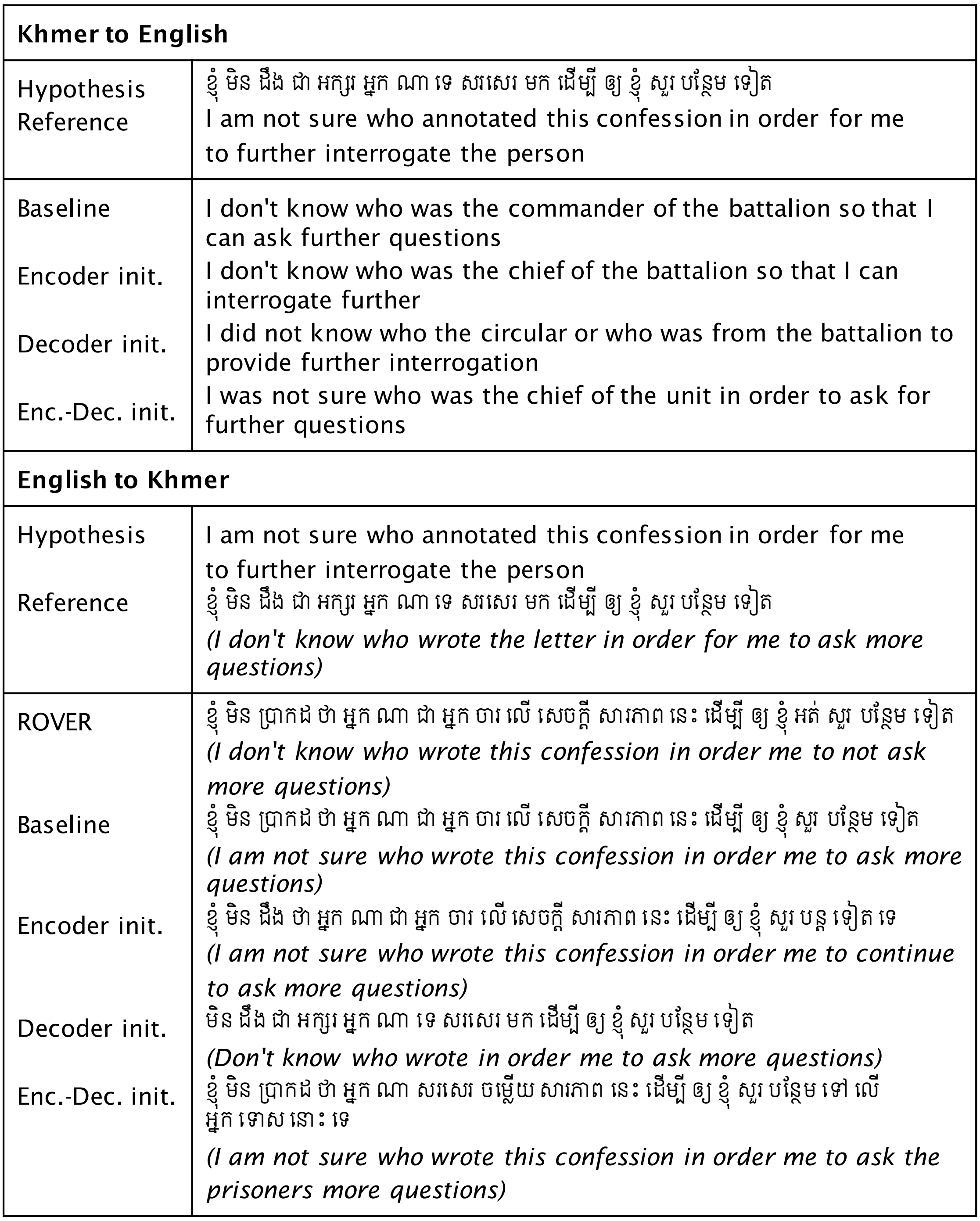
Overall, using the same source or target between the new and pretrained MT models can enhance the performance of low-resource MT systems because the knowledge of the pretrained model improved the alignment between the source and target languages. As a result, the finetuning process improved the BLEU score by 2.24 and 1.59 points for Khmer-to-French and French-to-Khmer, respectively. On the other hand, the translations between Khmer and English were improved by only 1.53 and 1.17 points for Khmer-to-English and English-to-Khmer, respectively.

Figure 3.5 and 3.6 show examples of the compared methods in the translated result between Khmer and other languages. The output of the transfer learning shows consistent improvement as it gives a complete sentence with the same meaning, whereas the ROVER method sometimes generated an incomplete sentence, or changed the meaning of the output sentence because this method copies different words from other system outputs.



Khmer to English	
Hypothesis Reference	ខ្ញុំ មិន ដឹង ជា អក្សរ អ្នក ណា ទេ សរសេរ មក ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ថែម ទៀត I am not sure who annotated this confession in order for me to further interrogate the person
Baseline	I don't know who was the commander of the battalion so that I can ask further questions
Encoder init.	I don't know who was the chief of the battalion so that I can interrogate further
Decoder init.	I did not know who the circular or who was from the battalion to provide further interrogation
Enc.-Dec. init.	I was not sure who was the chief of the unit in order to ask for further questions
English to Khmer	
Hypothesis	I am not sure who annotated this confession in order for me to further interrogate the person
Reference	ខ្ញុំ មិន ដឹង ជា អក្សរ អ្នក ណា ទេ សរសេរ មក ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ថែម ទៀត *(I don't know who wrote the letter in order for me to ask more questions)*
ROVER	ខ្ញុំ មិន ប្រាកដ ថា អ្នក ណា ជា អ្នក ចារ លើ សេចក្ដី សារភាព នេះ ដើម្បី ឲ្យ ខ្ញុំ អត់ សួរ បន្ថែម ទៀត *(I don't know who wrote this confession in order me to not ask more questions)*
Baseline	ខ្ញុំ មិន ប្រាកដ ថា អ្នក ណា ជា អ្នក ចារ លើ សេចក្ដី សារភាព នេះ ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ថែម ទៀត *(I am not sure who wrote this confession in order me to ask more questions)*
Encoder init.	ខ្ញុំ មិន ដឹង ថា អ្នក ណា ជា អ្នក ចារ លើ សេចក្ដី សារភាព នេះ ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ត ទៀត ទេ *(I am not sure who wrote this confession in order me to continue to ask more questions)*
Decoder init.	មិន ដឹង ជា អក្សរ អ្នក ណា ទេ សរសេរ មក ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ថែម ទៀត *(Don't know who wrote in order me to ask more questions)*
Enc.-Dec. init.	ខ្ញុំ មិន ប្រាកដ ថា អ្នក ណា សរសេរ ចម្លើយ សារភាព នេះ ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ថែម ទៅ លើ អ្នក ទោស នោះ ទេ *(I am not sure who wrote this confession in order me to ask the prisoners more questions)*

Figure 3.5: Examples of all methods in English-Khmer MT models, the *italic text* in the “()" is the translated text into English.



Khmer to English	
Hypothesis Reference	ខ្ញុំ មិន ដឹង ជា អក្សរ អ្នក ណា ទេ សរសេរ មក ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ថែម ទៀត I am not sure who annotated this confession in order for me to further interrogate the person
Baseline Encoder init. Decoder init. Enc.-Dec. init.	I don't know who was the commander of the battalion so that I can ask further questions I don't know who was the chief of the battalion so that I can interrogate further I did not know who the circular or who was from the battalion to provide further interrogation I was not sure who was the chief of the unit in order to ask for further questions
English to Khmer	
Hypothesis Reference	I am not sure who annotated this confession in order for me to further interrogate the person ខ្ញុំ មិន ដឹង ជា អក្សរ អ្នក ណា ទេ សរសេរ មក ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ថែម ទៀត *(I don't know who wrote the letter in order for me to ask more questions)*
ROVER Baseline Encoder init. Decoder init. Enc.-Dec. init.	ខ្ញុំ មិន ប្រាកដ ថា អ្នក ណា ជា អ្នក ចារ លើ សេចក្តី សារភាព នេះ ដើម្បី ឲ្យ ខ្ញុំ អត់ សួរ បន្ថែម ទៀត *(I don't know who wrote this confession in order me to not ask more questions)* ខ្ញុំ មិន ប្រាកដ ថា អ្នក ណា ជា អ្នក ចារ លើ សេចក្តី សារភាព នេះ ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ថែម ទៀត *(I am not sure who wrote this confession in order me to ask more questions)* ខ្ញុំ មិន ដឹង ថា អ្នក ណា ជា អ្នក ចារ លើ សេចក្តី សារភាព នេះ ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ត ទៀត ទេ *(I am not sure who wrote this confession in order me to continue to ask more questions)* មិន ដឹង ជា អក្សរ អ្នក ណា ទេ សរសេរ មក ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ថែម ទៀត *(Don't know who wrote in order me to ask more questions)* ខ្ញុំ មិន ប្រាកដ ថា អ្នក ណា សរសេរ ចម្លើយ សារភាព នេះ ដើម្បី ឲ្យ ខ្ញុំ សួរ បន្ថែម ទៅ លើ អ្នក ទោស នោះ ទេ *(I am not sure who wrote this confession in order me to ask the prisoners more questions)*

Figure 3.6: Examples of all methods in French-Khmer MT models, the *italic text* in the "()" is the translated text into English.

3.4 Monolingual ASR Corpus of Khmer ECCC

The generated corpus in Section 3.1 is usable for many downstream tasks. However, the performance of the Khmer ASR system has a large gap compared to

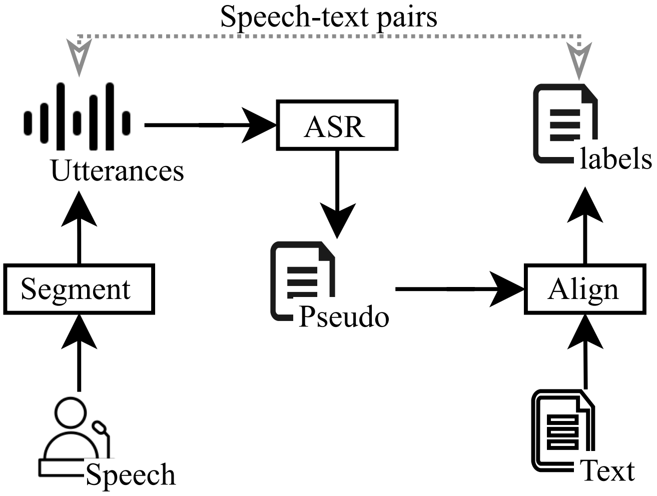


Figure 3.7: Process of creating monolingual speech-to-text of the ECCC corpus

the English ASR system. There are two problems: Khmer speech, which contains a long noise or silent speech, and label of speech, which was not well prepared because of transcribing a long speech data and domain mismatch in the ASR pretrained model. We thus conduct the new alignment method to create only the ASR corpus of Khmer by following Figure 3.7. The speech-to-text alignment has three main steps.

Algorithm 1 Splitting the audio data for ASR

```
**Result:** Small chunks of audio
1  while i is a chunk do
2      $length \leftarrow duration(i)$   if length < 1s then
3          $delete(i)$
4      else if length < 3s then
5          $merge(i-1, i)$
6      else if length > 30s then
7          $resegment(i)$   go to line 1
8      else
9          $save(i)$
10     end
11 end
```

The process was started by first splitting the long audio file into the small

Table 3.9: CER (%) and WER (%) of Khmer ASR baseline on ECCC

Model	CER (%)		WER (%)	
	Test	Dev.	Test	Dev.
Transformer w/ SP + SA	11.30	7.46	18.27	15.48

chunk based on the zero-cross number and power level[5]. After that, we apply the algorithm 1 to filter the chunk of audio to make sure that these audio files have to be at least 3s and no longer than the 30s in length.

Secondly, we generated the pseudo labels for each chunk of audio using the Vosk[6] with a new acoustic model from the Basic Expressions Travel Corpus [77] that was used in [28].

We finally conducted sentence alignment between the pseudo labels of ASR output and the original text corpus. At this stage, the alignment was able to generate the ground-truth sentence from the corresponding pseudo sentence without requiring the timestamp information.

Finally, we can generate the speech-text pairs data for about 186 hours of speech. This corpus was then randomly split into 166, 10, and 10 hours for training, testing, and evaluation sets, respectively. This ASR corpus was then evaluated by training the Transformer-based ASR model using the configuration same as in Section 3.2. Speed perturbation (SP) and SpecAugment (SA) were also applied in this training. The performance was evaluated in both WER and character error rate (CER).

Overall, we can save large speech-text pairs for Khmer, which is about 30 hours of speech larger than in TriECCC. Moreover, the ASR system of this corpus was much better, obtaining the WER of testing and development sets by 18.27% and 15.48%, respectively. That is reasonable because this corpus is monolingual which was not aligned with other languages. Additionally, the generated pseudo label is also better with the sort utterance decoded, leading to a good alignment with the ground-truth text. However, this corpus is only for ASR systems and speech classification. The performance of the Khmer ASR

[5] https://julius.osdn.jp/refman/adintool.html.en/

[6] https://alphacephei.com/vosk/

system is still limited compared to English and French ASR systems in TriECCC because of the disfluency and emotional speech of native Khmer speakers. Noted that this corpus has 28 speakers. There are 18 speakers in training and evaluation sets and 10 speakers in the testing set.

3.5 Conclusions

In this work, we created the largest-ever simultaneous SLT corpus from the ECCC dataset of 222 sessions for six directions in Khmer, English, and French. We kept a large proportion of the original dataset by using monotonic sentence alignment and word-based distance calculation. This alignment requires the segmentation of the sentences in the source language only. This method is very effective and helpful in aligning a rich-resource language with other low-resource languages. Finally, we built the 146, 148, and 125 hours in length of speech and 1.6, 1.2, and 1.3 million words in the text of Khmer, English, and French, respectively. Furthermore, we conducted E2E ASR, MT, and ST experiments on the constructed corpus and obtained reasonable performance.

To improve the Khmer MT, we conducted ROVER and finetuned the pretrained models of English-to-French and French-to-English. The results show that the ROVER is practical for combining systems with similar performance. Meanwhile, the use of the pretrained model was effective in improving the BLEU score. Initializing both the encoder and decoder modules is the most effective.

Additionally, we have constructed an exclusive Khmer speech-to-text corpus with a concrete baseline system.

These corpora will be useful for speech and language research of the Khmer language. It will be helpful for many kinds of applications in speech and language processing research, including ASR, MT, and ST, and multi-lingual or multi-source ASR, MT, and ST, or even speech classification tasks. Moreover, this alignment method will benefit similar datasets such as meetings, classroom lectures, and TV programs.

