DISSERTATION

DOMAIN ADAPTATION OF ASR MODELS WITH AUGMENTED DATA USING RE-RECORDED SPEECH IN REAL ENVIRONMENT

June 2023

Shizuoka University
Graduate School of Science and Technology
Educational Division

S M RAUFUN NAHAR

DECLARATION

I declare that the work in this dissertation titled “Domain Adaptation of ASR Models with Augmented Data using Re-recorded Speech in Real Environment", the work has been carried out by the candidate to fulfill one of the requirements of Shizuoka University for the degree of Doctor of Philosophy. The tasks performed with the assistance of others are indicated in the specific references in this dissertation. No part of this dissertation has been submitted anywhere else for any other academic degree.

S M Raufun Nahar
June 2023

Dedicated to my

PARENTS

with gratitude and love

ACKNOWLEDGMENTS

This dissertation is the precious result of years of study, learning and hard works. This work would not have been successful without technical, mental and moral support of respectful and dear individuals around me and I thank the Almighty Allah for giving me the strength to carry on day by day.

First and foremost, I express my gratitude to my respected supervisor Prof. Dr. Atsuhiko Kai for accepting me as his disciple and providing with all the necessary supports to continue the research patiently. Without his sincere guidance and unconditional moral support, it would not have been possible to produce any fruitful conclusion in the end.

I also say my thanks to Prof. Dr. Nakagawa for his valuable opinion in realizing the content of this dissertation. I am also grateful to the reviewers of the defense Prof. Dr. Kiriyama, Prof. Dr. Morita, Prof. Dr. Ohashi, and Prof. Dr. Nishida for their valuable comments which led the dissertation paper to its final direction.

The re-recorded Mobile LTE data used in Chapter 3 and Chapter 4 are recorded in collaboration with Mr. Takahiro Kunisaki of Nextgen Inc. The data are recorded at the receiving terminal of call center in Tokyo. I thank him for the cooperation in data acquisition.

I would also like to thank all of my lab mates from the bottom of my heart for creating such warm and enthusiastic environment in the laboratory which gave me strength to carry on with my research activity and everyday life. Also their presence

has helped me to understand the trend and way of education throughout the years. I especially thank Shogo Miwa and Rino Suzuki for helping me to perform experiments despite having his own research schedule.

The person I am most thankful to besides is my best friend Dr. Rebeka Sultana, who has supported me unconditionally with love and care no less than my family in this foreign land.

I am also thankful for Shizuoka University International Exchange Fund scholarship and Japan Student Services Organization (JASSO) scholarship for stipends from time to time to help me cope with financial hardship and continue my study.

I would like to express my gratitude to my Supervisor from Bachelors, Prof. Dr. MD. Ekramul Hamid and Prof. Dr. Shamim Ahmad for their inspiration and encouragement to pursue higher education.

Last but not the least, I am thankful to my parents for never giving up on the hope and belief in me and the sacrifices they have made to see me successful in my life. Therefore, I dedicate my success to my parents.

ABSTRACT

Automatic Speech Recognition (ASR) is a vast field of AI. In this field, numerous research has been performed and published on improving ASR technologies. ASR technologies help machines to recognize human speech and further process it to produce understandable results so that it can contribute to replace the menial tasks like taking notes or help post processing the text data. Though the success of the tasks mentioned above depend greatly on the success of the post processing part, still the most crucial of all is for the ASR to be able to recognise the speech correctly.

The most recent approaches of ASR involve deep learning-based approaches where the speech features and a dictionary of words or characters are used to train acoustic models and language models explicitly in case of deep neural network (DNN)-hidden Markov model (HMM) hybrid acoustic models or implicitly in cases of end-to-end (E2E) ASR models. Ideally, these models are trained using a large amount of training data. Traditionally, it is easy to acquire such models or dataset to train ASR models in quiet environment (clean) domain. However, depending on the situation of the target domain, it is often difficult to acquire such data with perfect transcription. In limited data situation, fine-tuning a pre-trained ASR model to adapt it to the target domain is quite common. Even so, to fine-tune an ASR, it is necessary to have speech samples with transcription.

In this research, to solve the problem of domain adaptation with limited data, the fine-tuning approach of domain adaptation is adopted. In this research, two recording environments are considered as target domain (mobile telephone, classroom) for validating the proposed method. By re-recording clean data in target domain,

recording and transcription cost could be reduced significantly. However, the re-recorded data suffer from some unintentional problems such as, temporal misalignment from packet loss and restoration in case of mobile telephone, and variability in recording quality in case of long-term re-recording condition. Therefore, though re-recording transcribed clean data may seem to be an easy solution, depending on real-world problems, it could be difficult to achieve desired performance just by it.

One of the goals in this research is to augment data for target domain by using clean and re-recorded data pairs to train a DNN-based regression model to map clean features to target domain features and generate features with target domain characteristics (feature transformation). In this thesis, a frame-by-frame approach of training feed-forward DNN is proposed. Due to the frame-by-frame computation policy, the misaligned telephone speech could not have been used. Therefore, a geometric approach of correcting misalignment and a filtering method to filter out internally misaligned utterances are adopted. However, it results in reduction of usable utterances significantly (34% of the re-recorded utterances could be used). The DNN for feature transformation is trained using a small portion of the filtered and aligned speech features as target with the clean counterpart as input. This DNN is used to perform data augmentation by generating re-recording-like data. The generated features are used with the re-recorded and augmented clean features to perform domain adaptation by fine-tuning ASR models with them.

The effect of domain adaptation is observed independently to the ASR models such as, two state-of-the-art models, time delay neural network (TDNN)-based DNN-HMM acoustic model and hybrid CTC/Transformer-based E2E ASR model. Both of the model structures encompass mechanism to handle temporal aspect of speech data. By training the ASRs with augmented large clean dataset, it is possible to get closer to the target domain. By fine-tuning them with even small amount of target domain data along with DNN-based augmented features, it has been possible to achieve 27% character error rate reduction (CERR) for telephone speech using LTE network for DNN-HMM hybrid acoustic model and 36.4% for hybrid CTC/Transformer E2E

ASR model.

Despite the success of the method mentioned above with telephone speech, the performance for classroom wireless pin mic recording was not satisfactory since the recording levels (dB) are different for different recording sessions. Also, the number of recordings is very small, which makes it difficult to train the DNN-based feature transformation model with mixed condition. Therefore, another state-of the-art, a self-supervised learning (SSL)-based E2E approach is investigated for domain adaptation that incorporates multi-lingual pre-trained self-supervised model and fine-tune it with large Japanese data followed by small amount of session-dependent target domain data. This approach helps us achieve character error rate of 16.5% for the first session and 17.9% for the second session (16.6% and 22.2% respectively for the end-to-end approach mentioned in the paragraph above) for the classroom wireless pin mic recordings. It shows that the SSL E2E model is robust for domain adaptation by fine-tuning.

CONTENT **Page**

LIST OF FIGURES

FIGURE **Page**

LIST OF TABLES

TABLE **Page**

CHAPTER 1

INTRODUCTION

This chapter provides the introduction of this dissertation. It introduces the background of automatic speech recognition system in Section 1.1, challenges of speech recognition in real world environment in Section 1.2, the motivation behind the research and the contribution in Section 1.3. Finally, it introduces the structure of this dissertation in Section 1.4.

1.1 Background of Automatic Speech Recognition (ASR)

The field of Artificial Intelligence (AI) has started flourishing since mid twentieth century. Since then, it has been in the center of interest of scientists of all disciplines. More than seventy years later, the journey of AI seem to have taken leap to another dimension. Now, AI is tightly knitted to our everyday life in the contemporary world. AI has made life easier for people in many ways, such as, smart voice assistants, driving automation, and so on. In the heart of every AI research, the understanding and implementation of human brain functionality is involved. If we compare AI to its equivalent human model, the field of computer vision is comparable to seeing and the functionality of eyes, speech recognition is equivalent to the task of hearing and the functionality of ears and speech synthesis is comparable with speaking and imitating the functionality of using vocal organs to express not only spoken terms

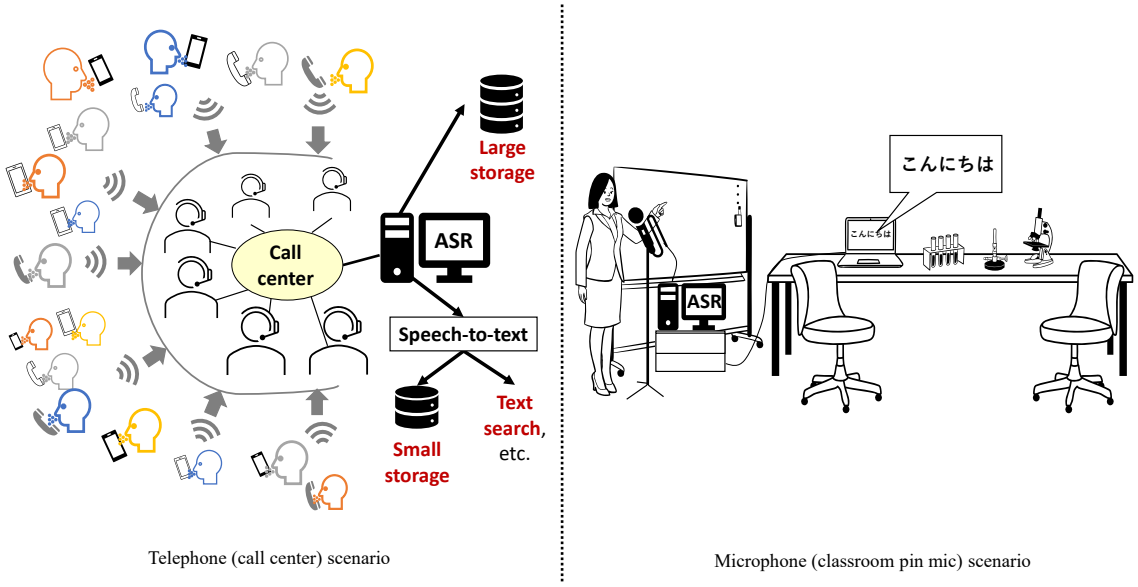


Figure 1.1: Example of real world domains.

but also emotions. There are numerous other human activities and functionalities those are currently being researched to be imitated by AI with the best interest of humanity. With these attempts being successful, the life of people in need, such as, people who are hearing impaired, visually impaired or physically challenged can be easier and rather normal comparing to the past.