Chapter 4

Incorporating Speaker Information for ASR in Speaker-imbalanced Corpus

End-to-end (E2E) modeling [6–9, 41] solves the complex problem of sequence labeling between the input speech and output labels. It has been applied to automatic speech recognition (ASR) and speaker recognition (SRE), achieving promising results. In Chapter 3, we built the large spontaneous speech corpora which are usable for many kinds of downstream tasks in E2E modeling. Among them, ASR and SRE are complementary to each other. This means that we can decipher speech content and other meta information together and simultaneously. In other words, when we identify speakers, it is often easy to recognize their speech.

In this context, several previous studies investigated the embedding of speaker information into E2E ASR systems. In [93], speaker-representing features were extracted using a sequence summary network and then added to the encoder input as auxiliary features. Instead of using i-vectors directly as speaker embedding, Fan et al. [94] generated speaker embedding by concatenating the attention of the encoder output to i-vectors at each time step. Similarly, a speaker-aware persistent memory [95] concatenated i-vectors to the self-attention part of the Transformer speech encoder [9]. Within the same architecture, Shetty et al. [96] studied the effectiveness of providing speaker information to ASR, such as one-hot speaker vector and x-vector embedded into the input and output of the encoder, and

Sari et al. [97] proposed the speaker embedding by concatenating the memory vector (M-vector), a memory block that holds the extracted speaker i-vectors from training data and relevant i-vectors from the memory through an attention mechanism, to the acoustic features or to the hidden layer. In this approach, speaker information is used to improve ASR, but it neither explicitly conducts SRE nor uses the supervision of the speaker information for ASR.

Another approach is joint SRE and ASR. Multi-task learning (MTL) is introduced to unify the training of transcribing the speech and identifying the speakers simultaneously by sharing the same speech feature extraction layers [98–100]. Adversarial learning (AL) adopts a similar architecture to that of MTL but learns a speaker-invariant model so that it is generalized to new speakers by reducing the effects of speaker variability [101–105]. Most recently, speech attribute augmentation (SAug) was introduced as a fully E2E system integrating SRE and ASR; SAug adds the speaker attribute tags into the training label and generates those tags together with the transcription in a single encoder-decoder model [106–108]. Unlike the speaker embedding approach, however, these methods do not use the speaker information explicitly for ASR.

Generally, E2E models require a large amount of speech corpus and work well with a balanced amount of speech per speaker, as in the cases of Librispeech [109], TEDLIUM [110] and the Corpus of Spontaneous Japanese (CSJ) [111]. On the other hand, in many low-resource languages, this assumption does not hold and utterance amounts over speakers are unbalanced, in that there are often dominant speakers and auxiliary speakers. This is called the class imbalance or speaker imbalance problem. It also occurs even in resource-rich languages in many cases such as TV programs, meetings, and court proceedings, in which there is a limited set of speakers.

In this Chapter, we address the effective use of speaker information for ASR and also tackle the speaker-imbalanced problem using the corpus of the Extraordinary Chambers in the Courts of Cambodia (ECCC). We identify major speakers and compensate minor speakers by clustering them. Inspired by the speaker embedding for ASR, we propose an extension of MTL that shares the

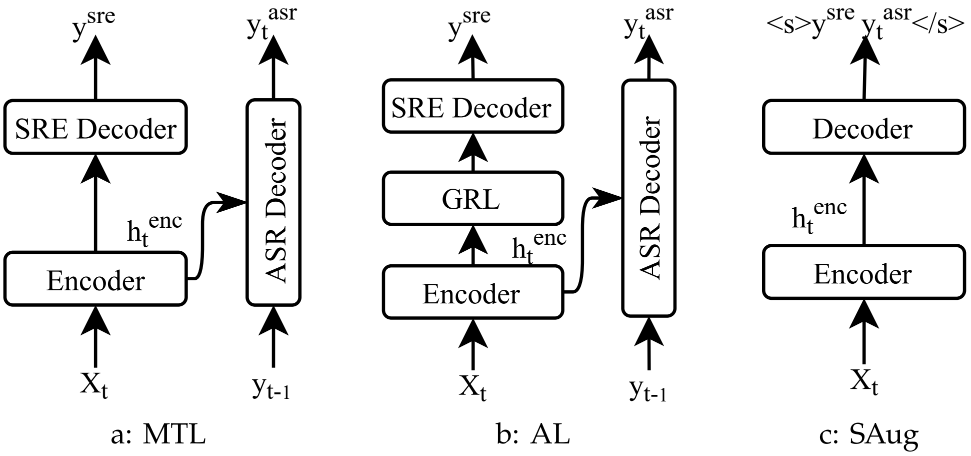


Figure 4.1: Overview of joint speaker and speech recognition methods. MTL: multi-task learning, AL: adversarial learning, SAug: speech attribute augmentation.

encoder for SRE and ASR, and takes the speaker output of SRE as the speaker embedding, then feeds it to the ASR decoder. We investigate the effectiveness of using this speaker embedding in the Transformer decoder. We also compare our proposed method with MTL, AL, and SAug systems, which perform the SRE and ASR simultaneously.

4.1 Joint Speaker and Speech Recognition

In this Section, we review previous methods for joint speaker and speech recognition. We present the system architectures built on top of the Transformer architecture to produce the speaker ID and speech transcription in single or multiple decoders.

4.1.1 Multi-Task Learning (MTL)

In MTL, both SRE and ASR are given the same sequence of acoustic features $X_t = \{x_1, ..., x_n\}$ as input. A speaker ID, s, is predicted in SRE, whereas a sequence of vocabulary tokens $Y_t = \{y_1, ..., y_m\}$ is predicted in the ASR decoder. In this system, we can benefit from sharing the same encoder and employ a dual decoder of these tasks as shown in Figure 4.1a. The encoder and ASR decoder are based on the Transformer architecture, whereas the SRE decoder comprises two linear layers followed by the ReLU and softmax activation functions. Thus, in training this MTL network, we jointly optimize both SRE and ASR losses. The loss is therefore defined as:

$$\mathcal{L}_{total} = (1-\alpha)\mathcal{L}_{asr} + \alpha * \mathcal{L}_{sre}, \tag{4.1}$$

where α is the weight of the SRE task.

This joint recognition is possible when the number of speakers is limited and there is a large amount of data for each speaker. However, this method cannot be applied to the case of many speakers with little data for each, and thus we introduce clustering of the speakers.

4.1.2 Adversarial Learning (AL)

Similar but different from MTL, AL learns an acoustic representation that is speaker invariant to reduce speaker variability by incorporating the adversarial loss of SRE, which is combined with the loss of ASR. This network has a similar architecture to MTL, but it uses the gradient reversal layer (GRL) shown in Figure 4.1b, which reverses the gradient of backward propagation [101].

Let the parameters θ_{enc}, θ_{asr} and θ_{sre} respectively denote the encoder, ASR and SRE decoders. The parameters are updated via back-propagation as follows:

$$\theta_{asr} \longleftarrow \theta_{asr} - \epsilon\frac{\partial\mathcal{L}_{asr}}{\partial\theta_{asr}}, \tag{4.2}$$

$$\theta_{sre} \longleftarrow \theta_{sre} - \epsilon\frac{\partial\mathcal{L}_{sre}}{\partial\theta_{sre}}, \tag{4.3}$$

$$\theta_{enc} \longleftarrow \theta_{enc} - \epsilon(\frac{\partial\mathcal{L}_{asr}}{\partial\theta_{enc}} - \lambda\frac{\partial\mathcal{L}_{sre}}{\partial\theta_{enc}}), \tag{4.4}$$

where ϵ is a learning rate and a negative coefficient -λ is used to remove the speaker variability from the speaker classification.

AL learns to improve the ASR as a main task, whereas SRE is an auxiliary task. AL is expected to be robust for unseen speakers, but it does not leverage speaker information for ASR.

4.1.3 Speech Attribute Augmentation (SAug)

SAug is a fully E2E method integrating SRE and ASR in a single encoder-decoder architecture. The speech attribute is analogous to a language ID in a multilingual system. It can be a speaker ID, gender, or age label [107]. Speech attributes are placed in front of the lexical token sequence of each utterance. Given a sequence of acoustic features X_t, a model must produce a label sequence $Y_t = \{s, y_1, ..., y_m\}$, where s is a speech attribute and y is a sequence of vocabulary tokens.

This network is usually trained to output the attribute label at the beginning of speech transcription for each utterance with a single decoder as shown in Figure 4.1c, and thus we do not have to prepare classifiers for the attributes explicitly. However, it is reported that the speaker ID attribute is not effective for improving ASR [107] because SRE and ASR are usually correlated negatively to each other. It is therefore impractical to improve the performance of these tasks together in a single decoder.

4.2 Proposed Method

As presented in Section 4.1, MTL and AL do not use speaker information explicitly for ASR, whereas SAug uses a single decoder. In this study, we propose the direct use of the speaker embedding of the SRE output to the ASR decoder. Unlike the previous speaker embedding, the proposed architecture is an E2E network, conducting both ASR and SRE with supervision for speaker IDs. The proposed system is expected to be useful for speaker-sparse and imbalanced datasets. The speaker information is effective for major speakers, and speaker clustering is conducted for minor speakers. The proposed network injects a speaker output (y^{sre}) into the ASR decoder as shown in Figure 5.3. We investigate five options, namely self-attention (A), after self-attention (B), cross-attention (C), after cross-attention (D), and after the feed-forward network (E). Each of these methods is tested one by one, and the combination of two methods, such as AC and BD, is also evaluated. In this process, the ASR loss is backpropagated to the SRE network, which means SRE is enhanced based on ASR.

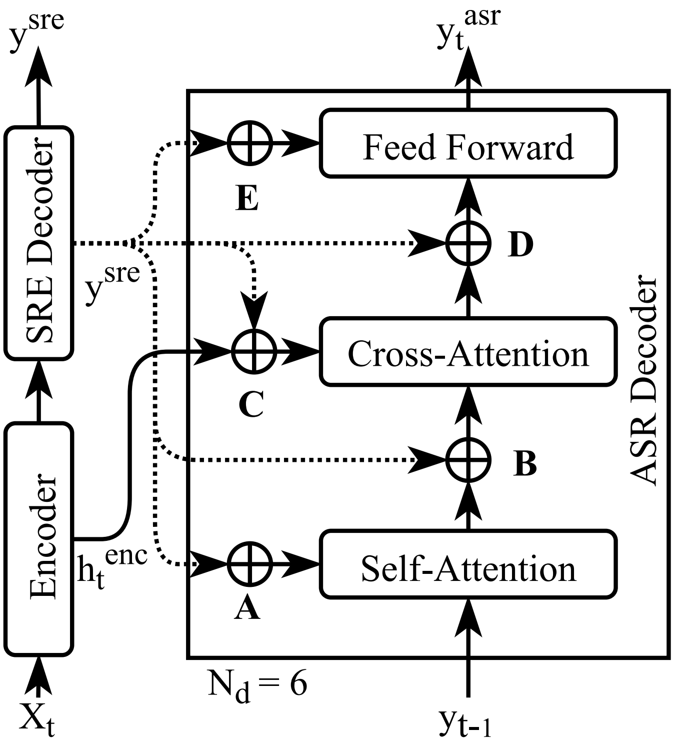


Figure 4.2: Proposed method. $\oplus$ denotes the "sum" operation.

Let h_t^{enc}, y^{sre}, y_t respectively denote the encoder output, the SRE output, and the decoder input at time step t. The embedding operation comprises the weighted sum of h_t^{enc} and y^{sre} or that of y_t and y^{sre}. Note that the operation in B, D, or E is merging a residual connection at the early time step $t-1$ (y_{t-1}) and the speaker information (y^{sre}). Meanwhile, in A, y^{sre} is merged with the key of self-attention at the previous time (K_{t-1}), and in C, y^{sre} is combined with the key of cross-attention at the current time (K_t).

4.3 Experimental Evaluations

4.3.1 Data Setup

In this Chapter, we use the monolingual ASR corpus of ECCC (Section 3.4) that is comprised of 186 hours with 28 speakers (22 male and 6 female). The task of this Chapter also includes the SRE, thus we will evaluate only the development set, which is randomly split by ratio 5:95 for testing and training as presented in

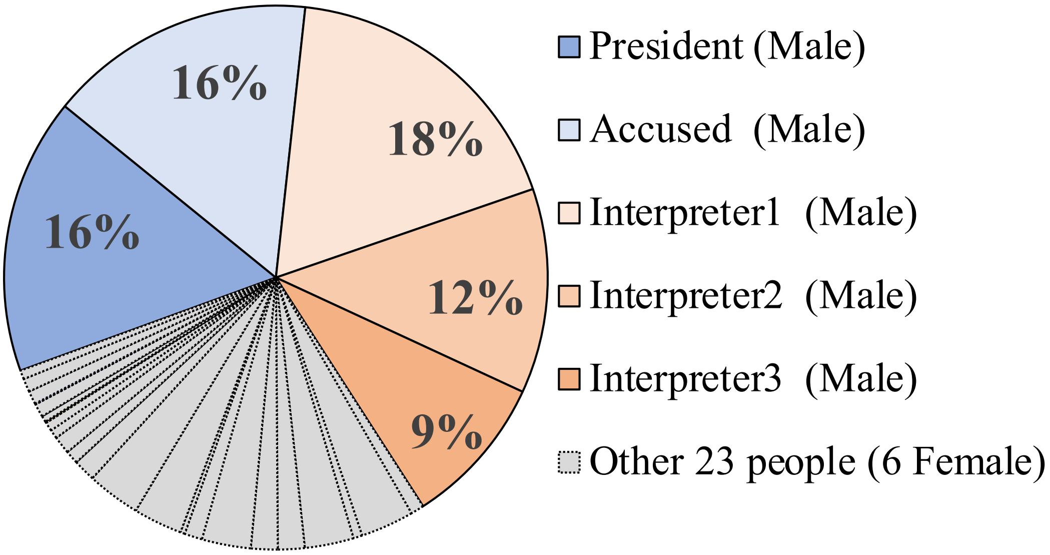


Figure 4.3: Speaker distribution in the ECCC; five major speakers are the president of the chamber, the accused, and three interpreters.

Table 4.1: Data statistics used in these experiments

Dataset	#utterance (#hour)	#character
Training	166	6.02 M
Test (development)	3,733 (10)	294 K

Table 4.1.

Figure 4.3 illustrates the speaker distribution in the ECCC corpus based on the duration of the speech. It presents the measurement in percentage of each speaker. This pie chart shows that the dataset has a crucial speaker imbalance, in that five major speakers, the president of the chamber, the accused, and three interpreters, talk more than 70% of speech among all speakers. Similarly, it also shows that male speakers are dominant compared to female speakers that have a small proportion in the "Other people" group. It means that the gender-based classification is impractical. We thus classify the speakers into a group of six (Gr6), comprising the five major speakers and a combination of other speakers. Thus, we experiment on the original dataset of 28 speakers (Gr28) and Gr6 for our proposed method.

4.3.2 System Configurations

Baseline System

We adopt a Transformer-based ASR system, which is comprised of the encoder block ($N_e = 6$) and the decoder block ($N_d = 6$) with the feed-forward inner dimension of 1024, the model dimension of 256, and the attention head number of 4, which are unchanged in all experiments. The 80-dimensional log-Mel filter bank features, which were mean and variance normalized per speaker, were extracted with a 10-ms frame shift of a 25-ms window. We then subsampled the input features using a two-layer time-axis CNN with ReLU activation with 256 channels, stride size 2, and kernel size 3. The model was jointly trained with CTC (weight $\alpha = 0.2$). The "noam" optimizer was used with 25,000 warmup steps and an initial learning rate of 5. The model was trained with ESPnet toolkit [80] using 32 batch size for 30 epochs on a 12-GB Titan X GPU.

For the baseline SRE, we separately experimented using the x-vector [112] following Kaldi's SRE16 recipe.

MTL and AL Systems

The MTL and AL network takes the 80-dimensional log-Mel filterbank features to produce a sequence of vocabulary tokens y_t and a speaker label s separately. Here y_t has 73 characters and s has 6 speaker IDs. The ASR decoder is the same as the baseline system, whereas the SRE decoder takes a mean value of h_t^{enc} of the encoder output in 256 dimensions and feeds it to a linear layer followed by the ReLU activation function and then down-projects to six as the number of speakers using another linear layer. Finally, we use a softmax layer to generate the speaker label output. In the preliminary experiments, we tested $\alpha \in \{0.2, 0.5, 0.7\}$, and found that $\alpha = 0.5$ gives the best performance.

In the GRL of AL shown in Figure 4.1b, we multiply the gradient by $\lambda = -1$ to compute the reversed gradient at the backward propagation phase, whereas in the forward propagation phase, MTL and AL are acted in the same operation.

SAug System

The SAug system is a single encoder-decoder similar to the original Transformer architecture. The configuration of this model is therefore the same as that of the baseline system except for the output label. The six-speaker IDs were added as the speech attribute labels to ground-truth in training similar to [107]. These IDs are generated together with speech transcription. We thus calculate SRE performance with the speaker attribute label and simply remove this beginning attribute from the transcription and then calculate the character error rate (CER) for ASR performance.

Proposed System

We conduct experiments on both Gr28 and Gr6. For the speaker embedding operation, we investigate each option from A to E and layer-wise from lower to deeper by a single layer or multiple layers.

The summation of vectors is used in this operation. We take the speaker output vectors having 28 (Gr28) or 6 (Gr6) dimensions according to the number of speakers in the training set, which is enlarged via a linear layer to 256 dimensions to match the encoder layer output or decoder input. We then normalize this output using a layer normalization [113] before executing the summation operation. Only in the C option, we sum with the output of the encoder. Otherwise, we sum with a residual output of the decoder module.

4.3.3 Results and Discussions

We evaluate the performance of all ASR models on the basis of the CER, whereas the SRE performance is on the basis of the ratio of utterances of an incorrect prediction. Table 4.2 presents the best performance of ASR and SRE with each method. Only for the baseline system, ASR and SRE were conducted with different models, in which SRE is conducted with the x-vector model. The table shows that the SAug has a better result for SRE, but it is not as effective as MTL, AL, and the proposed method in terms of ASR performance. This

Table 4.2: Comparison of all systems for SRE and ASR.

System	SRE (%incorrect)	ASR (%CER)
Baseline (Gr6)		
- X-vector	9.72	/
- Transformer	/	7.46
Joint speaker and speech recognition methods		
- MTL (Gr6)	9.09	7.30
- AL (Gr6)	75.16	7.30
- SAug (Gr6)	**8.81**	7.37
Proposed method		
- Gr6 (option AC; all layers)	8.97	**7.21**
- Gr28 (option AC; all layers)	11.27	7.26

Table 4.3: Comparison of embedding options applied to all layers in the proposed method (Gr6)

Embedded option (all layers)	SRE (%incorrect)	ASR (%CER)
Option A	9.08	7.30
Option B	9.10	7.33
Option C	9.21	7.26
Option D	9.02	7.33
Option E	9.18	7.26
Option AC	8.97	**7.21**
Option BD	**8.92**	7.40

suggests that it is difficult to train the model in a single decoder. The use of AL improved the ASR, however, it does not work as SRE. MTL is effective for both tasks, but our proposed method is more effective than the other compared methods in the clustering (Gr6) settings. This demonstrates that embedding speaker information into the ASR decoder does not only improve the ASR but also tunes the performance of SRE. Moreover, Gr6 gives better performance than Gr28, showing that the combination of minor speakers is critical to solve the speaker-imbalanced problem.

Regarding the proposed method, we compared different options for the speaker embedding applied to all layers of the decoder. Table 4.3 shows that combined options AC is the most effective in both tasks. With this AC option, we

Table 4.4: Comparison of layer-wise applications of AC option of the proposed method (Gr6)

Embedded Layer (AC)	SRE (%incorrect)	ASR (%CER)
Layer 1	9.18	**7.20**
Layer 2	9.10	7.26
Layer 3	9.35	7.33
Layer 4	9.26	7.30
Layer 5	9.24	7.35
Layer 6	9.91	7.26
Layer 1,2	**9.02**	7.28
Layer 1,2,3	9.10	7.24
Layer 1,2,3,4	9.08	7.27
Layer 5,6	9.48	7.29
Layer 4,5,6	9.61	7.28
Layer 3,4,5,6	9.30	7.27

tested the layer-wise performance by embedding the speaker information into a single layer or multiple layers. Table 4.4 shows that embedding the speaker information into only a single layer is as effective as embedding into all layers in terms of ASR performance but slightly degrades the SRE performance. Moreover, it is shown that embedding speaker information into lower layers of the decoder shows better improvement for SRE and ASR together. This is reasonable as the speaker information is usually reduced in the ASR decoder.

In summary, the proposed method improved not only ASR but also SRE performance from the baseline model by 3.35% and 8.23% for ASR and SRE, respectively.

4.4 Conclusions

This chapter presents a method that integrates the speaker information into the ASR decoder and also addresses the problem of speaker-imbalanced problem in ASR by identifying major speakers and clustering other minor speakers. The proposed method outperformed MTL and AL in both ASR and SRE, and it outperformed SAug in terms of ASR performance in a large margin. It has the potential to be extended to multilingual systems in the future.

Chapter 5

Incorporating Domain and Language Information for Adapting ASR using Heterogeneous Datasets

In Chapter 4, we addressed the speaker-imbalance problem and proposed the use of speaker information as speaker embedding for the ASR task. Although it was effective in improving ASR performance, the amount of training data is very large. That is not applicable to many low-resource languages. Fortunately, large-scale pretrained models based on self-supervised learning (SSL) [114–119] have been intensively studied in speech and language processing communities. In ASR, we can achieve impressive performance for low-resource languages by finetuning only a much smaller amount of labeled data compared to training conventional end-to-end (E2E) networks from scratch [9, 10], which need massive amounts of training data [120]. This is a new gateway allowing intensive studies on ASR of low-resource languages. Moreover, finetuning the large-scale pretrained model is also effective for other downstream tasks in speech processing including speaker recognition (SRE) [121, 122], language identification (LID) [123], and speech emotion recognition (SER) [124]. Those tasks are usually complemented with the ASR because it is often easy to recognize the speech content when we identify one of the meta-information. This means we can decipher speech content and other meta information, such as domain and language, together and simultaneously.

However, some previous studies [125–128] have shown that this finetuning still requires a considerable amount of labeled data, like 10 hours, to achieve satisfactory ASR performance for languages that are not well covered by the pretrained model, such as wav2vec 2.0 [114], XLSR-53 [116], and XLS-R [117]. Among the SSL pretrained models, XLS-R is most attractive because it was trained with a massive amount of unlabeled speech data from 128 languages, and it has only a Transformer encoder module [9] that requires less time-consuming in finetuning process.

Since a labeled speech dataset of 10 hours is still difficult to collect for many low-resource languages, in this Chapter, we address effective finetuning for ASR of low-resource languages using heterogeneous datasets with meta-information classification and embedding. We address a very low-resource setting using the target-labeled dataset from only one hour to 10 hours, which can be applied to most of the living languages in the world.

The finetuning process involves adapting the target dataset in the same domain and language. In this work, we formulate the finetuning process into two adaptations: domain adaptation and language adaptation. Here domain adaptation is concerned with application systems, speaking style, and input environments. A straightforward method is to conduct domain adaptation using matched-domain datasets and then language adaptation with the target-language datasets. Although the dataset matched with both domain and language of the target task is very limited (i.e., one hour), we often have access to other datasets, which are matched only with the domain (but in different languages) or only with the language (but in different domains). This will allow the adaptation process to use the different kinds of datasets selectively. With these kinds of heterogeneous datasets, we explore the effective and efficient incorporation of these kinds of heterogeneous datasets. Specifically, we propose multi-task learning (MTL) or adversarial learning with auxiliary tasks of domain and language identifications, which are based on the same pretrained model. We then incorporate the auxiliary tasks, such as language and domain identifications, into the ASR systems for effective adaptation.

For experimental evaluations, we use a part of the corpus of Extraordinary Chambers in the Courts of Cambodia (ECCC), in which we focus on ASR of the Khmer language in the criminal court domain. We can make use of the dataset of the same corpus in different languages, i.e., English and French, for domain adaptation. We also use another dataset of the Khmer language in reading speech collected by Google. They are used for domain adaptation and language adaptation, respectively. A variety of adaptation methods are evaluated using the target datasets of one hour to 10 hours with a comparable amount of heterogeneous datasets, to see the effect of the proposed method in different settings and the impact of the data amount.

5.1 Finetuning Pretrained Model for ASR systems

In low-resource language ASR, finetuning the wav2vec 2.0 pretrained model has been shown to be effective. For instance, Yi et al. [125] improved the ASR of various spoken languages, which recorded in different scenarios from the speech in the pretrained model of English speech. Similarly, Krishna et al. [126] also improved the low-resource multi-lingual ASR of both seen and unseen languages by finetuning the wav2vec 2.0-based pretrained model.

The ASR systems can be improved by multi-step finetuning and information fusing as presented by Fatehi et al. [128], they demonstrated the improvement of low-resource ASR by two-step finetuning: first pretraining a model in a high-resource language datasets and then finetuning with the low-resource language datasets to obtain language-dependent units. Meanwhile, Yi et al. [127] improved the ASR system by incorporating the encoders of wav2vec 2.0 together with BERT [119].

Moreover, finetuning a large-scale SSL pretrained model can be effective for other tasks including speaker recognition, which were conducted by Baskar et al. [121] and Vaessen et al. [122]. Meanwhile, Tjandra et al. [123] showed an improvement in the language identification task by finetuning a large-scale SSL pretrained model. The speech emotion recognition task can also be improved by applying the MTL with ASR. This finetuning was investigated by Cai et al. [124] using the wav2vec2.0-based pretrained model.

Many studies tried to improve ASR systems by training jointly with other related tasks via MTL [98–100], or adversarial learning [101–105]. Moreover, speaker embedding [93–95] was also effective in enhancing the ASR systems. As shown in Chapter 4, we investigated speaker ID embedding for ASR, which achieved the improvement of not only ASR but also speaker recognition performance together.

5.2 Proposed method

5.2.1 Adaptation using Heterogeneous Datasets

Even in a very low-resource setting, we often have access to heterogeneous datasets that are partially matched either in domain or language. For low-resource languages, it is reasonable to use these heterogeneous datasets. Even in major languages, the matched dataset of the target task is often limited, but datasets of different domains can be exploited.

With these datasets, we can design domain adaptation and language adaptation. Domain adaptation uses domain-matched multi-lingual datasets, and language adaptation uses language-matched multi-domain datasets. However, the simple combination of different kinds of datasets makes the adaptation process difficult. Thus, we propose to use meta-information in MTL or adversarial learning. Moreover, we propose a two-step framework, in which each step conducts domain adaptation or language adaptation individually.

5.2.2 Multi-task Learning (MTL)

Let $X = \{x_1, ..., x_n\}$ as input speech, and we can predict the sequence of vocabulary tokens $Y = \{y_1, ..., y_m\}$ via ASR system and predict meta information such as domain ID or language ID via classification tasks. This process can be done through multi-task learning (MTL), which trains a neural network jointly for several different tasks but related to each other. The network learns one task and adds one or more auxiliary tasks. The auxiliary tasks aim at helping the model to converge faster and better, which can benefit the main task.

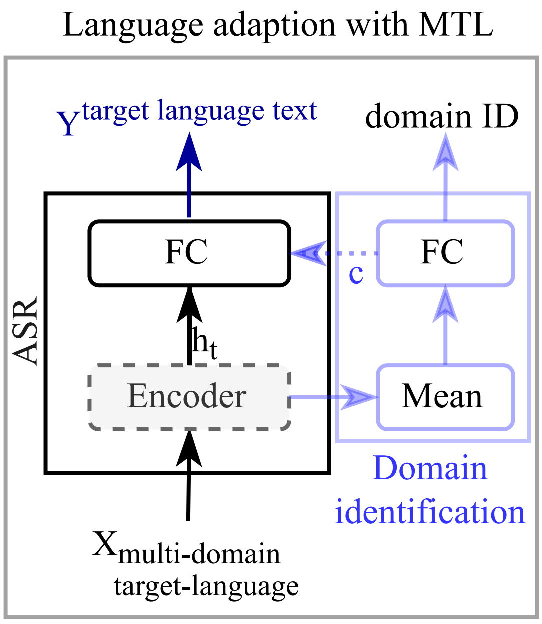


Figure 5.1: The proposed method of adaptation with multi-task learning (MTL) for language adaptation. There are two options for MTL: simple MTL and MTL with ID embedding, where ID can be the domain ID in language adaptation or language ID in domain adaptation.

Here, we are interested in incorporating the MTL for adaptation, in which the identification task is performed as an auxiliary task. For language adaptation, the classification is the domain identification, depicted in the right-most part of Fig. 5.1. It aggregates the features of the encoder output over all time frames and applies a dense layer for identification. MTL is expected to guide the network to use the datasets selectively, namely, use the in-domain dataset and the out-of-domain dataset in a different way. In the case of domain adaptation, the classification is the language identification, which is conducted in the same manner, except the selective datasets are multi-lingual in the same domain.