The study of automatic speech recognition (ASR) is a vast area of AI. ASR is important for AI as much as hearing is for humans. ASR enables the machines and software to understand the representation of sound computationally. The representations can be fragments of a second of speech data also known as a frame, a word or a sequence of words. Depending on the purpose, the ASR can be equipped to perform tasks like converting speech to text, a well known application which enables us to transcribe of meetings, conversations; automatic caption generation for people with hearing impairment and so on. Figure 1.1 shows automatic speech recognition in telephone domain and classroom domain. telephone speech recognition in call centers can reduce storage problem and increase usability of speech data in text form. Speech-to-text technology in classroom can help as learning assistant and be helpful for students with hearing impairment, etc.

ASR models explicitly or implicitly involve two vital components. One is the language model, which deals with the linguistic aspects of speech, and the other is the acoustic model to process acoustic aspects of speech. Conventionally, acoustic models are trained using large-scale clean data recorded in ideal condition – noise free environment with high quality close-talking microphone, and so on. These acoustic models perform well for recognizing clean speech data. However, they do not perform as effectively for data which are recorded in conditions with numerous variables which can affect the quality of the resulting speech data to be processed. The recording condition of data can range from simple environment like a classroom equipped with a regular wireless pin-mic to complex environment like telephone channel recording which can involve countless types of handsets, transmission network, background noise at the location of the caller and receiver, etc. Therefore, it is necessary to improve the performance of acoustic models for real environment speech data in terms of cost-effectiveness and convenience.

1.2 Challenges of Speech Recognition in Real Environment

The general theory in speech recognition is that acoustic models and language models or the end-to-end ASR models need to be trained with relevant and adequate amount of data to achieve the best performance [1–4]. However, the task of real world speech recognition is rather difficult to date. Real-world recording media are bound to vary in terms of recording conditions, transmission channels, etc. The performance of ASR models degrade significantly when evaluated in mismatched new domain. It is important to prepare enough data for each condition with proper transcription for training an acoustic model appropriate for the environments accordingly. However, acquiring and preparing such data can be cost expensive and time consuming. Figure 1.2 shows the challenges of real world speech recognition. Therefore, data augmentation, domain adaptive technologies, etc. are adopted to solve this problem.

The number of possibility of variability in domain is uncountable. The domains can vary in different aspects. Two major aspects among them are acoustic and linguistic

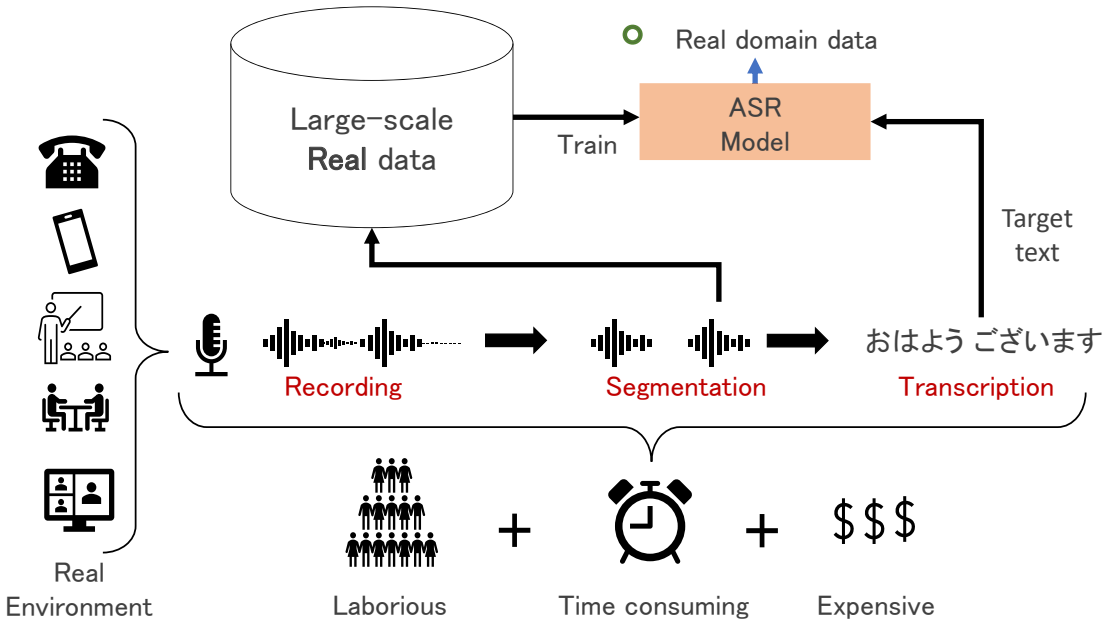


Figure 1.2: Challenge of speech recognition in real world domains.

aspects. Acoustic aspect contains several factors, such as speaker characteristics, transmission channels (e. g. microphone, room acoustics, etc.), and so on. Linguistic aspect contains languages, topics, social background and so on. Therefore, every scenario in a domain has its own characteristics. Some of them are common, some of them are completely different. Examples of real world scenario may include telephone speech, wireless pin mic speech in different room condition, meeting room scenario, etc. On top of that the target problem can contain multi-domain, multi-speaker, etc. In this research, we particularly take interest in recognizing data recorded using mobile telephone channel over fourth generation (4G) cellular network as well as find interest in data recorded at university classroom. This research is based on the acoustic aspect of the domains.

This real-world telephone and classroom situation is simulated by re-recording a small amount of data in classroom by playing through loudspeaker and recording them using telephone device (calling and receiving) or low-quality wireless microphone. Previous research on supervised training of ASR indicates the requirement of large-scale transcribed data in target environment. However, it is costly to record and transcribe such amount of data for desired environment. Therefore, we adopt DNN-

based data augmentation method independent to ASR models. We also investigate the effectiveness of self-supervised-learning (SSL) based feature extraction with implicit end-to-end model to perform ASR task for small quantity classroom data.

Apart from the inherent challenge of domain adaptation difficulty, we face an interesting challenge in case of frame-by-frame processing of re-recorded data. Analysis shows temporal distortion in re-recorded speech data. We therefore proposed temporal alignment adjustment for each recording and filtering method for eliminating utterances with internal distortions.

1.3 Motivation and Contribution

There are previous researches on how to improve the acoustic model's performance for real data. According to them, one method to achieve improvement is to perform domain adaptation. To perform domain adaptation for real-world data, data augmentation is often used. Previous researches on this subject showed data augmentation by adding various recorded noises at desired SNR level or performing speed perturbation on clean data or using room impulse responses (RIR) to simulate the desired room acoustics. These augmented data contribute in producing improved performance of ASR systems for the target domain [5–9]. Also, there are DNN-based data augmentation methods which extract acoustic information to represent acoustic environments instead of creating simulated data [10].

The research interest in this dissertaion lies in performing research on automatic speech recognition (ASR) system of real-world scenario, such as, telephone speech or classroom lecture, where the speech is acquired by a low-quality wireless pin-mic. Both of the scenarios are considered particularly from the perspective of their acoustic aspect. With all the back ground study of improving speech recognition system, we indulge into a research where the data of target domain can be acquired in a cost effective fashion as well as can create data with similar acoustic attributes. Therefore, we take these motivations further and propose an automatic speech recognition system robust in real-world scenario, regardless of the architecture of

speech recognition model.

As stated before, there are numerous study about data augmentation or adaptation approaches which contribute to the improvement of the performance of ASR. They create a large amount of data using methods capable of adding RIR, noise, etc. to clean data to simulate target real environment speech. Sometimes DNN-based methods are applied, for example, variational autoencoder (VAE)-based data augmentation approach which creates source-to-target and target-to-source data to increase the amount of training data [11]. Also, there are researches on adapting Gaussian-based system for dysarthric speech, when collecting data for the target domain speakers is difficult [12]. However, there are a few researches that involves domain adaptation for target domain with neural network(NN)-based ASR. For NN-based approaches, most of the examples involve feature normalization, such as cepstral mean variance normalization (CMVN), applying vocal tract length normalization (VTLN) for speaker adaptation, or feature transforms such as feature space maximum likelihood linear regression (fMLLR), etc. [13,14]. There are research about increasing amount of data by applying data augmentation approaches that creates large amount of data of target domain [7, 9, 15, 16]. Apart from data augmentation, domain adaptation is also important for fine-tuning models for limited target domain data where the full model fine-tuning and partial fine-tuning are introduced [17–24].

In this research, a novel method of data augmentation is proposed that helps to imitate real environment by using a DNN-based feature transformation model for model-independent domain adaptation of DNN-HMM and end-to-end ASR models [25]. Similar approach of training DNN-based regression model has been adapted to perform denoising or dereverberation as pre-processing [26]. To the best of our knowledge, such small sized data have not been taken into consideration yet. Therefore, in this research, we train the regression model to learn on pairs of clean-rerecorded data of very small size (553 utterances). Experimental results show that it successfully transforms clean input data to real environment output data by regression-based mapping of feature catted feature transformation in this

dissertation. Since it learns to transform feature directly from real environment data, the computation cost is also minimal. The augmented data that prepared in this way are then used to perform fine-tuning of acoustic models. To acquire the best result, some of the original re-recorded speech are used along with the transformed features.