MTL is effective by sharing the encoder and employing dual decoders. The encoder is based on the Transformer architecture, whereas the ASR decoder is based on Connectionist Temporal Classification (CTC) [6], and the identification task comprises pooling, linear, and normalization layers followed by the softmax

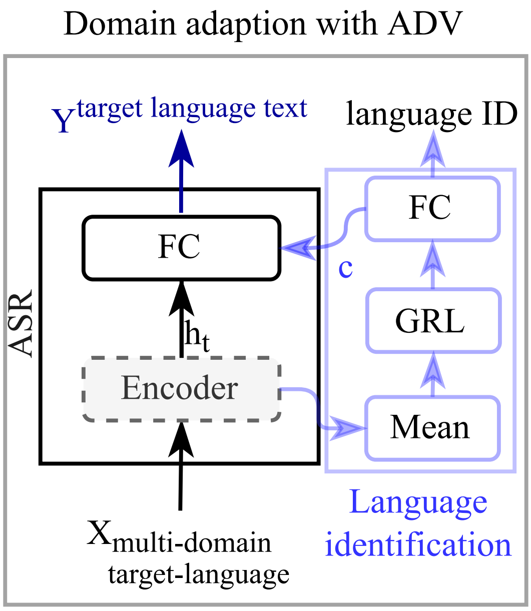


Figure 5.2: The proposed method of adaptation with adversarial learning (ADV) for domain adaptation. There are two options for ADV: simple ADV and ADV with language ID embedding.

layer. For MTL, we jointly optimize ASR and identification losses, defined as:

$$\mathcal{L}_{total} = (1-\alpha)\mathcal{L}_{CTC} + \alpha * \mathcal{L}_{CE}, \tag{5.1}$$

where α and $\mathcal{L}_{CE}$ are the weight and the cross-entropy loss of the identification task, which can be $\mathcal{L}_{\text{LID}}$ or $\mathcal{L}_{\text{DID}}$ for language or domain identification, respectively, and $\mathcal{L}_{CTC}$ is the CTC loss of the ASR task. The loss is used to finetune the entire network except for the feature extraction module of the pretrained model.

5.2.3 Adversarial Learning (ADV)

In MTL, the entire network is optimized to distinguish different languages and domains. However, domain adaptation should be done by learning language-invariant features. In this case, language ID should not be distinguished. Thus, we apply adversarial learning (ADV) to reduce language diversity in the domain adaptation method. This network has a similar architecture to MTL, but it uses the gradient reversal layer (GRL) shown in the right-most part of Fig. 5.2, which reverses the gradient of the backward propagation [101].

Let θ_{enc}, θ_{asr} and θ_{LID} be the parameters of the encoder, ASR, and language classification decoders, respectively. The parameters are then updated via back-propagation as follows:

$$\theta_{\text{asr}} \longleftarrow \theta_{\text{asr}} - \epsilon \frac{\partial \mathcal{L}_{\text{CTC}}}{\partial \theta_{\text{asr}}}, \tag{5.2}$$

$$\theta_{\text{LID}} \longleftarrow \theta_{\text{LID}} - \epsilon \frac{\partial \mathcal{L}_{\text{LID}}}{\partial \theta_{\text{LID}}}, \tag{5.3}$$

$$\theta_{\text{enc}} \longleftarrow \theta_{\text{enc}} - \epsilon (\frac{\partial \mathcal{L}_{\text{asr}}}{\partial \theta_{\text{enc}}} - \lambda \frac{\partial \mathcal{L}_{\text{LID}}}{\partial \theta_{\text{enc}}}), \tag{5.4}$$

where ϵ is a learning rate and a negative coefficient -λ is used to remove the language variability from the language classification.

For explicit guidance, adversarial learning learns an acoustic representation that is language invariant to reduce language variability by incorporating the adversarial loss of language identification (LID) combined with the loss of ASR. The loss is calculated as Eq. (5.1), while the weighted sum of language identification embedding is computed as Eq. (5.5).

5.2.4 Embedding Domain and Language ID for ASR Decoder

For more explicit guidance, the domain ID embedding as the result of domain identification is fused to the encoder output, which is used for the ASR decoder, in domain adaptation, whereas language identification is fused to the encoder output in language adaptation. In this case, we add two more layers, a linear layer and a normalization layer [113], for the identification output. Then, the summation of the vectors is used in fusing with the encoder output. Here, we introduce a weighted sum of the ID embedding c and the encoder output of the pretrained model h_t to compute the final output h'_t used for ASR.

$$h'_t = h_t + \gamma * c, \tag{5.5}$$

where γ is the weight of the identification task. Here, the identification result for the utterance c is used for the entire frames.

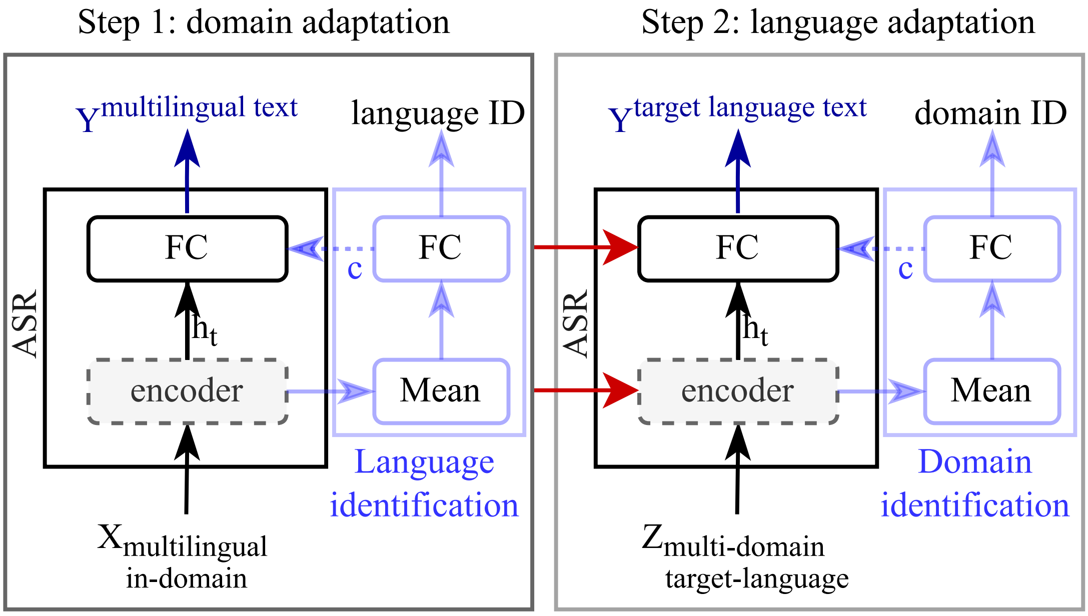


Figure 5.3: The proposed method of two-step adaptation using heterogeneous datasets. Step 1: domain adaption with multilingual in-domain datasets, Step 2: language adaptation with multi-domain datasets of the same target language. Domain adaptation and language adaptation can be done in different orders.

5.2.5 Incremental Two-step Adaptation

Then we conduct both domain adaptation and language adaptation. Rather than conducting them jointly, we propose a two-step framework. In each step, either domain adaptation or language adaptation is conducted with MTL or adversarial learning. This framework allows each step to focus on individual adaptation.

The two-step adaptation is illustrated in Fig. 5.3. This figure shows that domain adaptation is conducted in the first step, and then language adaptation is done in the second step, but they can be performed in a different order. The pretrained model used in the first step is the original XLS-R, a wav2vec 2.0-based multilingual SSL speech representation model. Then, we use the finetuned model from the first step as the pretrained model for the second step.

5.3 Experimental Evaluations

5.3.1 Datasets

The target task is an automatic transcription of Khmer speech in ECCC (Extraordinary Chambers in the Courts of Cambodia) [55]. The trilingual speech translation corpus (TriECCC) was also compiled (Section 3.1) for Khmer, English, and French [84, 129]. Thus, we can use English and French datasets for domain adaptation.

In addition, we have access to another Khmer speech dataset of Google text-to-speech corpus,[1] which is reading-style. It is matched with the language but much different in the domain, such as vocabulary, speaking styles, and recording environments. Thus, it is used for language adaptation.

For training datasets, we randomly select the target-task dataset ECCC (ECCC_KM) [55] from one hour to 10 hours as presented in Table 5.1, one hour to 10 hours per language of English (ECCC_EN) and French (ECCC_FR) from TriECCC [129], and one hour to 4 hours of Khmer speech from Google text-to-speech (Google_KM), whereas the testing and validation sets are the original data from ECCC [55], each about 10 hours.

In summary, domain adaptation is conducted with the combination of ECCC_KM, ECCC_EN, and ECCC_FR in three folds of speech: 3 hours, 15 hours, and 30 hours. It also can be used for three-language identification of Khmer, English, and French. On the other hand, language adaptation is conducted with three folds of speech from ECCC_KM and Google_KM: 2 hours, 9 hours, and 14 hours, which can be used for two domain classifications of ECCC (court) and Google (read).

5.3.2 System Settings

We conducted experiments using XLS-R with 300 million parameters,[2] which is a large-scale wav2vec 2.0-based multilingual pretrained model for speech. It is a

[1] https://openslr.org/42/

[2] https://huggingface.co/facebook/wav2vec2-xls-r-300m

Table 5.1: The heterogeneous datasets used in these experiments

dataset	#hour	description
ECCC_KM	$1, 5, 10$	In-domain target Khmer
ECCC_EN	$1, 5, 10$	In-domain English for domain adaptation
ECCC_FR	$1, 5, 10$	In-domain French for domain adaptation
Google_KM	$1, 4, 4$	Out-of-domain Khmer for language adaptation

Transformer-based model comprised of 7 convolutional neural network (CNN) layers (each layer has 512 channels) and 24 encoder layers (each hidden layer size is $1,024$). The implementation is based on the Transformers [9]. During finetuning, we froze all the CNN layers, which are used primarily for feature extraction and had already been sufficiently trained during pretraining. A linear layer is added on top of the Transformer encoder layers. This linear layer takes the contextualized output of the encoder and converts them to tokens for ASR with the softmax operation. The CTC loss, which does not require the alignment information between the output sequences and the input speech, was used as the objective loss function of ASR. In this study, we use 112 output characters, which are defined by the vocabulary in multi-lingual settings (74 characters for Khmer).

In both MTL and adversarial learning of ASR and domain/language identification, the weight α in Eq. (5.1) was set to 0.01, and the weight to the domain/language ID embedding γ in Eq. (5.5) was set to 0.01.

To speed up the training time, we group samples of similar input lengths into one batch to reduce the overall number of useless padding tokens passed through the model. The seed of learning rate was set to $3e$-4 to warm up until the finetuning has become stable. During training, SpecAugment [83] was also applied by masking some time frames and channels, and the last 2 checkpoints were saved asynchronously for every 500 training step. Each checkpoint was used to decode the validation set and evaluated with the character error rate (CER). Due to the large memory consumption, we used 16 batch sizes in each GPU with 2-step gradient accumulation on 2 GPUs. The total training batch size was 64, with $5,000$ in training steps for all models.

An input speech sample is decoded at inference time with a single step

Table 5.2: Effect of domain and language adaptation in a single step

method	#hour	CER(%)
one-hour target		
in-domain target: ECCC_KM (Baseline)	1	21.74
out-of-domain: Google_KM	1	35.01
domain adaptation: ECCC_{KM, EN, FR}	3	17.75
language adaptation: {ECCC, Google}_KM	2	**16.12**
domain and language adaptations: ECCC, Google	4	18.79
5-hour target		
in-domain target: ECCC_KM (Baseline)	5	12.11
out-of-domain: Google_KM	4	32.26
domain adaptation: ECCC_{KM, EN, FR}	15	13.61
language adaptation: {ECCC, Google}_KM	9	**12.09**
domain and language adaptations: ECCC, Google	19	15.07
10-hour target		
in-domain target: ECCC_KM (Baseline)	10	11.10
domain adaptation: ECCC_{KM, EN, FR}	30	12.30
language adaptation: {ECCC, Google}_KM	14	**10.89**
domain and language adaptations: ECCC, Google	34	14.57

by the finetuned ASR model. In experimental evaluations, we tested various domain and language adaptation combinations in different orders with MTL and adversarial learning.

5.4 Results

We evaluate the performance of all ASR models based on the character error rate (CER) of the 10-hour test set of Khmer ECCC for ASR in Section 3.4. Note that the experimental results in Chapter 4 were evaluated on the development set of the Khmer ECCC for ASR in Section 3.4.

5.4.1 Baseline One-step Finetuning

Table 5.2 presents the simple baseline adaptation results of a domain or language in a single step individually. The domain adaptation is done by finetuning with the combination of the target dataset ECCC_KM with other languages in the same domain, ECCC_EN, and ECCC_FR, whereas the language adaptation is done by the combination with the out-of-domain dataset in the same language, Google_KM. Here, meta-information is not used and thus MTL or adversarial learning is not conducted.

The first part presents the results of finetuning using one hour of the target-labeled speech data. The baseline is finetuning only with the ECCC_KM, which is matched with domain and language. Its CER is 21.74%. For reference, when we finetune with the Google_KM dataset, the CER is much worse (35.01%), which confirms a serious mismatch in terms of the domain. When we conduct domain adaptation, a large improvement (3.99% absolute) is gained from the baseline despite using speech data from different languages. The result confirms the significance of domain adaptation. When we conduct language adaptation, we achieve an even larger improvement (5.62% absolute). The result demonstrates the effect of language adaptation is larger than that of domain adaptation. This is partly because the target language is not covered well in the pretrained model of XLS-R. However, combining all datasets in domain and language is not as effective as individual domain and language adaptation. It is not straightforward to conduct language adaptation and domain adaptation jointly.

The second part of Table 5.2 presents the results for a 5-hour target-labeled dataset, 5-hour of each language from ECCC_EN, and ECCC_FR, and 4-hour of Google_KM.[3] The baseline of finetuning ECCC_KM in 5 hours of the target labeled speech data reduces the CER absolutely by 9.63%. It is a large improvement from the one-hour setting. This result shows the effect of increasing the data matched in both domain and language. An additional improvement is obtained by conducting the language adaption with the combination of the target dataset, ECCC_KM, and a matched language data, Google_KM. However, the combination of multi-lingual ECCC in domain adaptation and multi-lingual multi-domain adaptation is not effective. These results show that it is not straightforward to incorporate domain adaptation with this data size.

The last part of Table 5.2 presents the results of increasing the target speech data of ECCC_KM to 10 hours. The baseline system shows an improvement

[3] Google_KM has speech of only 4 hours in total.