Moreover, re-recording of data is also an important part of this research. In case of supervised training of acoustic models, the data need to have proper segmentation and transcription. Recording in real environment is difficult. Even if such data are acquired, preparing segments and texts accordingly is more difficult. To overcome this difficulty, re-recording of speech from official data corpus with proper transcription is recommended. In this way, data of desired real environment, such as telephone channels, classroom data, etc. can be created with much less effort. However, recording for long hours in particular domains can cause distortion in resulting re-recorded speech or the recording quality may degrade due to human error. Therefore, it is recomended to acuquire re-recorded data as small as possible. Since there is evidence of feature transformation considering speaker related aspect using fMLLR with just two minutes of data per speaker according to the research cited in [14], re-recording small amount of data in the target acoustic domain should be enough with our proposed feature transformation approach as well. It encouraged the development of the solution by playing monologue speech through telephone and recording it at the callers end. Another variation is classroom data. Monologue speeches are also re-recorded in similar fashion with necessary adjustments for classroom environment with a normal low-quality wireless pin mic. The dataset is used in the experiments with the proposed models. The experiments prove that using small amount of re-recorded data with proposed augmentation is effective for domain adaptation for fully supervised ASR models and using a small amount of re-recorded data performs the best for self-supervised model-based ASR model.

The contributions of the research are as follows:

- A domain adaptation approach independent to ASR models by preparing speech data which contain target domain characteristics is proposed. Experiments on

state-of-the-art ASR models are performed and proved the effectiveness of the proposed method on them.

- Limited duration of re-recorded speech is used for acoustic domain adaptation.
- To prepare speech data with target domain characteristics in low cost, we adopt following approaches:
  1. Use an already existing corpus
  2. Re-recording the corpus data by playing them in real environment for only a limited short period of time
  3. Performing post processing on them to adapt them for ASR model training
- It is proved that by involving a simple regression model for feature transformation, it is possible to obtain data with target domain characteristics with only small amount of real-data (duration of less than 1 h) to improve the performance of ASR to a great extent.
- Comparative study of two end-to-end ASR models are performed to investigate the effectiveness of domain adaptation with limited data in case of data with variability in recording conditions.
  - Self-supervised learning model-based audio encoder is pretrained using large amount of data of different language, speaker and acoustic domain. Such pre-training method is effective in encoding acoustic information for fine-tuning ASR model.

1.4 Outline of the Dissertation

In Chapter 1, the background, challenges, motivations and contributions of this research are presented. In Chapter 2, the overview of speech recognition system is described in details including fundamental and modern approaches such as supervised and self-supervised learning-based training of speech recognition models. In Chapter

3, the proposed speech recognition system for real-environment fine-tuned with DNN-based data augmentation is presented. This chapter also describes acquisition and proposed pre-processing of re-recorded data in desired target environment leading to improvement of speech recognition performance for the target domain. In Chapter 4, self-supervised learning-based approach of domain adaptation is proposed for limited target domain data with low audibility, which gives better performance than previous research. In Chapter 5, the conclusion of the research performed in this dissertation and the future directions for where there are rooms for improvements are stated.

CHAPTER 2

OVERVIEW OF AUTOMATIC SPEECH RECOGNITION SYSTEM

This chapter provides an overview of the automatic speech recognition (ASR) technology. First, it describes the general theory of an ASR system and the fundamental technologies in Section 2.1, and the basics principle of deep learning models is described in Section 2.2. As an application of DNN architecture, the automatic speech recognition models using DNN architectures are introduced in Section 2.3. A modern approach of audio encoding, called self-supervised learning-based model is discussed in Section 2.4.

2.1 Fundamentals of ASR System

An ASR system in a broad sense is a technology that recognizes the content of speech spoken by a human. Some recognition systems integrate other personal vocal information, such as prosody for emotion or dialect recognition along with acoustic and pitch information. In this research, we focus on the specific aspect of speech recognition, where it recognizes acoustic and linguistic information of speech segments, also known as utterances and transcribes it to text.

The traditional way of speech recognition is to apply statistical fuctions as shown in the following equation.

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \log P(\boldsymbol{y}|\boldsymbol{x}) \tag{2.1}$$

$\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$ is the acoustic feature vector sequence given to the system as an input and $\boldsymbol{y} = (y_1, y_2, \ldots, y_T)$ is $\boldsymbol{x}$'s corresponding label vector, which can be words, syllables, phonemes, etc. When the probability $P(\boldsymbol{y}|\boldsymbol{x})$ is given into the equation, the maximum likelihood $\hat{\boldsymbol{y}}$ is achieved. This $P(\boldsymbol{y}|\boldsymbol{x})$ can be expressed according to Bayes' theorem as the following equation.

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y})}{P(\boldsymbol{x})} \tag{2.2}$$

Eq. (2.2) is substituted into Eq. (2.1). As the denominator in Eq. (2.2) has nothing to maximize with respect to $\boldsymbol{y}$, Eq. (2.1) can be rewritten as

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} P(\boldsymbol{x}|\boldsymbol{y})P(\boldsymbol{y}) \tag{2.3}$$

An additional term is further added to the above-mentioned equation by taking its algorithm to consider scores according to the length of a label.

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \{\log P(\boldsymbol{x}|\boldsymbol{y}) + \alpha P(\boldsymbol{y}) + \beta|\boldsymbol{y}|\} \tag{2.4}$$

$P(\boldsymbol{x}|\boldsymbol{y})$ is the probability of acoustic feature vector sequence $\boldsymbol{x}$ getting observed when a certain label vector $\boldsymbol{y}$ is given and is referred as *acoustic model*. $P(\boldsymbol{y})$ is a prior distribution of the label vector $\boldsymbol{y}$ and is referred as *language model*. α is a parameter that adjusts the range the difference between the acoustic model score and language model score and is referred as *language weight*. β is a parameter that adjusts the frequency of omission or insertion by the recognition system by adding the score according to the word length and is referred as *insertion penalty*. Speech recognition can be referred as a problem which searches for $\boldsymbol{y}$ that maximizes the product of the acoustic model and the language model probabilities. This search is called *decoding*. The flow of a traditional speech recognition is shown in Figure 2.1.

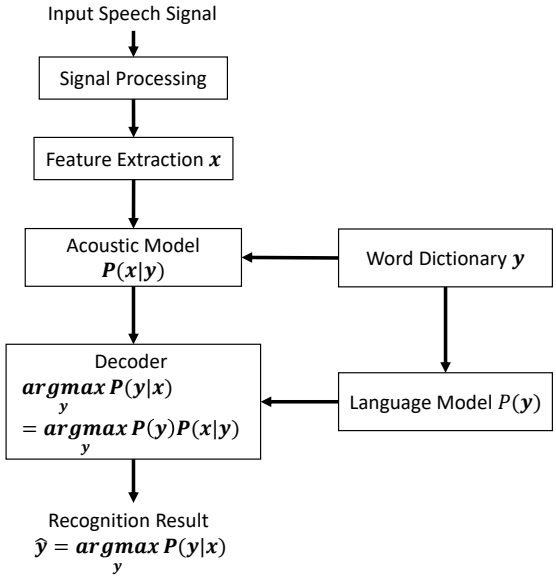


Figure 2.1: Block diagram of speech recognition system

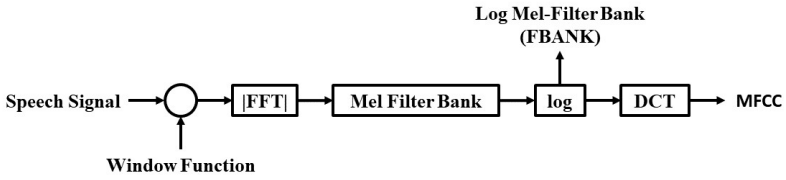


Figure 2.2: Flow of feature extraction

2.1.1 Extraction of Acoustic Feature

Feature extraction section expresses features that are useful for recognition as a vector series. This section removes information that is too small to be relevant for phonemes or word sequences to be recognized. The basic extraction method that is often used for speech recognition is Mel-Frequency Cepstrum Coefficient (MFCC). In this research, we use MFCC as parameters of speech features along with Log Mel-Filter Bank (FBANK), which is obtained in the middle stage of MFCC calculation. The flow of MFCC feature extraction is shown in Figure 2.2.

The speech signal, which is an analog signal, is sampled according to the sampling theorem and passed through an A/D converter to become a digital signal. In this research, sampling is performed at a sampling frequency of 16kHz and quantization at a quantization bit rate of 16 bits. Based on the assumption that speech signal

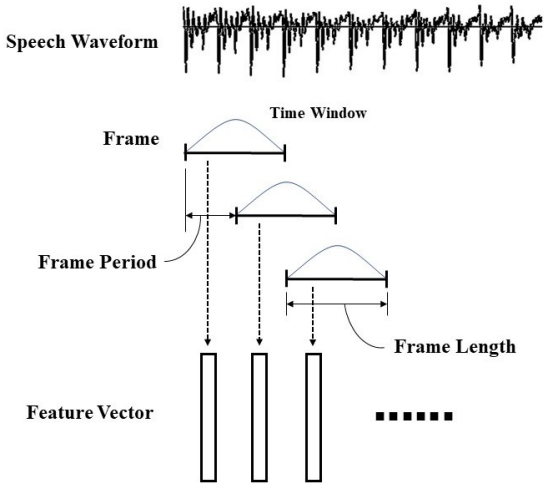


Figure 2.3: Feature extraction from speech waveform

does not change dramatically in a short period of time, we apply a window function with a period of about 20–30ms to cut out a section from the signal. The window function is used to prevent sudden changes at the start and end points of the cutout space. In this research, the following Hamming window is used.

$$\omega(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right), 0 \leq n \leq N-1 \tag{2.5}$$

The cutout section mentioned above is called a frame. Its length is called a window length or frame length and the period of moving this section is called a frame period or frame shift (10 ms) as shown in Figure 2.3.

In order to compensate for the loss of information caused by multiplying by the window function, the frame period is generally set half the frame length and the analysis section is overlapped. The Fast Fourier Transform (FFT) is then performed on the speech waveform of N points cut out by the window function. The obtained N amplitude spectrum are subjected to filter bank analysis using L number of band-pass filters (triangular windows) equally spaced along the Mel frequency axis as shown in Figure 2.4. That is, the power of the frequency axis of the frequency band signal corresponding to the width of the window is obtained by the weighted sum of the amplitude spectrum $|S(k)|$ of the single spectral channels.

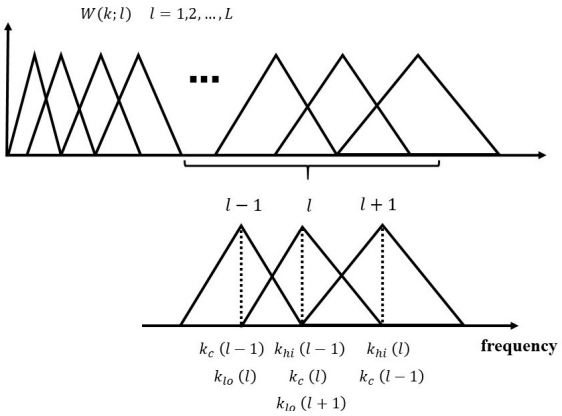


Figure 2.4: Filter used for analysis

$$m(l) = \sum_{k=lo}^{hi} W(k;l)|S(k)| \quad (l = 1, \ldots, L) \tag{2.6}$$

$$W(k;L) = \begin{cases} \frac{k-k_{lo}(l)}{k_c(l)-k_{lo}(l)} & k_{lo}(l) \leq k \leq k_c(l) \\ \frac{k_{hi}(l)-k}{k_{hi}(l)-k_c(l)} & k_c(l) \leq k \leq k_{hi}(l) \end{cases} \tag{2.7}$$

In the above equation, $k_{lo}(l)$, $k_c(l)$ and $k_{hi}(l)$ indicate the lower, center and upper limit spectral channel numbers of the l-th filter respectively. The relationship between these filters are shown in the following equation.

$$k_c(l) = k_{hi}(l-1) = k_{lo}(l+1) \tag{2.8}$$

In addition, $k_c(l)$ is also evenly spaced on the Mel frequency axis. Mel frequency is a human measurement of intervals towards the pitch of a certain sound. It is calculated by the following equation.

$$Mel(f) = 2569\log_{10}\left(1 + \frac{f}{700}\right) \tag{2.9}$$

Finally, by taking logarithm of power in L bands obtained by the filter bank analysis and by discrete cosine transforming the F-bank features of the L-dimensional vector, the MFCC features are obtained. In this research, we mostly use 43 dimension of speech features consisting of 40 dimension of F-bank features and 3 dimension of

pitch feature. Pitch is a feature that is incorporated with the fundamental frequency f_0 of any sound, especially for human voice. In this case, the pitch is extracted by considering fundamental frequency guided by Normalized Cross Correlation Function (NCCF) within a shifting window of time frame [36]. The use of pitch information is effective for this research since it helps us achieving speaker information when the amount of data is very small, thus allowing us to exploit it for our benefit.

2.1.2 Cepstral Mean Variance Normalization

In order to reduce the effect of differences in transmission during speech input, it is effective to normalize the speech features. A common method to do this is Cepstral Mean Normalization [37]. The normalization is done by subtracting the long-term mean of cepstrum from the value of cepstrum of each frame, assuming the following two points.

1. The effect caused by the noise characteristics is stationary.
2. The steady transfer characteristics can be approximated by averaging the cepstral coefficients of the audio signal over a relatively long period of time.

Assuming that the noise characteristics $T(e^{j\omega})$ are convoluted in the speech, the logarithmic amplitude spectrum of the speech to be analyzed is as follows.

$$\log|S(e^{j\omega})| = \log|G(e^{j\omega})| + \log|H(e^{j\omega})| + \log|T(e^{j\omega})| \tag{2.10}$$

where $G(e^{j\omega})$ and $H(e^{j\omega})$ each represent the spectrum of the audio signal, and the transmission characteristics of the tone filter. Considering that the sound source component is removed from the cepstral coefficient series, the Fourier transform of the cepstrum series can be expressed as the following equation.

$$F[C(n)] = \log|H(e^{j\omega}; n)| + \log|T(e^{j\omega})| \tag{2.11}$$

where $F[\cdot]$ represents the Fourier transform and n the frame number. If the noise characteristics are frame independent and constant, the Fourier transform when

averaging the cepstral coefficients in frames is shown below.

$$F\left[\frac{1}{N}\sum_{n=1}^{N}c(n)\right] = \frac{1}{N}\sum_{n=1}^{N}\log|H(e^{j\omega};n)| + \log|T(e^{j\omega})| \tag{2.12}$$

Thus, the cepstral coefficients, with the mean removed, are expressed by the following equation.

$$F\left[c(n) - \frac{1}{N}\sum_{n=1}^{N}c(n)\right] = \log|H(e^{j\omega};n)| - \frac{1}{N}\log|H(e^{j\omega};n)| \tag{2.13}$$

The equation above shows features where the component $T(e^{j\omega})$, which corresponds to stationary noise and other characteristics, has been removed. In the process of learning and recognizing a speech recognition system, CMN is expected to behave environmentally robust when the time averages of the logarithmic spectrum of speech can be regarded as equal. Furthermore, the method of normalizing not only the mean, but also the variance is called Cepstral Mean and Variance Normalization (CMVN). In this study, CMVN is applied to the features for speech recognition experiments.

2.1.3 Hidden Markov Model for Acoustic Modeling

This section describes acoustic modeling. Acoustic Model defines the probability that the acoustic feature $\boldsymbol{x}$ outputs when the label sequence $\boldsymbol{y}$ is given. It is shown as the probability $P(\boldsymbol{x}|\boldsymbol{y})$ in Eq. (2.4). In speech recognition, the Hidden Markov Model (HMM) is commonly used to model time-series patterns with temporal stretching. HMMs are defined as nondeterministic finite state automaton in the sense that the state transitions are not uniquely determined by observable output symbols. In the acoustic model used in speech recognition, the symbol sequence, which is the sequence of the input speech, is treated as a generative model of speech that is created while transitioning between states. Generally, continuous density HMM, which models the HMM with the feature vector sequence after feature extraction is used. The continuous density HMM for speech recognition consists of a set of transitional states,

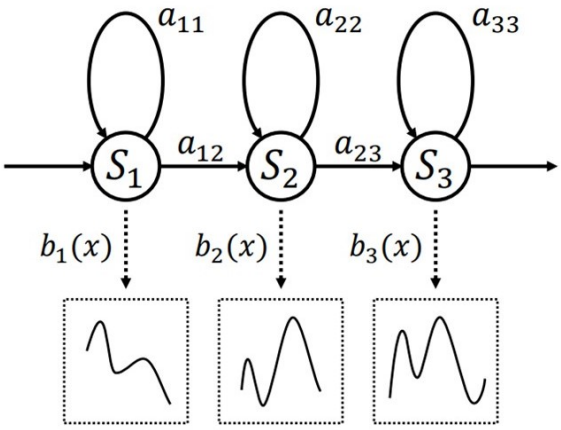


Figure 2.5: Example of HMM acoustic model

a transitional probability between states and an output probability distribution (output probability density function) at the time of state transition. The shape of HMM is generally left-to-right as shown in Figure 2.5 which is also used for this research.

In Figure 2.5, S_i represents the i-th state, a_{ij} represents the transition probability from state S_i to the next stage S_j, and $b_j(\boldsymbol{x})$ represents the output probability density of outputting the feature vector $\boldsymbol{x}$ in state S_j at the time of transition from state S_i to the next state S_j. Here, the sum of the transition probability to self and the transition probability to the next state for the transition probability a_{ij} is 1. In HMM, the likelihood of the observed value sequence (feature vector sequence) $\boldsymbol{X} = \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T$ for this model is expressed by the following equation.

$$P(\boldsymbol{X}|\lambda) = \sum_{S_1,S_2,\ldots,S_T} \prod_{t=1}^{T} a_{ij} b_j(\boldsymbol{x}_t) \tag{2.14}$$

where λ denotes the set of HMM parameters $\{a_{ij}\}_{i,j=1}^{N}$ and $\{b_i(x)\}_{i,j=1}^{N}$, and $\{S_1, S_2, \ldots, S_T\}$ denotes a state sequence. In general, a multidimensional normal distribution (Gaussian distribution) is often used as the output of probability density function. The multidimensional Gaussian distribution can be expressed by using K-order feature

vector $\boldsymbol{x}$, mean vector $\boldsymbol{\mu}$ and covariance matrix $\sum$ as the following equation.

$$N(\boldsymbol{x}|\boldsymbol{\mu}, \sum) = \frac{1}{\sqrt{2\pi|\sum|^{-1}}} exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^{T}\sum^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}_{i}\right)\right) \tag{2.15}$$

When the distribution of the output probability is too complex to be represented by a single multidimensional Gaussian distribution, it can be better represented by a weighted sum of M multidimensional normal distributions, called as the Gaussian Mixture Model (GMM). The output probability density that generates the feature vector $\boldsymbol{x}_t$ observed at a certain point in time t can be calculated using the GMM. The power probability density to generate the feature vector $\boldsymbol{x}_t$ observed at time t using the GMM is calculated as the following equation.

$$b(\boldsymbol{x}_t) = \sum_{m=1}^{M} \lambda_m N(\boldsymbol{x}_t|\boldsymbol{\mu}_m, \sum_m) \tag{2.16}$$

where λ_m is the weight of the m-th multidimensional Gaussian distribution and $\sum_{m=1}^{M} \lambda_m = 1$.