Table 5.3: Effect of MTL or adversarial learning (ADV) and domain or language ID embedding in a single-step adaptation. (DID: domain identification, LID: language identification)

method	#hour	CER(%)
one-hour target		
language adaptation w/ DID MTL	2	16.47
+ w/ DID embedding (+ DID)	2	16.23
language adaptation w/ DID ADV	2	17.58
+ w/ DID embedding (+ DID)	2	16.57
domain adaptation w/ LID MTL	3	17.67
+ w/ LID embedding (+ LID)	3	16.11
domain adaptation w/ LID ADV	3	17.57
+ w/ LID embedding (+ LID)	3	**16.07**
5-hour target		
language adaptation w/ DID MTL	9	12.51
+ w/ DID embedding (+ DID)	9	**11.26**
language adaptation w/ DID ADV	9	12.56
+ w/ DID embedding (+ DID)	9	11.28
domain adaptation w/ LID MTL	15	13.33
+ w/ LID embedding (+ LID)	15	12.37
domain adaptation w/ LID ADV	15	13.37
+ w/ LID embedding (+ LID)	15	12.30
10-hour target		
language adaptation w/ DID MTL	14	10.89
+ w/ DID embedding (+ DID)	14	**10.23**
language adaptation w/ DID ADV	14	10.90
+ w/ DID embedding (+ DID)	14	10.29
domain adaptation w/ LID MTL	30	13.02
+ w/ LID embedding (+ LID)	30	11.28
domain adaptation w/ LID ADV	30	12.01
+ w/ LID embedding (+ LID)	30	11.26

by reducing the CER to 11.10%. In this case, too, domain adaptation does not improve but degrades the performance, while language adaption still improves the performance. These results show that increasing in-domain multi-lingual datasets does not benefit from multi-lingual ASR finetuning. However, out-of-domain data in the same language provides additional improvement even if it is small in size, especially for low-resource languages.

5.4.2 MTL and Adversarial Learning (ADV)

Table 5.3 presents the effectiveness of domain and language adaptation in a single step applying MTL and adversarial learning (ADV) with or without ID embedding. Language embedding is conducted in domain adaptation, whereas domain embedding is conducted in language adaptation.

The first part of Table 5.3 presents the results using a one-hour target-labeled dataset. The performance of the domain adaptation is significantly improved by MTL with language identification and embedding, which allows the model training to use the Khmer speech selectively. The result is comparable to the case of language adaptation in Table 5.2. Interestingly, applying adversarial learning in domain adaptation with language identification and embedding resulted in a larger improvement. Reducing the language variance is effective for enhancing the ASR in low-resource settings. On the other hand, domain identification does not help language adaptation so much. It is noted that both domain and language identifications were done almost 100% correctly in MTL, as they are relatively easy tasks, and language identification performance was almost 100% incorrect in adversarial learning.

A similar effect of domain adaptation by MTL with language identification and embedding is observed in the settings of 5-hour and 10-hour target datasets. In these settings, however, the effect of language adaptation by MTL with domain identification and embedding is superior. We also observe that domain adaptation was not so effective in these settings. The results suggest that when we have a sufficient amount ($>\sim$10 hours) of training data of the target language, incorporating data from different languages is less effective.

5.4.3 Two-step Finetuning

Table 5.4 presents the results of the proposed two-step adaptation method. The evaluations were conducted by applying MTL or ADV in each step and comparing with simple ASR single-task learning (STL). As presented in Table 5.3, the domain and language ID embedding were consistently effective, we thus always use the

Table 5.4: Results of two-step adaptation using only one-hour matched training dataset. (n, m) represents the number of hours in adaptation data of the first step (n) and the second step (m). (DID: domain identification, LID: language identification), "+ ID": ID embedding

method	#hour	CER(%)
two-step finetuning (ASR-STL → ASR-STL)		
domain → language adaptation	$(3, 2)$	15.59
language → domain adaptation	$(2, 3)$	16.00
two-step finetuning (MTL/ADV → MTL/ADV)		
domain-MTL + LID → language-MTL + DID	$(3, 2)$	**14.83**
language-MTL + DID → domain-MTL + LID	$(2, 3)$	15.04
domain-ADV + LID → language-MTL + DID	$(3, 2)$	15.21
language-MTL + DID → domain-ADV + LID	$(2, 3)$	14.92

Table 5.5: Results of two-step adaptation using 5-hour and 10-hour matched training datasets. (DID: domain identification, LID: language identification)

method	#hour	CER(%)
5-hour target		
domain-MTL + LID → language-MTL + DID	$(15, 9)$	**10.14**
language-MTL + DID → domain-MTL + LID	$(9, 15)$	11.01
domain-ADV + LID → language-MTL + DID	$(15, 9)$	11.16
language-MTL + DID → domain-ADV + LID	$(9, 15)$	10.20
10-hour target		
domain-MTL + LID → language-MTL + DID	$(30, 14)$	**10.07**
language-MTL + DID → domain-MTL + LID	$(14, 30)$	10.14
domain-ADV + LID → language-MTL + DID	$(30, 14)$	10.10
language-MTL + DID → domain-ADV + LID	$(14, 30)$	10.09

ID embedding option in MTL and adversarial learning in this experiment.

When we compare the results of the ASR-STL condition in the first part with those of Table 5.2, the two-step adaptation always gives an additional large improvement. Among them, domain adaptation followed by language adaptation obtained the most significant improvement. As the language adaptation is more effective than the domain adaptation, as shown in Table 5.2, the better-matched dataset must be used in the final finetuning.

The next part of Table 5.4 presents the effect of MTL or ADV applied in each step. In this experiment, domain and language ID embedding is always adopted.

We observe a significant improvement from the ASR-STL (upper part of Table 5.4) and also the single-step MTL or ADV (top part of Table 5.3). The result shows the combined effect of the two-step adaptation method and MTL or ADV. There is not much difference in the order of the adaptation, but the best performance was achieved by first applying domain adaptation and then applying language adaptation. The use of adversarial learning was also effective but comparable to MTL.

Finally, we present the results of the two-step adaptation in the settings of 5-hour and 10-hour target datasets in Table 5.5. We again observe that the best performance was achieved by first applying domain adaptation and then applying language adaptation. We also observe a significant improvement from the single-step adaptation for the 5-hour setting, but the improvement is not so much for the 10-hour setting. In fact, there is little difference in CER between the 5-hour and the 10-hour settings in Table 5.5. These results suggest that the performance achievable by finetuning the SSL pretrained model becomes almost saturated around this point. This means that the proposed method achieves almost saturated performance only with the target dataset of 5 hours.

5.5 Conclusions

We have presented effective finetuning methods of two-step domain and language adaptation using heterogeneous datasets for very low-resource settings. The domain adaptation is composed of either MTL or adversarial learning with language ID embedding using the matched datasets in the domain, whereas language adaptation is conducted MTL with domain ID embedding using the matched language data.

In experimental evaluations, finetuning the ASR model with MTL or adversarial learning is effective in all settings and all adaptation steps. The two-step adaptation consistently outperforms the single-step adaptation in all settings. The best improvement was obtained by conducting the domain adaptation followed by language adaptation (with ID embedding of MTL in both steps). The best case improves the CER of the baseline relatively by 31.8%, 16.3%, and 9.3% for one-hour, 5-hour, and 10-hour target speech datasets, respectively.

Our findings in this Chapter are as follows: (1) we can achieve a stable performance by using only 5 hours of target speech dataset, which will be applied to most languages in the world, (2) increasing the language-matched dataset can benefit low-resource settings, and (3) the domain adaptation using datasets of other languages can be effective only for very small training datasets (one-hour) or by using the two-step adaptation frameworks. In the future, we will investigate the effectiveness of the proposed method by increasing other language datasets or applying this method to other low-resource languages.

Chapter 6

Leveraging Simultaneous Translation for Enhancing Transcription of Low-resource Language

While end-to-end (E2E) modeling [6, 8–10, 41] has significantly advanced automatic speech recognition (ASR), ASR of low-resource languages still remains one of the big challenges. Another task for low-resource languages is machine translation (MT) or spoken language translation (SLT) because many foreign people do not understand these languages. In international meetings such as UN conventions [130] and EU Parliaments [59], simultaneous interpretation by human translators is often available [131]. In this Chapter, we use a TriECCC [129], in which Khmer is the primary language and English and French translations are available. MT and SLT corpora have been built on these datasets (Section 3.1).

We focus on the ASR of low-resource languages (e.g. Khmer), which is also the basis of the SLT of these languages, by leveraging the translation corpus. Here we assume ASR of fluent speech of the human translators in a resource-rich language (e.g. English and French) is perfect, thus use the output text instead of speech. Note that transcription of the original speech (i.e. Khmer) is still mainly required for the Khmer people. In this setting, the content of the back-translation of the translation text (e.g. English-to-Khmer) must be the same as the transcription of the original speech (i.e. Khmer). Therefore, the former is expected to enhance the latter, specifically, MT output is expected to complement the ASR process. This is analogous to a scenario in which Japanese people can more easily recognize a foreign-language (e.g. English) movie with simultaneous subtitles of the native language (e.g. Japanese).

In previous studies, multi-task learning and system combination of multiple models have been investigated to improve ASR performance, for example, the integration between ASR and MT models trained in multiple iterative stages [132,133]. This integration is also applied to computer-assisted translation application [134–138]. However, these works used independent systems of ASR and MT, similar to the idea in ROVER [70], which ensembles the output of multiple ASR recognizers using an alignment and then a voting mechanism. Another approach is to train a large text-only or text-to-text model to be coupled with the ASR model. Wang et al. [139] trained a large decoder of a text corpus to alleviate the need for an external language model. Yusuf et al. [140] trained a bank of shallow task-specific modality encoders including MT and masked language model (MLM) as the auxiliary task to ASR. These works require a large text corpus, which is not the case in low-resource languages.

In contrast, we propose a joint ASR-MT framework to enhance the ASR performance of a low-resource language using the MT output. It trains ASR and MT modules using input sources of speech and its parallel translation text simultaneously. The proposed method jointly trains dual encoders of ASR and MT together and then uses the translation knowledge from a rich-resource language to assist the transcription of a low-resource language via a cross-attention mechanism in a single E2E model [9]. Although the proposed method trains multiple encoders simultaneously, it is different from multi-source MT [61–63], which uses multiple inputs of text in different languages, and it is different from cascade speech translation (ST), which is stacking the ASR and MT systems, and the E2E-ST, which uses the ASR encoder and MT decoder.

We first evaluate the proposed method using the multi-lingual SLT corpus of ECCC, in which the goal is to improve the transcription performance of Khmer speech using the translation from English or French. We then apply this method to the Fisher-CallHome corpus [58] for improving the transcription of Spanish with the use of translation from English.

6.1 ASR Enhancement using MT Task

The study of enhancing the ASR system on the target language of the human translator using the translation of the source document was investigated by Paulik et al. [132], who analyzed the effects of different MT models to be integrated into the ASR system in multiple iterations. In each iteration, they updated an n-gram language model for rescoring the ASR n-best list, whereas in [133], the ASR system was improved by extracting the MT n-best list in several iterations to rescore the ASR n-best list, where both ASR and MT were conducted in parallel. Similarly, Khadivi et al. [134, 135] also integrated MT and ASR models for computer-assisted translation. In these works, they used independent ASR and MT models and then interactively updated the n-gram language model of each system in multiple iterations or integrated the outputs of these systems.

Recently, Macháček et al. [131] compared quality and latency of spoken translation systems from English to Czech using Europarl Simultaneous Interpreting Corpus. This investigation showed that the interpreters tend to compress and simplify the speech, which means the translations keep the content but are not necessarily literal. Yusuf et al. [140] proposed a framework to improve ASR with a unified speech and text encoder-decoder, in which the system jointly trained an attention-based of ASR and a variety of text-to-text transduction tasks including MT and MLM. All tasks shared parameters of the encoder layers and the decoder modules, but MT and MLM were trained on a large text corpus which is unpaired to the ASR corpus.

In this study, we enhance ASR of a low-resource language by jointly training the ASR and MT in a single E2E model using the paired data between audio-to-text for ASR and text-to-text for MT.

6.2 Proposed Method

The tasks of ASR and MT are to generate a text from a source speech and from another language text, respectively. Therefore, we propose to jointly train these ASR and MT models in a single E2E model. Specifically, we incorporate the

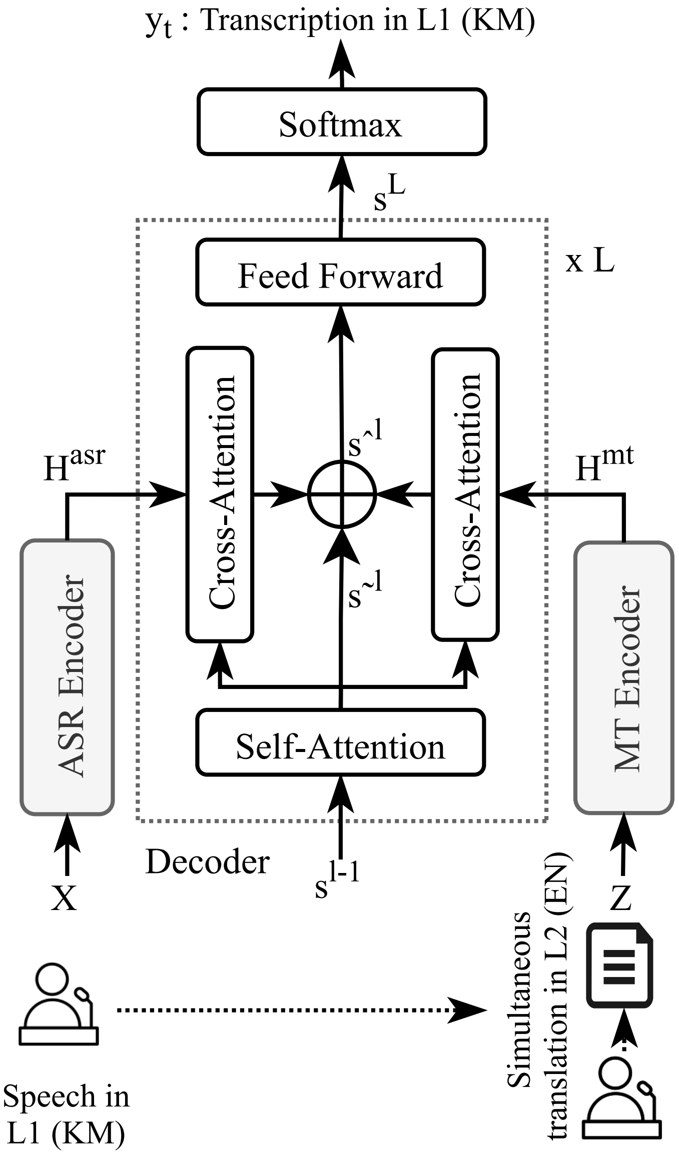


Figure 6.1: Proposed method of joint ASR and MT

translation knowledge from a rich-resource language to enhance the transcription of speech of a low-resource language.

Similar to multi-task learning, we conduct a joint training of both ASR and MT encoders as shown in Figure 6.1, in which an original speech in a language (L1, e.g. Khmer) and its corresponding translation in another language (L2, e.g. English) are used as the input sources. Note that we assume ASR of the translators' speech (fluent English/French) is perfect, thus use the transcription text instead of speech in this work[1]. We then combine the cross-attention of ASR and MT encoders to the joint decoder to improve automatic transcription. With this combination, the translation knowledge is used to enhance the transcription process.

This proposed framework formulates that, with a given set of speech utterances in L1, $\{X_1, X_2, ..., X_e\}$, and their translations in L2, $\{Z_1, Z_2, ..., Z_e\}$, the model predicts text transcription in L1, $\{Y_1, Y_2, ..., Y_e\}$, where e is the total number of sentences or utterances.

6.2.1 Dual Encoders

The proposed architecture comprises of both ASR and MT encoders. Each encoder is based on the Transformer architecture [9], but we train both encoders jointly in a single model. For each sequence of n acoustic features in L1, $X = \{x_1, x_2, ..., x_n\}$, and sequence of m tokens in L2, $Z = \{z_1, z_2, ..., z_m\}$, the encoders predict the intermediate representation matrices H^{asr} and H^{mt}.

$$\begin{aligned} H^{\text{asr}} &= \text{Encoder}(X), \\ H^{\text{mt}} &= \text{Encoder}(Z). \end{aligned} \tag{6.1}$$

6.2.2 Joint Decoder

The decoder network is implemented as a stack of L modified Transformer layers. Unlike the standard Transformer decoder, each layer in our decoder has two distinct cross attention components in order to combine information from both of the ASR and MT encoders. Specifically, the output of each layer at the t-th decoding step $S^l_t = \{s^l_1, s^l_2, ..., s^l_t\}$ is calculated using the representation from the ASR encoder H^{asr} and that from the MT encoder H^{mt}, as well as the output of the previous decoder layer S^{l-1}_t. Note that we define s^0_j as the embedding of the j-th predicted token y_i .

[1] This assumption is not so unrealistic, given the WER of Librispeech is less than 3% [141]. Moreover, the number of translators is only three in this dataset, thus we can have similar performance when we train the speaker-adapted model.

Each s_t^l is calculated as:

$$\tilde{s}_t^l = \text{Attention}(s_t^{l-1}, S_t^{l-1}, S_t^{l-1}), \tag{6.2}$$

$$\begin{aligned}\hat{s}_t^l = {} & \text{Attention}(\tilde{s}_t^l, H^{\text{asr}}, H^{\text{asr}}) + \\ & \text{Attention}(\tilde{s}_t^l, H^{\text{mt}}, H^{\text{mt}}) + \tilde{s}_t^l,\end{aligned} \tag{6.3}$$

$$s_t^l = \text{FeedForward}(\hat{s}_t^l). \tag{6.4}$$

Here, each self-attention component takes query Q, key K and value V as the inputs, and its output is obtained as:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{6.5}$$

Then, the output probability of the current token y_t is given as:

$$P(y_t|S_t^0, X, Z) = \text{Softmax}(s_t^L) \tag{6.6}$$

Finally, the probability of the transcription text Y is defined as:

$$P(Y|X, Z) = \prod_{t=1} P(y_t|S_t^0, X, Z) \tag{6.7}$$

Note that without cross-attention from MT, the network is virtually the same as the standard Transformer-based ASR system. Thus, the proposed system is regarded as its extension.

6.2.3 Objective Function

To optimize the model training, each task has a well-defined loss function. With the proposed method, there are two losses of ASR and MT, which could be optimized with multi-task learning. However, the output of these two tasks are essentially the same, and each loss is propagated to the respective model.

6.3 Experimental Evaluations

6.3.1 Dataset

SLT corpus of the Khmer to English and French, TriECCC (Section 3.1), is used. Our main target is to improve speech transcription by incorporating MT (English to Khmer or French to Khmer). The SLT corpus of 155 hours in length of speech and 1.7M words in text are used to conduct the experiments.

6.3.2 Model Training

We implemented the model using a Transformer-based architecture of the ESPnet [80]. Following the standard setup, we used 80-dimensional log-melscale filterbank coefficients and 3-dimensional pitch features. Speech perturbation [82] and SpecAugment [83] were applied for speech data augmentation. The network is composed of six encoder layers and six decoder layers. The dimension of the feed-forward network was set to 2, 048, and the dropout was set to 0.1. The model used 4-head self-attention with the dimension of 256. This network was started with down-sampling using a two-layer time-axis convolutional layer with 256 channels, stride size of 2, and kernel size of 3. The model was jointly trained with CTC (weight $\alpha = 0.3$) for 45 epochs with a single 12-GB Titan X GPU using a batch size of 64. The "Noam" optimizer was used with 25, 000 warmup steps and an initial learning rate of 5. The byte pair encoding (BPE) [81] of the source and target languages was set to 5, 000 for each. We used a joint source and target vocabularies for the proposed method, thus for each pair of English-Khmer and French-Khmer, we employed the 10, 000 BPE tokens.

The model has parallel ASR and MT encoders. The ASR encoder uses 83-dimensional source speech features as the input, while the MT encoder takes another language text as the input where the vocabulary size is the input dimension. The decoder part is comprised of two cross-attentions. The summation operation was conducted to combine the 256-dimension of each attention and residual connection into a single 256-dimension output, as shown in Equation (3).

6.3.3 System Evaluation

Table 6.1 presents the performance of the proposed method of joint training with English to Khmer MT (Joint$_{\mathrm{en}}$) and French to Khmer MT (Joint$_{\mathrm{fr}}$). The proposed method outperformed the baseline Khmer ASR model in all experimented models.

Table 6.1: WER (%) of Khmer ASR on ECCC test set; ** and * indicates statistically significant difference with $p < 0.01$ and $p < 0.05$ from baseline, respectively.

Model	WER (%) of the Khmer		
	Baseline	Joint$_{\text{en}}$	Joint$_{\text{fr}}$
w/o augmentation	23.6	22.2**	22.3**
w/ Speed perturbation (SP)	22.2	21.1**	21.4**
w/ SpecAugment (SA)	21.8	20.5**	20.6**
w/ SP + SA	21.4	**19.5****	20.2**

Table 6.2: ASR improvement with proposed method for each group of speakers.

Speaker Group	Hour	Average WER (%)		
		Baseline	Joint$_{\text{en}}$	Relative
Witness	5	23.4	19.7**	15.8
Co-prosecutor	2	19.7	19.5	1.0
Civil-party	0.7	15.3	13.7**	10.5
Judge	0.3	17.0	17.1	-

Table 6.3: ASR improvement with the proposed method in accordance with baseline WER distribution.

Baseline WER (%)	# utterance	Average WER (%)		
		Baseline	Joint$_{\text{en}}$	Relative
$0-10$	$1,137$	4.5	5.3**	-
$10-20$	810	14.9	14.2*	4.7
$20-30$	538	25.8	23.6**	8.5
$30-40$	248	37.8	32.5**	14.0
$40-50$	165	49.4	43.3**	12.3
$50-100$	303	88.1	75.3**	14.5

All improvements are statistically significant ($p < 0.01$), but Joint$_{\text{en}}$ gave a larger improvement compared to Joint$_{\text{fr}}$. This is reasonable because English to Khmer MT has better performance. For the best performing model with SpecAugment (SA) and speed perturbation (SP), the proposed method reduced a large margin of WER by 1.9% (8.9% relative).

Regarding the best result for Joint$_{\text{en}}$, Table 6.2 shows the system performance in each group of speakers. The proposed method had a significant improvement on “Witness" and “Civil-party," reducing the WER by 15.8% and 10.45% relative, respectively. These speaker groups include the victims of the Khmer Rouge

Table 6.4: ASR improvement with proposed method in accordance with MT BLEU distribution (English-to-Khmer).

Baseline	#	Average WER (%)		
BLEU	utterance	Baseline	Joint$_{\text{en}}$	Relative
$0 - 10$	895	23.4	21.7^{**}	7.3
$10 - 20$	$1,205$	20.2	18.4^{**}	8.9
$20 - 30$	572	20.5	18.6^{**}	9.3
$30 - 40$	268	18.7	17.1^{*}	8.6
$40 - 50$	126	19.3	18.2	5.7
$50 - 100$	136	23.5	18.6^{**}	20.9

System	Output
Reference	នៅ |ថ្ងៃ |សាមសិប |ខែ |ដប់ |ពីរ |ឆ្នាំ |មួយពាន់ |ប្រាំបួនរយ |ចិតសិប |ប្រាំពីរ |ពេលនោះ |ខ្ញុំបាទ |កំពុង |តែ |ប្រមូល |ផល *At |Day |30 |Month |10 |2 |Year |1000 |900 |70 |7 |That time |I |Was |Doing |Collect |Outcome*
ASR	នៅ |ថ្ងៃ |សាមសិប |ខែ |ដប់ |ពីរ |ឆ្នាំ |មួយពាន់ |ប្រាំបួនរយ |ចិតសិប |ប្រាំពីរ |ពេលនោះ |ខ្ញុំបាទ |កំពុង |តែ |ប្រមូល |ផល |កសិកម្ម |នៅ *At |Day |30 |Month |10 |2 |Year |1000 |900 |70 |7 |That time |I |Was |Doing |Collect |Outcome |Farming |At*
MT	ថ្ងៃ |សាមសិប |ធ្នូ |មួយពាន់ |ប្រាំបួនរយ |ចិតសិប |ប្រាំពីរ |ដោយ |ពេល |ដែល |ខ្ញុំ |ទៅ |ធ្វើ |ស្រែ *Day |30 |December |1000 |900 |70 |7 |By |When |That |I |Go |Do |Field*
Joint$_{\text{en}}$	នៅ |ថ្ងៃ |សាមសិប |ខែ |ដប់ |ពីរ |ឆ្នាំ |មួយពាន់ |ប្រាំបួនរយ |ចិតសិប |ប្រាំពីរ |ពេលនោះ |ខ្ញុំបាទ |កំពុង |តែ |ប្រមូល |ផល |កសិកម្ម *At |Day |30 |Month |10 |2 |Year |1000 |900 |70 |7 |That time |I |Was |Doing |Collect |Outcome |Farming*

Figure 6.2: Examples of the comparison of all methods in the Khmer language, the *italic text* is the translated text into English.

regime, who are elderly and illiterate, thus had problems in their speech; they sometimes could not pronounce words correctly and exhibited disfluency and emotions in their speech during the trial. On the other hand, we did not obtain improvement for the group of "Judge" and "Co-prosecutor," who spoke fluently.

Table 6.3 presents the effectiveness of the proposed method in terms of the distribution of baseline WER. The worse the baseline ASR was, the more improvement is achieved with the proposed method. This trend is preferable in applications. In this case, the best improvement reduced the WER by 14.5% relative.

Table 6.4 presents the system performance in terms of the distribution of MT BLEU scores. It shows that a better MT performance generally results in a better improvement in the transcription of speech. This tendency is reasonable. With this result, the best MT BLEU score reduced the WER by 20.9% relative.

The baseline MT of English to Khmer has a BLEU score of 14.44, which is better than the translation quality of French to Khmer, the BLEU score of which

Table 6.5: WER (%) of speech transcription on Fisher-CallHome Spanish test set.

Test set	w/ SP		w/ SP+SA	
	Baseline	Joint$_{\text{en}}$	Baseline	Joint$_{\text{en}}$
Fisher				
- dev	24.2	24.0	23.1	**22.8**
- dev2	23.6	23.1	22.5	**22.3**
- test	21.5	21.7	20.8	**20.5**
CallHome				
- devtest	41.1	40.5	40.2	**39.5**
- evltest	41.4	41.0	39.6	**39.4**

is 10.54. This is reasonable because English sentences were used as the source in sentence alignment and segmentation to Khmer and French as described in Section 3.1.

Figure 6.2 presents an example of the output of the baseline ASR, MT, and the proposed method. We also investigated the possibility to combine the output hypotheses of ASR and MT. However, we found the hypotheses of MT are generally shorter (deletions of >30%) and much less accurate (substitutions of >30%) than the ASR hypotheses. This is because MT can have rephrasing without matching with speech (as annotated in "Blue text" in Figure 6.2) and less redundancy (no fillers, discourse markers). With this large difference between ASR and MT, we cannot combine the hypotheses of ASR and MT with ROVER. Moreover, it is not easy to combine two hypotheses with a simple voting mechanism. Instead, we propose a scheme to refer to MT for enhancing ASR hypotheses.

We also experimented the condition of replacing MT with ST, in which interpreters' speeches (e.g. English) are used for the input. In this setting, the WER was 20.0%, which is significantly improved the baseline but slightly lower than the originally proposed method using MT. This is due to the performance of the end-to-end ST. Since there is a limited number of interpreters in this corpus, separating ASR and MT is more practical.

6.3.4 Application to Fisher-CallHome-Spanish

To confirm that our proposed method can be generalized to other corpora, we conducted an experiment using Fisher-CallHome Spanish, which is a speech translation corpus of a conversational telephone speech in Spanish to English. It contains 160 hours of Spanish speech, corresponding transcription, and English translation text. The standard data preparation [58] was used, and the performances of Fisher-{dev, dev2, test} and CallHome-{devtest, evltest} were investigated.

The network architecture of this implementation followed the given recipe in the ESPnet. Texts in English and Spanish were stripped of all punctuation and were lower-cased. The BPE was then used to tokenize the text by using $1,000$ tokens per language, which means that we employed $2,000$ BPE tokens in total.

Table 6.5 presents the results of the baseline ASR model and our proposed method (Joint$_{\text{en}}$) in each evaluation set. In all test sets, the joint training of Spanish ASR and English to Spanish MT improved the transcription of Spanish speech. Especially, with SA and SP data augmentations, Joint$_{\text{en}}$ reduced the WER up to 0.7% absolute (1.7% relative) in “devtest" of CallHome. These results demonstrate the generalization of the proposed method.

6.4 Conclusions

In this Chapter, we have proposed a joint model of ASR and MT for improving the transcription of a low-resource language using a simultaneous translation from a rich-resource language. The proposed method was not only effective in improving the transcription in Khmer, but also in Spanish. The results demonstrate that translated knowledge is useful for enhancing the transcription of speech, especially for the lower-performance ASR with the higher translation quality of MT. This work is motivated from a language resource consideration, but in reality, the proposed approach may be helpful in acoustically challenging conditions. Additionally, this method can be applied to many settings of simultaneous transcription and translation in multi-lingual meetings or court proceedings.