In recent years, DNN-HMMs, which use Deep Neural Networks (DNNs) for output probability calculation, have replaced GMMs. Neural Networks and DNN-HMM acoustic models are discussed later in Section 2.2.

2.1.4 Word N-gram for Language Modeling

Language Model is defined as $P(\boldsymbol{y})$ in Eq. (2.4), which refers to the prior distribution of the label sequence $\boldsymbol{y}$. In speech recognition, free contextual methods and word N-gram are used for language modeling. Word N-gram language models are commonly used in the case of large-vocabulary speech recognition system. It focuses only on the local ordering of words and predict the next word in probabilistic way when $N-1$ words are known. The N-gram models are called unigram, bigram and trigram respectively when $N = 1, 2, 3$. In this research, trigram ($N = 3$) is used as the language model in case of DNN-HMM hybrid speech recognition system.

2.1.5 Decoder

Decoder is a module which finds the recognition system's optimal word sequence by combining scores from acoustic and language models. Its goal is to find the $\hat{\boldsymbol{y}}$ that maximizes Eq. (2.4). In the case of having N words, the number of possible word sequences for a vocabulary size $|V|$ is very large $|V|^N$. Large-Vocabulary Continuous Speech Recognition (LVCSR) Systems use a variety of algorithms to reduce the computational complexity and improve the efficiency of the search.

In this research, the triphone-unit GMM-HMM learning method is used to obtain the HMM state alignment required for DNN-HMM training. During the training of the DNN-HMMs, feature transformation methods called fMLLR (feature space Maximum Likelihood Linear Regression) adaptation [27], SAT (Speaker Adaptive Training) [28], LDA (Linear Discriminant Analysis) features [29] and MLLT (Maximum Likelihood Linear Transformation) [30] are used to improve speech recognition accuracy with GMM-HMM acoustic model.

2.2 Basic Principle of Deep Learning Model

In the recent years, Neural Network-based methods have shown their effectiveness in various fields, such as speech recognition and image recognition. In this section, we describe the basic principles of Neural Networks and the representative methods applicable for speech recognition. It is often modeled after the functionalism of brains. Information processing in the brain of an organism is carried out by a network of neurons that are connected to each other. Each neuron receives electric signals from the neurons it is connected to, and if the sum of the signals reaches a certain value, the neuron itself emits an electric signal. An Artificial Neural Network (ANN) is an engineering model of the working principle of a neuron as a node. It is represented as the following equation.

$$y = u\left[\sum_i w_i x_i - \theta\right] \tag{2.17}$$

where w_i is the synaptic coupling weight for input x_i, θ is the threshold and $u[\cdot]$ is the unit step function. In Eq. (2.17), the transmitted signals x_i are weighted and added together, and when they exceed a certain threshold, the nueron fires.

A typical ANN model uses the sigmoid function shown in Eq. (2.18) instead of unit step function in Eq. (2.17). The sigmoid function is differential but $[u]$ is not.

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{2.18}$$

A simple ANN is one that combines many of these units. The model of an ANN is constructed by combining a number of artificial neurons. A neural network for pattern recognition is generally constructed by combining several artificial neurons. As shown in Figure 2.6, a neural network applied to pattern recognition is generally connected only in the forward direction from the input unit to the output unit, and a simple hierarchical structure is created by stacking these layers. This structure is called a Feed-Forward Neural Network (FFNN) [31]. An FFNN generally consists of three layers: an input layer, a hidden layer and an output layer. The input layer

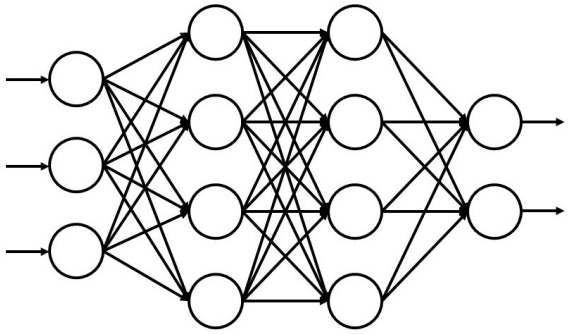


Figure 2.6: Layer Structure of a multi layer perceptron neural network model

corresponds to cells that receive information from the lower area (e.g., optic nerve cells) while the middle layer corresponds to cells that transmit the signals to the brain. The output layer corresponds to brain cells which identify classes.

A Deep Neural Network (DNN) is a neural network with a deep structure consisting of multiple layers of intermediate layers. A neural network can only extract simple features in the layers closest to the input, but its expressive capability can be improved by taking the weighted sum of these layers. For that reason, it was thought that increasing the number of layers to be trained and creating a multi-layered structure would improve the representational capability of the model. However, simply increasing the number of layers and training using Back Propagation (BP) method [32] does not converge the network since the error signal is less likely to propagate away from the output layer. In addition, since a random number is used as the initial value of the coupling strength, it is easy to fall into a local minimum solution, and Multi Layer Perceptron (MLP) wuth three or so shallow layers has been commonly used. To address these problems, Hinton et al. proposed a method for constructing DNNs using a Restricted Boltzmann Machine (RBM) [33].

The deep learning-based approaches of speech recognition adopted in this research largely fall into two of the most popular categories. One is supervised learning-based approaches described in Section 2.3 including one of the traditional DNN-HMM hybrid model and rather modern encoder-decoder based end-to-end speech recognition model. Another approach is called self-supervised learning-based speech recognition, described in Section 2.4.

2.3 DNN-based ASR

Supervised learning-based approach of speech recognition is the most traditional approach of training a recognition system with training representatives and their corresponding labels.

2.3.1 DNN-HMM Hybrid ASR Model

In this section, we describe DNN-HMM, an acoustic model using DNN, as a method of applying DNN to speech recognition. As described in Section 2.1.3, acoustic models have traditionally been modeled by HMMs, and the output density function $b_i(\boldsymbol{x}_t)$ which generates the feature vector $\boldsymbol{x}_t$ observed at a particular time t and is represented by a GMM as shown in Eq. (2.16). DNN-HMM, which replaces the calculation of the output probability by GMM with a representation by DNN, has a higher output probability than the conventional GMM-HMM [34].

To train the DNNs, the state labels of the HMM corresponding to each time of the input signal are used as the teacher signal. This allows us to learn a nonlinear function that calculates the probability value $P(S_i|\boldsymbol{x}_t)$ of the state class S for each phoneme with respect to the input features. The probability value of the phoneme class obtained from the output of the DNN can then be transformed into the probability value of the phoneme class by Bayes' theorem.

$$P(\boldsymbol{x}_t|S_i) = \frac{P(S_i|\boldsymbol{x}_t)P(\boldsymbol{x}_t)}{P(S_i)} \tag{2.19}$$

$$\propto \frac{P(S_i|\boldsymbol{x}_t)}{P(S_i)} \tag{2.20}$$

Here, $P(\boldsymbol{x}_t)$ is omitted since it does not affect the optimization of Eq. (2.4). $P(S_i)$ is obtained from the frequency of occurence of the correct answer label in the training data. Speech recognition using DNN-HMM is performed by replacing the calculation of the output distribution of HMM by transformation in the form of $P(\boldsymbol{x}_t|S_i)$. Figure 2.7 depics a scematic diagram of DNN-HMM hybrid ASR model.

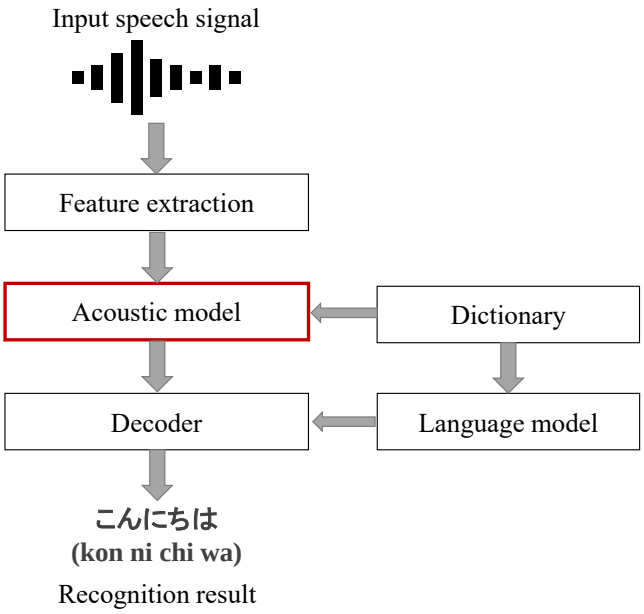


Figure 2.7: Scematic diagram of DNN-HMM hybrid ASR model.

The input to a GMM-HMM acoustic model is generally a feature set consisting of MFCCs and their dynamic features. For the DNN-HMM acoustic model, it is known that the recognition accuracy is higher when the input is a log-Mel Filter Bank. In addition, it is common to concatenate the central frame with several frames before and after it and feed them to the input simultaneously, while the output is only the central frame.