Chapter 7

Conclusions

In this dissertation, we studied the incorporation of meta-information to enhance ASR performance in low-resource languages and also in very low-resource settings. We presented several novel techniques which leverage the meta-information to learn good representation for improving ASR performance when having access to only limited resources. This Chapter provides a succinct summary of the main contributions and then discusses certain limitations and future research directions.

7.1 Summary of Thesis Contributions

This dissertation overall addressed the *data scarcity* problem in low-resource languages. Firstly, we addressed the problem of bilingual alignment: (1) bilingual sentence alignment with low-resource languages, which lack good language processing toolkits, and (2) the missed timestamp information for speech-to-text alignment. We kept a large proportion of the original dataset by using monotonic sentence alignment and word-based distance calculation. We showed that this alignment requires the segmentation of the sentences in the source language only and will benefit similar datasets such as meetings, classroom lectures, and TV programs. Using rich-resource languages in the parallel corpus was effective for enhancing sentence alignment to low-resource languages. Moreover, using pretrained models of rich-resource languages was effective for tuning the performance of MTs in low-resource languages.

Secondly, we focused on the speaker-imbalance problem, which can occur in many scenarios including resource-rich languages. This challenge exists on a limited set of speakers in many cases such as TV programs, meetings, and court proceedings. Particularly in Chapter 4, we proposed the use of speaker information as speaker embedding. We showed that it was effective to enhance ASR performance when we clustered the minority speakers and used those speakers' information explicitly in the supervision of the speaker information for ASR.

Thirdly, we presented the effective method of finetuning a large-scale pretrained model in very low-resource settings. Specifically in Chapter 5, we proposed a two-step adaptation, which is composed of domain adaptation and language adaptation, to finetune a pretrained model with a limited dataset from one hour to 10 hours matched dataset. Our experiments showed that conducting domain adaptation first and then language adaptation was more effective. Using domain or language identification in MTL or adversarial learning was crucial for improved performance. Interestingly, it was possible to conduct adaptation with only a one-hour matched dataset and obtain almost saturated performance with a 5-hour matched dataset.

Lastly, we presented the incorporation of a translation knowledge of rich-resource language to enhance the transcription of a low-resource language in Chapter 6. With the assumption that the content of its back-translation is the same as the transcription of the original speech, we were able to formulate this framework as a joint process of ASR and MT/ST and then fused them together at the cross-attention of the decoder module. We showed that the translation knowledge of rich resources in parallel datasets effectively improved the ASR performance of low-resource languages.

7.2 Limitations and Future Research Directions

Finally, we describe several open problems regarding the limitations of the methods developed in this thesis and the directions for future research.

This study focused on non-streaming ASR systems. However, the transcription and translation of court proceedings are often needed in real-time. Thus, streaming ASR is suitable for this kind of scenario and this is the case for meetings and lectures. For low-resource languages or low-resource settings, finetuning the pretrained models consumes very small targeted datasets. Moreover, CTC is applicable for streaming and robust against long-form speech. Thus, it is not so difficult to apply the proposed method (Chpater 5) for simultaneous streaming ASR.

All proposed methods can enhance the ASR performance using only meta-information or auxiliary information. On the other hand, using the language models or large language models (LLM) is an effective method for rescoring and tuning the ASR performance since collecting text-only data is much easier compared to preparing speech-text datasets. Thus, finetuning the LLM on the target language should be explored for improving the ASR performance, especially in the limited targeted resources or low-resource languages.

Among the proposed methods, adding the labeled speech data of the target language was most effective for improving the ASR performance even in very low-resource settings. On the other hand, we should explore the effective use of unlabeled speech datasets to enhance ASR performance since collecting speech-only datasets is also much easier.

Bibliography

[1] P. Joshi, S. Santy, A. Budhiraja, K. Bali, and M. Choudhury, "The state and fate of linguistic diversity and inclusion in the NLP world," in *Proceedings of ACL*, pp. 6282–6293, Association for Computational Linguistics, July 2020.

[2] T. Sakai and S. Doshita, "The Phonetic Typewriter: Its Fundamentals and Mechanism," in *IFIP Congress 62,*, pp. 445–450, 1962.

[3] K.-F. Lee, *Automatic speech recognition: the development of the SPHINX system*, vol. 62. Springer Science & Business Media, 1988.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[5] A. Graves and N. Jaitly, "Towards End-To-End speech recognition with recurrent neural networks," in *International Conference on Machine Learning (ICML)*, pp. 1764–1772, 2014.

[6] A. Graves, S. Fernandez, F. Gomez, and J. Shmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of ICML*, 2006.

[7] A. Graves, "Sequence Transduction with Recurrent Neural Networks," *ICML Representation Learning Workshop*, Nov. 2012.

[8] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proceedings of IEEE-ICASSP*, 2016.

[9] A. Vaswani, N. S. abd Niki Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proceedings of*

NeurlPS, 2017.

[10] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented Transformer for Speech Recognition,” in *Proceedings of Interspeech*, 2020.

[11] U. Shrawankar and V. Thakare, “Adverse conditions and asr techniques for robust speech user interface,” 2013.

[12] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, “Automatic speech recognition for under-resourced languages: A survey,” *Speech Communication*, vol. 56, pp. 85–100, 2014.

[13] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, “Moses: Open source toolkit for statistical machine translation,” in *Proceedings of ACL*, 2007.

[14] T. Kiss and J. Strunk, “Unsupervised multilingual sentence boundary detection,” *Comput. Linguist.*, vol. 32, p. 485–525, Dec. 2006.

[15] B. Thompson and P. Koehn, “Vecalign: Improved sentence alignment in linear time and space,” in *Proceedings of EMNLP-IJCNLP*, (Hong Kong, China), pp. 1342–1348, Association for Computational Linguistics, Nov. 2019.

[16] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, “Montreal forced aligner: Trainable text-speech alignment using kaldi,” in *Proceedings of Interspeech*, 2017.

[17] K. Matsuura, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, “Speech corpus of Ainu folklore and end-to-end speech recognition for Ainu language,” in *Proceedings of LREC*, pp. 2622–2628, May 2020.

[18] T. Schultz and A. Waibel, “Language independent and language adaptive acoustic modeling for speech recognition,” *Speech Communication*, vol. 35, no. 1-2, pp. 31–51, 2001.

[19] A. Stolcke, F. Grezl, M.-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, “Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons,” in *Proceedings of ICASSP*, vol. 1, pp. I–I, 2006.

[20] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proceedings of ICASSP*, pp. 7304–7308, 2013.

[21] X. Li, S. Dalmia, A. W. Black, and F. Metze, "Multilingual Speech Recognition with Corpus Relatedness Sampling," in *Proceedings of Interspeech*, pp. 2120–2124, 2019.

[22] S. Dalmia, R. Sanabria, F. Metze, and A. W. Black, "Sequence-based multi-lingual low resource speech recognition," in *Proceedings of ICASSP*, p. 4909–4913, IEEE Press, 2018.

[23] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *Proceedings of ASRU*, pp. 265–271, 2017.

[24] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. J. Moreno, E. Weinstein, and K. Rao, "Multilingual speech recognition with a single end-to-end model," *Proceedings of ICASSP*, pp. 4904–4908, 2018.

[25] S. Dalmia, R. Sanabria, F. Metze, and A. W. Black, "Sequence-based multi-lingual low resource speech recognition," *Proceedings of ICASSP*, pp. 4909–4913, 2018.

[26] J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. R. Mallidi, N. Yalta, M. Karafiát, S. Watanabe, and T. Hori, "Multilingual sequence-to-sequence speech recognition: Architecture, transfer learning, and language modeling," *Proceedings of SLT*, pp. 521–527, 2018.

[27] H. Inaguma, J. Cho, M. K. Baskar, T. Kawahara, and S. Watanabe, "Transfer learning of language-independent end-to-end asr with language model fusion," *Proceedings of ICASSP*, pp. 6096–6100, 2018.

[28] K. Soky, S. Li, T. Kawahara, and S. Seng, "Multi-lingual transformer training for khmer automatic speech recognition," in *Proceedings of APSIPA ASC*, pp. 1893–1896, 2019.

[29] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot trans-

lation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.

[30] T.-L. Ha, J. Niehues, and A. Waibel, "Toward multilingual neural machine translation with universal encoder and decoder," in *Proceedings of IWSLT*, dec 2016.

[31] T.-L. Ha, J. Niehues, and A. Waibel, "Effective strategies in zero-shot neural machine translation," in *Proceedings of IWSLT*, pp. 105–112, Dec. 2017.

[32] H. Inaguma, K. Duh, T. Kawahara, and S. Watanabe, "Multilingual end-to-end speech translation," *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 570–577, 2019.

[33] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "MuST-C: a Multilingual Speech Translation Corpus," in *Proceedings of NAACL-HLT*, 2019.

[34] M. Gales and S. Young, *Application of Hidden Markov Models in Speech Recognition*. Now Foundations and Trends, 2008.

[35] L. Deng, D. Yu, and G. Hinton, "Deep learning for speech recognition and related applications," in *NIPS workshop*, 2009.

[36] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proceedings of Interspeech*, pp. 437–440, 2011.

[37] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: first results," in *Advances in Neural Information Processing Systems (NIPS) Workshop on Deep Learning*, 2014.

[38] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-Based Models for Speech Recognition," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 577–585, 2015.

[39] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and lstm encoder decoder models for asr," in *Proceedings of ASRU*, pp. 8–15, 2019.

[40] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end

speech recognition using multi-task learning," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4835–4839, 2016.

[41] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in Joint CTC-Attention Based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM," in *Proceedings of Interspeech*, 2017.

[42] T. Hori, S. Watanabe, and J. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proceedings of ACL*, pp. 518–529, 2017.

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

[44] S. Zhou, L. Dong, S. Xu, and B. Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," in *Proceedings of Interspeech*, pp. 791–795, 2018.

[45] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration," in *Proceedings of Interspeech*, pp. 1408–1412, 2019.

[46] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to machine translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, 1990.

[47] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of ACL*, pp. 1700–1709, Oct. 2013.

[48] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[49] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of ACL*, pp. 103–111, Oct. 2014.

[50] H. Ney, "Speech translation: coupling of recognition and translation," in *Proceedings of ICASSP*, vol. 1, pp. 517–520 vol.1, 1999.

[51] A. Bérard, O. Pietquin, L. Besacier, and C. Servan, “Listen and Translate: A Proof of Concept for End-to-End Speech-to-Text Translation,” in *NIPS Workshop on end-to-end learning for speech and audio processing*, (Barcelona, Spain), Dec. 2016.

[52] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, “Sequence-to-sequence models can directly translate foreign speech,” in *Proc. Interspeech 2017*, pp. 2625–2629, 2017.

[53] A. Berard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, “End-to-end automatic speech translation of audiobooks,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6224–6228, 2018.

[54] P. Bahar, T. Bieschke, and H. Ney, “A comparative study on end-to-end speech to text translation,” *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 792–799, 2019.

[55] K. Soky, S. Li, M. Mimura, C. Chu, and T. Kawahara, “On the use of speaker information for automatic speech recognition in speaker-imbalanced corpora,” in *Proceedings of APSIPA ASC*, 2021.

[56] R. Caruana, “Multitask learning,” *Mach. Learn.*, vol. 28, p. 41–75, jul 1997.

[57] Y. S. Abu-Mostafa, “Learning from hints in neural networks,” *Journal of Complexity*, vol. 6, no. 2, pp. 192–198, 1990.

[58] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, “Improved speech-to-text translation with the Fisher and Callhome Spanish–English speech translation corpus,” in *Proceedings of IWSLT*, 2013.

[59] J. Iranzo-Sánchez, J. A. Silvestre-Cerdà, J. Jorge, N. Roselló, A. Giménez, A. Sanchis, J. Civera, and A. Juan, “Europarl-ST: A Multilingual Corpus for Speech Translation of Parliamentary Debates,” in *Proceedings of IEEE-ICASSP*, 2020.

[60] E. Salesky, M. Wiesner, J. Bremerman, R. Cattoni, M. Negri, M. Turchi, D. Oard, and M. Post, “The multilingual tedx corpus for speech recognition and translation,” vol. abs/2102.01757, pp. 3655–3659, 2021.

[61] F. J. Och and H. Ney, “Statistical multi-source translation,” in *Proceedings of*

MT Summit, 2001.

[62] E. Garmash and C. Monz, "Ensemble learning for multi-source neural machine translation," in *Proceedings of COLING*, pp. 1409–1418, 2016.

[63] B. Zoph and K. Knight, "Multi-source neural translation," in *Proceedings of NAACL-HLT*, pp. 30–34, June 2016.

[64] R. Dabre, F. Cromieres, and S. Kurohashi, "Enabling multi-source neural machine translation by concatenating source sentences in multiple languages," *Proceedings of MT Summit*, 2017.

[65] Y. Nishimura, K. Sudoh, G. Neubig, and S. Nakamura, "Multi-source neural machine translation with data augmentation," *arXiv preprint arXiv:1810.06826*, 2018.

[66] Y. Nishimura, K. Sudoh, G. Neubig, and S. Nakamura, "Multi-source neural machine translation with missing data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 569–580, 2020.

[67] Z. Lu, X. Li, Y. Liu, C. Zhou, J. Cui, B. Wang, M. Zhang, and J. Su, "Exploring multi-stage information interactions for multi-source neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[68] X. Zhou, E. Yılmaz, Y. Long, Y. Li, and H. Li, "Multi-Encoder-Decoder Transformer for Code-Switching Speech Recognition," in *Proceedings of Interspeech*, pp. 1042–1046, 2020.

[69] Y.-F. Cheng, H.-S. Lee, and H.-M. Wang, "AlloST: Low-Resource Speech Translation Without Source Transcription," in *Proceedings of Interspeech*, pp. 2252–2256, 2021.

[70] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proceedings of IEEE-ASRU*, 1997.

[71] F. E. Huffman, "Cambodian System of Writing and Beginning Reader," *Yale University Press*, 1970.

[72] T. Nakazawa, N. Doi, S. Higashiyama, C. Ding, R. Dabre, H. Mino, I. Goto, W. P. Pa, A. Kunchukuttan, Y. Oda, S. Parida, O. Bojar, and S. Kurohashi,

"Overview of the 6th workshop on Asian translation," in *Proceedings of ACL*, 2019.

[73] F. Braune and A. Fraser, "Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora," in *Proceedings of COLING*, (Beijing, China), pp. 81–89, Coling 2010 Organizing Committee, 2010.

[74] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *Proceedings of NAACL-HLI*, (Atlanta, Georgia), pp. 644–648, Association for Computational Linguistics, 2013.

[75] B. Thompson and P. Koehn, "Vecalign: Improved sentence alignment in linear time and space," in *Proceedings of EMNLP-IJCNLP*, (Hong Kong, China), pp. 1342–1348, Association for Computational Linguistics, 2019.

[76] V. Chea, Y. K. Thu, C. Ding, M. Utiyama, A. Finch, and E. Sumita, "Khmer word segmentation using conditional random fields," in *Proceedings of Khmer Natural Language Processing (KNLP)*, 2015.

[77] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, "Creating corpora for speech-to-speech translation," in *Proceedings of EUROSPEECH*, pp. 381–384, 2003.

[78] B. Marie, H. Kaing, A. M. Mon, C. Ding, A. Fujita, M. Utiyama, and E. Sumita, "Supervised and unsupervised machine translation for Myanmar-English and Khmer-English," in *Proceedings of the 6th Workshop on Asian Translation*, (Hong Kong, China), pp. 68–75, Association for Computational Linguistics, 2019.

[79] J. Niehues, E. Salesky, M. Turchi, and M. Negri, "Tutorial proposal: End-to-end speech translation," in *Proceedings of ACL*, pp. 10–13, Apr. 2021.

[80] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in *Proceedings of Interspeech*, 2018.

[81] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of ACL*, (Berlin, Germany), pp. 1715–1725, Association for Computational Linguistics, 2016.

[82] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio Augmentation for

Speech Recognition," in *Proceedings of Interspeech*, 2015.

[83] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proceedings of Interspeech*, 2019.

[84] K. Soky, M. Mimura, T. Kawahara, S. Li, C. Ding, C. Chu, and S. Sam, "Khmer Speech Translation Corpus of the Extraordinary Chambers in the Courts of Cambodia (ECCC)," in *Proceedings of O-COCOSDA*, 2021.

[85] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of ACL*, (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, July 2002.

[86] H. Inaguma, S. Kiyono, K. Duh, S. Karita, N. Yalta, T. Hayashi, and S. Watanabe, "ESPnet-ST: All-in-one speech translation toolkit," in *Proceedings of ACL*, (Online), pp. 302–311, Association for Computational Linguistics, 2020.

[87] O. Siohan, B. Ramabhadran, and B. Kingsbury, "Constructing ensembles of asr systems using randomized decision trees," in *Proceedings. ICASSP*, vol. 1, pp. I/197–I/200 Vol. 1, 2005.

[88] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," *CoRR*, vol. abs/1511.06066, 2015.

[89] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, (Austin, Texas), pp. 1568–1575, Association for Computational Linguistics, Nov. 2016.

[90] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

[91] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," in *Proceedings of NAACL-HLT*, pp. 58–68, 2019.

[92] Y. Kim, P. Petrov, P. Petrushkov, S. Khadivi, and H. Ney, "Pivot-based

transfer learning for neural machine translation between non-English languages," in *Proceedings of EMNLP-IJCNLP*, pp. 866–876, Nov. 2019.

[93] M. Delcroix and S. Watanabe and A. Ogawa and S. Karita and T. Nakatani, "Auxiliary Feature Based Adaptation of End-to-End ASR Systems," *Proceedings of INTERSPEECH*, vol. 2018-September, pp. 2444–2448, 2018.

[94] Z. Fan, J. Li, S. Zhou, and B. Xu, "Speaker-aware speech-transformer," in *Proceedings of (ASRU)*, pp. 222–229, 2019.

[95] Y. Zhao, C. Ni, C.-C. Leung, S. Joty, E. S. Chng, and B. Ma, "Speech Transformer with Speaker Aware Persistent Memory," in *Proceedings of Interspeech*, pp. 1261–1265, 2020.

[96] V. M. Shetty, M. S. Mary N. J, and S. Umesh, "Investigation of Speaker-adaptation methods in Transformer based ASR," *CoRR*, vol. abs/2008.03247, 2020.

[97] L. Sari, N. Moritz, T. Hori, and J. Le Roux, "Unsupervised Speaker Adaptation Using Attention-Based Speaker Memory For End-To-End ASR," in *Proceedings of ICASSP*, pp. 7384–7388, IEEE, Apr. 2020.

[98] Z. Tang, L. Li, and D. Wang, "Multi-task Recurrent Model for Speech and Speaker Recognition," in *Proceedings of APSIPA*, pp. 1–4, 2016.

[99] N. Kanda, Y. Gaur, X. Wang, Z. Meng, Z. Chen, T. Zhou, and T. Yoshioka, "Joint Speaker Counting, Speech Recognition, and Speaker Identification for Overlapped Speech of Any Number of Speakers," in *Proceedings of Interspeech*, ISCA, October 2020.

[100] N. Kanda, X. Chang, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "Investigation of End-To-End Speaker-Attributed ASR for Continuous Multi-Talker Recordings," in *Proceedings of SLT*, IEEE, January 2021.

[101] Y. Ganin and V. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1180–1189, 07–09 Jul 2015.

[102] Y. Shinohara, "Adversarial Multi-Task Learning of Deep Neural Networks for Robust Speech Recognition," in *Proceedings of Interspeech*, 2016.

[103] T. Tsuchiya, N. Tawara, T. Ogawa, and T. Kobayashi, "Speaker Invariant Fea-

ture Extraction for Zero-Resource Languages with Adversarial Learning," in *Proceedings of ICASSP*, pp. 2381–2385, 2018.

[104] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong, and B. Juang, "Speaker-Invariant Training Via Adversarial Learning," in *Proceedings of ICASSP*, pp. 5969–5973, 2018.

[105] Z. Meng, J. Li, and Y. Gong, "Adversarial speaker adaptation," in *Proceedings of ICASSP*, pp. 5721–5725, 2019.

[106] L. El Shafey, H. Soltau, and I. Shafran, "Joint Speech Recognition and Speaker Diarization via Sequence Transduction," in *Proceedings of Interspeech*, pp. 396–400, 2019.

[107] S. Li, D. Raj, X. Lu, P. Shen, T. Kawahara, and H. Kawai, "Improving Transformer-Based Speech Recognition Systems with Compressed Structure and Speech Attributes Augmentation," in *Proceedings of Interspeech*, 2019.

[108] H. Henry Mao, S. Li, J. McAuley, and G. W. Cottrell, "Speech Recognition and Multi-Speaker Diarization of Long Conversations," in *Proceedings of Interspeech*, pp. 691–695, 2020.

[109] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proceedings of ICASSP*, pp. 5206–5210, 2015.

[110] A. Rousseau, P. Deléglise, and Y. Estève, "TED-LIUM: an automatic speech recognition dedicated corpus," in *Proceedings of LREC*, (Istanbul, Turkey), pp. 125–129, European Language Resources Association (ELRA), May 2012.

[111] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous Speech Corpus of Japanese," in *Proceedings of LREC*, May 2000.

[112] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *Proceedings of ICASSP*, 2018.

[113] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *preprint arXiv:1607.06450*, 2016.

[114] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework

for Self-Supervised Learning of Speech Representations," in *Proceedings of NeurlPS*, vol. 33, pp. 12449–12460, 2020.

[115] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *CoRR*, vol. abs/1910.05453, 2019.

[116] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised Cross-Lingual Representation Learning for Speech Recognition," in *Proceedings of Interspeech*, pp. 2426–2430, 2021.

[117] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," in *Proceedings of Interspeech*, pp. 2278–2282, 2022.

[118] R. Alec, W. K. Jong, X. Tao, B. Greg, M. Christine, and S. Ilya, "Robust speech recognition via large-scale weak supervision.," 2022.

[119] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of ACL*, (Minneapolis, Minnesota), pp. 4171–4186, June 2019.

[120] J. Li, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, April 2022.

[121] M. K. Baskar, T. Herzig, D. Nguyen, M. Diez, T. Polzehl, L. Burget, and J. Černocký, "Speaker adaptation for Wav2vec2 based dysarthric ASR," in *Proceedings of Interspeech*, pp. 3403–3407, 2022.

[122] N. Vaessen and D. A. Van Leeuwen, "Fine-tuning wav2vec2 for speaker recognition," in *Proceedings of ICASSP*, pp. 7967–7971, 2022.

[123] A. Tjandra, D. G. Choudhury, F. Zhang, K. Singh, A. Conneau, A. Baevski, A. Sela, Y. Saraf, and M. Auli, "Improved language identification through cross-lingual self-supervised learning," in *Proceedings of ICASSP*, pp. 6877–6881, 2022.

[124] X. Cai, J. Yuan, R. Zheng, L. Huang, and K. Church, "Speech Emotion Recognition with Multi-Task Learning," in *Proceedings of Interspeech*, pp. 4508–4512, 2021.

[125] C. Yi, J. Wang, N. Cheng, S. Zhou, and B. Xu, "Applying wav2vec2.0 to speech recognition in various low-resource languages," *ArXiv*, vol. abs/2012.12121, 2020.

[126] K. D. N, P. Wang, and B. Bozza, "Using Large Self-Supervised Models for Low-Resource Speech Recognition," in *Proceedings of Interspeech*, pp. 2436–2440, 2021.

[127] C. Yi, S. Zhou, and B. Xu, "Efficiently fusing pretrained acoustic and linguistic encoders for low-resource speech recognition," *IEEE Signal Processing Letters*, vol. 28, pp. 788–792, 2021.

[128] K. Fatehi, M. Torres Torres, and A. Kucukyilmaz, "ScoutWav: Two-Step Fine-Tuning on Self-Supervised Automatic Speech Recognition for Low-Resource Environments," in *Proceedings of Interspeech*, pp. 3523–3527, 2022.

[129] K. Soky, M. Mimura, T. Kawahara, C. Chu, S. Li, C. Ding, and S. Sam, "TriECCC: Trilingual Corpus of the Extraordinary Chambers in the Courts of Cambodia for Speech Recognition and Translation Studies," *International Journal of Asian Language Processing*, vol. 31, no. 03&04, p. 2250007, 2022.

[130] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen, "The United Nations parallel corpus v1.0," in *Proceedings of LREC*, pp. 3530–3534, 2016.

[131] D. Macháček, M. Žilinec, and O. Bojar, "Lost in Interpreting: Speech Translation from Source or Interpreter?," in *Proceedings of Interspeech*, 2021.

[132] M. Paulik, S. Stuker, C. Fugen, T. Schultz, T. Schaaf, and A. Waibel, "Speech translation enhanced automatic speech recognition," in *Proceedings of IEEE-ASRU*, 2005.

[133] M. Paulik, C. Fügen, S. Stüker, T. Schultz, T. Schaaf, and A. Waibel, "Document driven machine translation enhanced asr," in *Proceedings of Eurospeech*, 2005.

[134] S. Khadivi, A. Zolnay, and H. Ney, "Automatic text dictation in computer-assisted translation," in *Proceedings of Eurospeech*, 2005.

[135] S. Khadivi, R. Zens, and H. Ney, "Integration of speech to computer-assisted translation using finite-state automata," in *Proceedings of COLING/ACL*, 2006.

[136] E. Vidal, F. Casacuberta, L. Rodríguez, J. Civera, and C. D. M. Hinarejos, “Computer-assisted translation using speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2006.

[137] S. Khadivi and H. Ney, “Integration of speech recognition and machine translation in computer-assisted translation,” *IEEE TASLP*, 2008.

[138] A. Reddy and R. C. Rose, “Integration of statistical models for dictation of document translations in a machine-aided human translation task,” *IEEE TASLP*, 2010.

[139] P. Wang, T. N. Sainath, and R. J. Weiss, “Multitask Training with Text Data for End-to-End Speech Recognition,” in *Proceedings of Interspeech*, pp. 2566–2570, 2021.

[140] B. Yusuf, A. Gandhe, and A. Sokolov, “USTED: Improving ASR with a Unified Speech and Text Encoder-Decoder,” 2022.

[141] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, “Pushing the limits of semi-supervised learning for automatic speech recognition,” 2020.

List of Publications

Refereed International Journal Papers

1) Kak Soky, Sheng Li, Chenhui Chu, Tatsuya Kawahara: Finetuning Pretrained Model with Embedding of Domain and Language Information for ASR of Very Low-Resource Settings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Under Review. → **Chapter 5**
2) Kak Soky, Masato Mimura, Tatsuya Kawahara, Chenhui Chu, Sheng Li, Chenchen Ding, Sethserey Sam: TriECCC: Trilingual Corpus of the Extraordinary Chambers in the Courts of Cambodia for Speech Recognition and Translation Studies. *International Journal of Asian Language Processing* Vol. 31, no. 03&04 2250007, 2022. → **Chapter 3**

Refereed International Conference Papers

1) Kak Soky, Sheng Li, Chenhui Chu, Tatsuya Kawahara: Domain and Language Adaptation using Heterogeneous Datasets for Wav2vec2.0-based Speech Recognition of Low-resource Language. In Proceedings of *ICASSP*, 2023. → **Chapter 5**
2) Kak Soky, Zhou Gong, Sheng Li: NICT-Tib1: A Public Speech Corpus of Lhasa Dialect for Benchmarking Tibetan Language Speech Recognition Systems. In Proceedings of *O-COCOSDA*, 2022.
3) Kak Soky, Sheng Li, Masato Mimura, Chenhui Chu, Tatsuya Kawahara: Leveraging Simultaneous Translation for Enhancing Transcription of Low-resource Language via Cross Attention Mechanism. In Proceedings of *INTERSPEECH*, pp.1362–1366, 2022. → **Chapter 6**

4) Kak Soky, Masato Mimura, Tatsuya Kawahara, Sheng Li, Chenchen Ding, Chenhui Chu, Sethserey Sam: Khmer Speech Translation Corpus of the Extraordinary Chambers in the Courts of Cambodia (ECCC). In Proceedings of *O-COCOSDA*, pp. 122-127, 2021.
5) Kak Soky, Sheng Li, Masato Mimura, Chenhui Chu, Tatsuya Kawahara: On the Use of Speaker Information for Automatic Speech Recognition in Speaker-imbalanced Corpora. In Proceedings of *APSIPA ASC*, pp. 433-437, 2021. → **Chapter 4**
6) Kak Soky, Sheng Li, Tatsuya Kawahara, Sopheap Seng: Multi-lingual Transformer Training for Khmer Automatic Speech Recognition. In Proceedings of *APSIPA ASC*, pp. 1893-1896, 2019.

Technical Reports

1) Kak Soky, Sheng Li, Chenhui Chu, Tatsuya Kawahara: Domain and Language Adaptation of Large-scale Pretrained Model for Speech Recognition of Low-resource Language, *IEICE-SP*, 2022.
2) Kak Soky, Sheng Li, Masato Mimura, Chenhui Chu, Tatsuya Kawahara: Comparison of End-to-End Models for Joint Speaker and Speech Recognition. *IEICE-SP*, 2021.