Time Delay Neural Network

A time delay neural network is a feed-forward neural network that effectively models long range temporal dependencies [38]. It exploits a modular and incremental design to create larger networks from sub-components [39]. Traditional architectures compute the hidden activations at all time steps which makes it computationally expensive. The architecture used in this research uses a sub-sampling technique which allows the hidden activations to be computed at only a few time steps at each level [40] making it computation friendly. In Figure 2.8 depicts the network architecture of a TDNN where deeper the layer is, the larger context it covers, thus making is robust to temporal deviation, yet computation friendly since it does not consider in between feature instances.

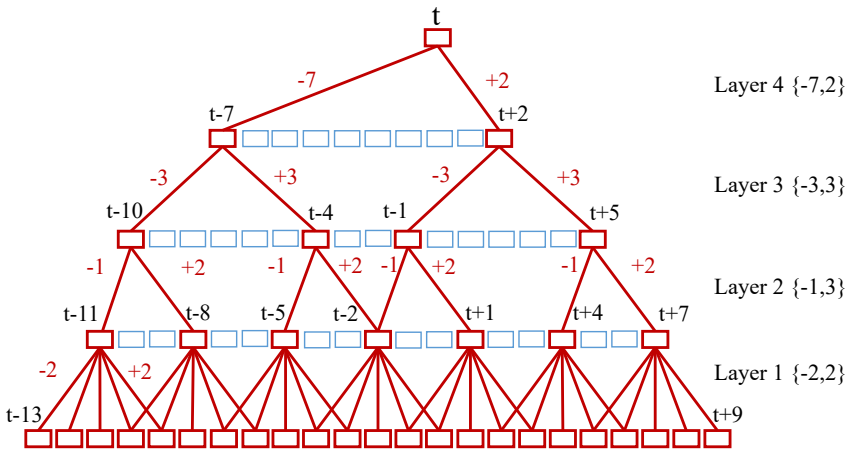


Figure 2.8: Layer Structure of a Time Delay Neural Network (TDNN) Model

2.3.2 Encoder-decoder based End-to-end Speech Recognition Model

DNN-based HMM acoustic models are powerful methods for speech recognition tasks, but they have some problems. Models using conventional neural networks are often built on the assumption that the correspondence between inputs and outputs is known. On the other hand, in speech recognition tasks, the frame length of the input speech feature sequence and the label length of the output label sequence are often different, and the correspondence between input and output is often unknown. Therefore, when using a DNN-based HMM acoustic model, we first assume the hidden state of the HMM corresponding to each speech frame using the GMM in order to train the DNN. The framework is often adopted. However, in this framework, the performance of the speech recognition system using DNN is affected by the performance of the GMM-HMM.

Therefore, in recent years, the end-to-end (E2E) Neural Network method has been attracting attention. Encoder-Decoder, which is an end-to-end model based on neural network, can learn the correspondence between input and output without explicit correspondence. However, encoder-decoder has the problem that it is difficult to learn when the correspondence between input and output is complicated. Figure 2.9 shows a simple block diagram of encoder-decoder end-to-end ASR model. Encoder-Decoder

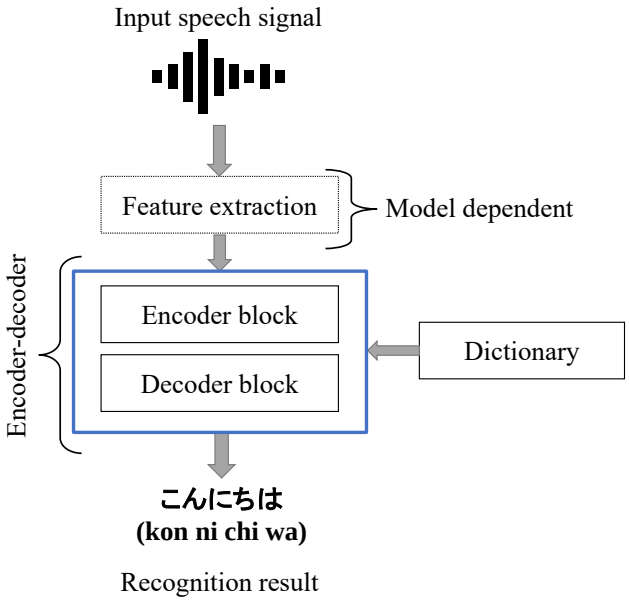


Figure 2.9: Scematic diagram of end-to-end ASR model.

with an attention mechanism was first shown to outperform the conventional method in the machine translation task [41], and then showed results comparable to the conventional method in the speech recognition task [42–44].

Conventionally, long short-term memory (LSTM) and bidirectional LSTM (BLSTM), which can capture time dependency information more efficiently, have been used as the neural network used for the encoder and decoder in the encoder-decoder, but recently a model structure called a transformer has been proposed. This method [45], which was proposed for the machine translation task, achieved both reduction of learning time and performance by using an approach called self-attention. This method was also used in speech recognition tasks and was shown to perform on par with the Encoder-Decoder using a bidirectional LSTM [53].

However, since the attention mechanism weights the entire input, there is a problem that weighting is difficult when there are repetitions of the same phoneme string at distant positions. To solve this problem, a method of adding penalties to distant positions during weight calculation [42] and a method of limiting the weighting range rather than the entire input have been proposed [43,44]. On the other hand, a method called Connectionist Temporal Classification (CTC) [47–49] maps

inputs and outputs in a different way from Encoder-Decoder. In CTC, by using a special symbol called a blank character, each label in the output corresponds to each frame in the input on a one-to-one basis. Therefore, a hybrid approach was proposed to solve the weighting problem by incorporating the temporal constraints of CTC into the attention mechanism, and both Encoder-Decoder [50–52] and Transformer [53] with bidirectional LSTMs were proposed. It was shown to improve performance.

Encoder-Decoder

Encoder-Decoder is not a simple one-layer Neural Network like autoencoder (AE) [54], but a Neural Network that can consider time-series information such as RNN and LSTM. This is a model that compresses to a long vector and restores it through a decoder. Figure 2.10 shows the structure of Encoder-Decoder. Encoder-Decoder is expressed mathematically as follows:

$$H = Encoder(X) \tag{2.21}$$

$$Y = Decoder(h_T) \tag{2.22}$$

where $X = x_1, x2, ..., x_T$ is input of length T frames, $H = h_1, h_2, ..., h_T$ is the feature vector and $Y = y_1, y_2, ..., y_N$ is the output label sequence of length N. Encoder is generally a neural network using bidirectional LSTM, and decoder is a neural network using LSTM. The Encoder compresses the input audio feature sequence X into a smaller feature representation h_T, and the Decoder restores the compressed feature representation h_T. At this time, when the Decoder estimates a certain output label y_n, not only the state of the hidden layer of the Decoder but also the previous output label information y_{n-1} of the Decoder itself. This makes it possible to extract and restore the required amount for each output from the feature representation h_T compressed by the Encoder.

The input layer Encoder and the output layer Decoder are separate RNNs, so the

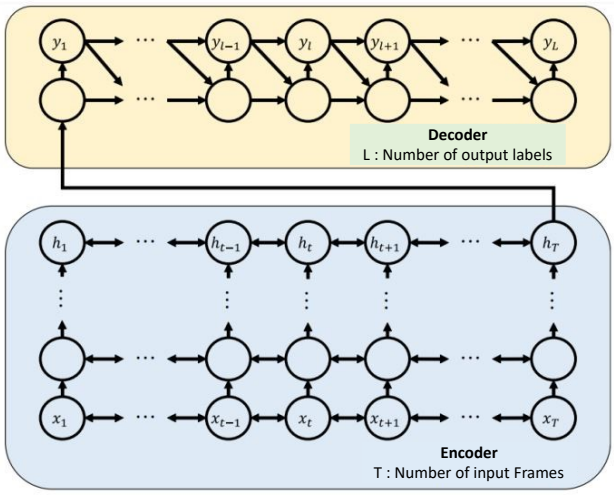


Figure 2.10: Encoder-decoder neural network model

input data and output data need not have the same length. In order to prevent the output from continuing infinitely, it is trained to output a special symbol (End of Sequence: EOS) that indicates the end during training. In speech recognition tasks, the length of the input acoustic feature sequence and the length of the correct label (phoneme, character, etc.) sequence often differ. The advantage is that it can be done. However, Encoder-Decoder can only hold the compressed feature representation in one fixed-length vector h_T. Therefore, when the input data is longer than the number of units in the hidden layer and the correspondence is complicated, all the information cannot be compressed into one fixed-length vector, and some information is lost. On the other hand, if the input data is too short for the number of units in the hidden layer, overfitting will occur. Therefore, a method of incorporating a structure called attention mechanism into the Encoder-Decoder was proposed. The attention mechanism is described in the next section.

Attention Mechanism

The attention mechanism is a method that converts the input into a compressed representation by calculating the weight for each frame of the input and taking the weighted sum [44]. There are two methods of this to compute the weights of attention mechanisms: inner product attention calculated by inner product and

additive attention calculated by a multilayer perceptron (MLP) of 1 hidden layer. In general, the inner product attention has fewer parameters and can shorten the computation time. In this study, we use inner product attention. An attention mechanism using inner product attention is expressed by the following formula:

$$Q = W_q Y + b_q \tag{2.23}$$

$$K = W_k X + b_k \tag{2.24}$$

$$A = softmax(\frac{QK_T}{d}) \tag{2.25}$$

$$V = W_v X + b_v \tag{2.26}$$

$$H = AV \tag{2.27}$$

Here, K is Key, V is Value, and each is a feature sequence obtained by transforming the input. In other words, it generates a pair of Key and corresponding Value for each frame like a dictionary object. Q is Query, and the weight is calculated from the relationship between Query and Key. d is a scaling constant, and if the value of the inner product QK_T becomes too large, the gradient of softmax of backpropagation becomes extremely small and learning fails. It is used for A is a value called attention weight, and is calculated so that the sum is 1 for each frame by using the softmax function. H is the output of the layer, which is a compressed representation of the input represented by the weighted sum from the relationship between X and Y. W_q, W_k, W_v are learned weight parameters, b_q, b_k, b_v are learned biases is a parameter. From the above, the attention mechanism can be thought of as an operation that, when a Query is given, searches for Keys that match it, and extracts the corresponding Values. The input audio feature sequence is used for X, but depending on the feature sequence used for Y, it is divided into two types: source-target attention and self-attention. Source-target attention is an attention mechanism that uses Y as a feature that converts the output label sequence into an embedded representation. At this time, the attention weight A

is calculated using the Query converted from the audio feature sequence X and the Key converted from the label sequence Y. The attention mechanism used in Encoder-Decoder is source-target attention. It is used before the Decoder layer so as to extract the frame related to each label for the intermediate feature H, which is the output of the Encoder. Self-attention is an attention mechanism that uses $Y = X$, that is, the input speech feature sequence. At this time, the attention weight A is calculated using Query and Key, which are obtained by transforming the speech feature sequence X, respectively. Thus, self-attention has the role of transforming the input into higher-order features. Since self-attention uses only one feature sequence as input, self-attention can be considered as a convolutional layer that convolves the entire input. In both attention mechanisms, the performance is improved by using a multi-head attention mechanism that divides the feature sequence into several heads in the dimension direction instead of using the feature sequence as it is, so that different heads process different subspaces. It has been shown experimentally [45]. Therefore, in this study, we also use a multi-head attention mechanism.

Figure 2.11 shows the model structure of Encoder-Decoder with attention mechanism (source-target attention). It is the same as when the attention mechanism is not used, up to the conversion of the input audio feature sequence X to the intermediate feature sequence H, which is converted to a higher-order feature using the Encoder. However, the Encoder-Decoder without the attention mechanism uses only h_T as the input of the Decoder, but the attention mechanism uses H. It generates a higher-order feature representation c_l by taking the weighted sum of all y_n and use it. As a result, appropriate information can be generated regardless of the input length, and it is possible to avoid missing information and over-learning.

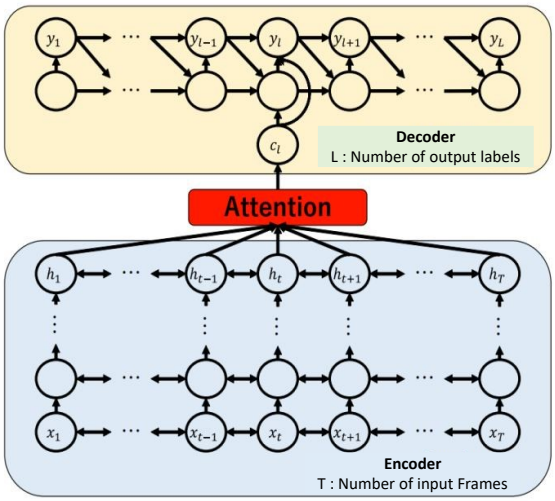


Figure 2.11: Encoder-decoder neural network model with attention mechanism

Transformer

Transformer [45] is a model structure that combines self-attention in the encoder layer and self-attention and source-target attention in the decoder layer without using RNNs such as LSTM and bidirectional LSTM as an encoder-decoder framework. In general, RNN computation uses the results of previous frames to compute a certain frame, so parallel computation is difficult. On the other hand, the self-attention used in Transformer can be calculated for all frames at once like CNN, so it is easy to perform parallel calculation and the performance of GPU can be used to the maximum. Therefore, the computation time can be shortened compared to using LSTM. However, unlike LSTM, Transformer does not have a network structure that expresses temporal relationships, so time series information must be given explicitly. Therefore, a method called positional encoding [45] is used. A position encoding is inserted before the Encoder layer:

$$\hat{X'} = X + PE \tag{2.28}$$

$$\boldsymbol{PE}[t,d] = \begin{cases} \sin \frac{t}{10000^{\frac{d}{D}}} & (d = \text{even number}) \\ \cos \frac{t}{10000^{\frac{d}{D}}} & (d = \text{odd number}) \end{cases} \tag{2.29}$$

where $t = 1, 2, ..., T$ represents frames and $d = 0, 1, ..., D-1$ By position encoding, temporal relations are embedded in the feature sequence, and Transformer can also consider temporal relations. In Transformer, a position-by-position feedforward network [45] is used to transform the feature sequence that takes the weighted sum after self-attention in the Encoder layer and after source-target attention in the Decoder layer. The position-wise feedforward network consists of MLP with ReLU as the activation function in the hidden layer 1 layer. Since Transformer converts to higher-order feature representations using a multi-layered network, the problem of vanishing gradients must be considered as learning progresses. Therefore, we deal with the gradient vanishing problem by using the residual connection [55] for each attention mechanism and the feedforward network for each position. In addition, dropout [56] is also used to avoid overfitting.

Connectionist Temporal Classification

Connectionist Temporal Classification (CTC) is a learning method that associates the input and output lengths with the same length by inserting a special symbol called a blank symbol into the output label string. CTC is expressed by the following formula:

$$\boldsymbol{C} = \text{softmax}(MLP(\boldsymbol{X})) \tag{2.30}$$

$$\boldsymbol{Y} = B(\boldsymbol{\pi}) \tag{2.31}$$

$$p(\pi \mid \boldsymbol{X}) = \prod_{t=1}^{T} C_{\pi_t}^{t} \tag{2.32}$$

$$p(\boldsymbol{Y} \mid \boldsymbol{X}) = \sum_{\pi \in B^{-1}(\boldsymbol{Y})} p(\boldsymbol{\pi} \mid \boldsymbol{X}) \tag{2.33}$$

where $X = x_1, x_2, ..., x_T$ are input speech feature sequences, $Y = y_1, y_2, ..., y_N$ is the output label sequence, and $T > N$. C is the output of Neural Network, which is converted to the probability of each label for each frame by the softmax function. π is a redundant label sequence of length T that includes blank symbols in the output

label sequence Y, and $B(\pi)$ is a function that converts the redundant label sequence to the original label sequence. $p(\pi|X)$ is the probability of outputting redundant label sequence π when input X is given. $p(Y|X)$ is the probability of outputting the label string Y when the input X is given, and the redundant label string converted to Y using the function B in the set $\pi \in B^{-1}(Y)$, it is obtained by summing the probabilities of each π. In CTC, each input frame corresponds to each output label on a one-to-one basis. Therefore, each frame has a constraint that it cannot be associated with labels that are far apart in time. The hybrid approach is an approach that utilizes this constraint and combines it with the attention mechanism. We describe the hybrid approach in the next section.

CTC/Attention Hybrid Approach

The Encoder-Decoder combined with the attention mechanism can now use the entire input information without excess or deficiency. However, since the attention mechanism weights the entire input, there is a problem that weighting is difficult when the same phoneme sequence is repeated at distant positions. Therefore, a method [50–53] was proposed to solve the weighting problem by incorporating the time constraint of CTC into the attention mechanism. The method of using this Encoder-Decoder and his CTC together is called a hybrid approach. A hybrid approach is expressed in the following formula:

$$H = Encoder(X) \tag{2.34}$$

$$L_{ctc} = CTC(H) \tag{2.35}$$

$$L_{dec} = Decoder(H) \tag{2.36}$$

$$L = \lambda L_{dec} + (1 - \lambda) L_{ctc} \tag{2.37}$$

where X is the input acoustic feature sequence, and H is the intermediate feature sequence obtained by transforming the acoustic feature into a higher-order feature representation. L_{ctc} is the loss of CTC, L_{dec} is the loss of the decoder using source-

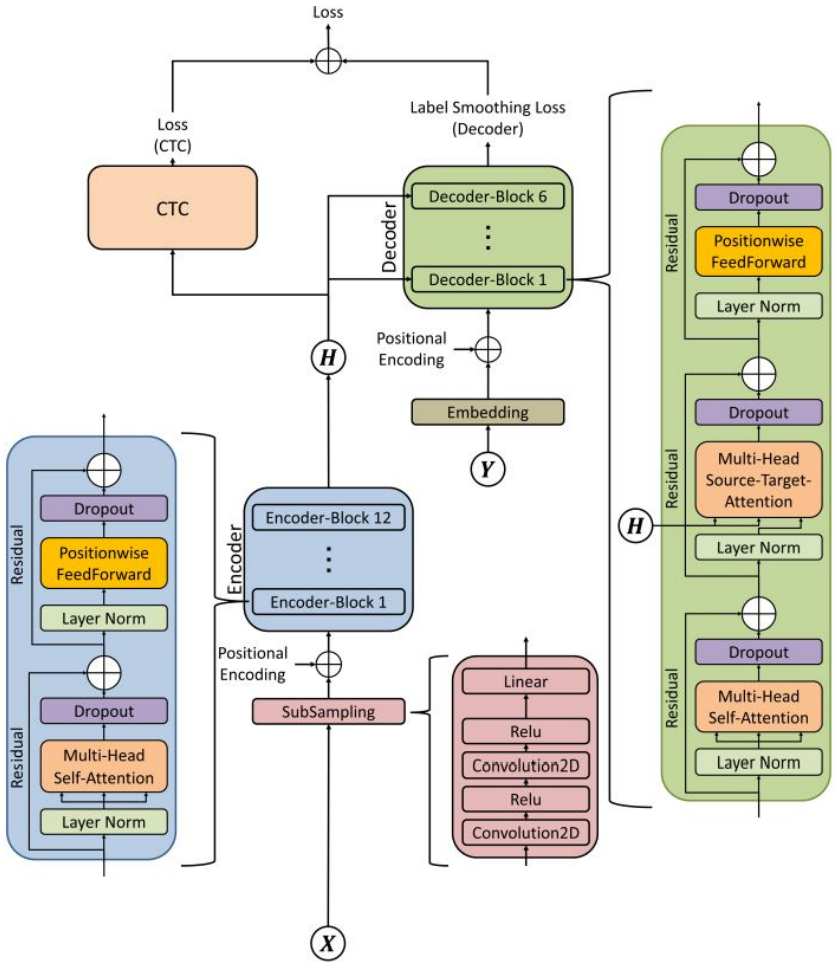


Figure 2.12: Hybrid end-to-end model structure using transformer [52]

target attention, and L is the loss of the whole model, which is the weighted sum of the loss of CTC and the loss of the decoder with the constant λ. In the hybrid approach, we use the same Encoder layer for the CTC and Decoder layers. This back-propagates the temporal constraint of CTC to the Encoder layer, so that source-target attention is not confused with the same phoneme at a temporally distant position. In this research, we use the Transformer model [53], which adds a convolutional layer to transform the speech feature sequence before the Encoder layer. Figure 2.12 shows the base model structure. In the base model, the speech feature sequence X is first transformed into a higher-order speech feature sequence by two convolution layers and a linear transformation layer. Furthermore, time-series information is given by using Positional Encoding. As the Encoder layer, we use 12 blocks that combine Multi-Head Self-Attention of multiple heads and Position-wise

Feed Forward to generate the intermediate feature sequence H. The intermediate feature sequence H is used for both the CTC input and the decoder layer input.

In the decoder layer of the base model, the embedding representation of the label sequence Y is used in addition to the intermediate feature sequence H. Time-series information is added to the embedded representation using positional encoding. In the Decoder layer, the label feature sequence is transformed by using self-attention of multiple heads, and it is set as Query, enabling it to extract intermediate features related to labels. The extracted intermediate features are transformed using a feed-forward network for each position. By using 6 layers of this block in the decoder layer, the probability distribution Y of the final output label is generated.

In the base model, the loss obtained from the output label probability distribution obtained from the CTC and the loss obtained from the output label probability distribution obtained from the Decoder layer are weighted and summed using a constant λ. In this research, we use $\lambda = 0.3$. To avoid overfitting and vanishing gradient problems during training, we combine Layer Norm [57], Dropout [56], and Residual layer [55]. Also, in the Decoder layer, overfitting is avoided by using a method called label smoothing that sets the probability of non-correct labels to a small value rather than 0. In this research, we use 0.1 as a constant for label smoothing. That is, (probability of correct label) : (probability of other label) = 0.9 : 0.1. End-to-end models trained with this approach are called **CTC/Transformer-based E2E** models hereinafter.

2.4 Self-Supervised Learning-based E2E ASR Modeling

In recent years self-supervised learning-based ASR modeling method has emerged which involves self-supervised learning (SSL) based learning representation extractor for audio encoder replacing the conventional feature input to the acoustic model [58–63]. This approach can be applied to an end-to-end-based method implicitly or explicitly. SSL-based neural networks produce pseudo labels for unlabelled data and they are later used to train the model similar to supervised approach. SSL

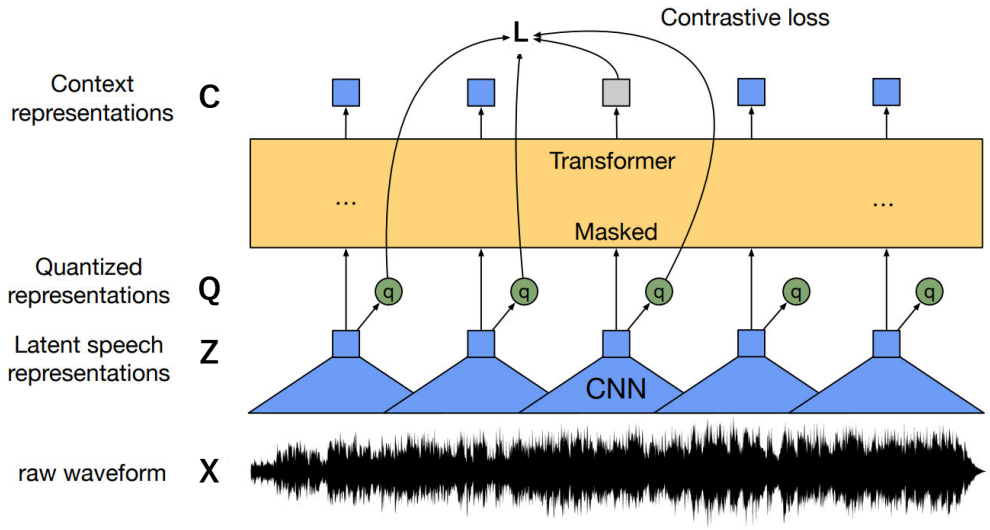


Figure 2.13: Architecture of self-supervised learning-based model wav2vec2.0 [59].

approaches are also considered as intermediate stage of unsupervised and supervised learning.

wav2vec2.0 [59] architecture is used as the self-supervised learning (SSL)-based audio encoder in this research. Figure 2.13 depicts the architecture of wav2vec2.0 audio encoder. wav2vec2.0 model comprises of feature encoder layer, transformer-based encoder layer and quantization layer. The feature encoder layer takes raw audio $\boldsymbol{X}$ and outputs feature representation $\boldsymbol{Z}$ of $\boldsymbol{T}$ frames for the transformer layer to use as input. The encoder layer of transformer produces contextualized representation $\boldsymbol{C}$ of $\boldsymbol{T}$. At this time, a proportion of feature representation is masked. Quantization module produces quantized representation $\boldsymbol{Q}$ from $\boldsymbol{Z}$.

The overall training loss L is the weighted sum of contrastive loss L_m and diversity loss L_d. The loss is represented by the following formula

$$L = L_m + \alpha L_d \tag{2.38}$$

where α is a tuned hyper parameter. Contrastive loss L_m is expressed by the following equation.

$$\mathcal{L}_m = -\log \frac{\exp\left(\mathrm{sim}\left(\mathbf{c}_t, \mathbf{q}_t\right)/\kappa\right)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp\left(\mathrm{sim}\left(\mathbf{c}_t, \tilde{\mathbf{q}}\right)/\kappa\right)} \tag{2.39}$$

here the cosine similarity $\text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T\mathbf{b}/\|\mathbf{a}\|\|\mathbf{b}\|$ is computed between context representation and quantized latent representation. Also, diversity loss L_d is represented by the following equation.

$$\mathcal{L}_d = \frac{1}{GV}\sum_{g=1}^{G} -H\left(\bar{p}_g\right) = \frac{1}{GV}\sum_{g=1}^{G}\sum_{v=1}^{V}\bar{p}_{g,v}\log\bar{p}_{g,v} \tag{2.40}$$

diversity loss is minimized to increase the use of quantized entries V in codebook G.

Pre-trained models are fine-tuned for speech recognition by adding a randomly initialized linear projection on top of the context network into $\boldsymbol{C}$ classes to represent the vocabulary of the task. ASR task is optimized by minimizing a CTC loss.

CHAPTER 3

DOMAIN ADAPTATION OF ASR MODELS AND DATA AUGMENTATION USING LIMITED RE-RECORDED SPEECH

This chapter describes the proposed method for domain adaptation of ASR models along with proposed data augmentation method using limited re-recorded speech. First, the background and task setting of the research is introduced in Section 3.1. The data acquisition using re-recording-based method is describes in Section 3.2. The problems regarding acquired data and the solutions are proposed in Section 3.3. The proposed domain adaptation methods of ASR for real environment re-recorded data are described in Section 3.4. Section 3.5 and Section 3.6 describe the experimental setup and discussion of results respectively. Finally, the idea and the findings are summarized in Section 3.7.

3.1 Introduction

The general theory of ASR is that acoustic models and language models or the end-to-end ASR models need to be trained with relevant and adequate amount of data to achieve the best performance [1–4]. However, real-world recording media are bound to vary in terms of recording conditions, transmission channels, etc. The performance of ASR models degrade significantly when evaluated in mismatched new domain. It is important to prepare enough data for each condition with proper

transcription for training an acoustic model suitable for the environments accordingly. However, acquiring and preparing such data can be costly and time consuming. Therefore, domain adaptive technologies are adopted to solve this problem.

In this chapter, we particularly take interest in recognizing data recorded using mobile telephone channel over fourth generation (4G) cellular network as well as data recorded at university classroom. This real-world telephone and classroom situation is simulated by re-recording a small amount of data in classroom by playing through loudspeaker and recording them using telephone devices or low-quality wireless microphone. Previous research on supervised training of ASR indicates the requirement of large-scale transcribed data in target environment. However, it is costly to record and transcribe such amount of data for desired environment. Therefore, we adopt DNN-based data augmentation method for DNN-HMM hybrid and end-to-end ASR models.

Apart from the difficulty in domain adaptation, we face a challenge when performing frame-by-frame training of feed-forward NN using re-recorded data. Analysis shows temporal distortion in re-recorded speech data. We therefore propose temporal alignment adjustment for each recording and filtering method for eliminating utterances with internal distortions.

There are previous research on how to improve the ASR model's performance for real data. According to them, one method to achieve improvement is to perform domain adaptation. To perform domain adaptation for real-world data, data augmentation is often used. Previous researches on this subject show data augmentation by adding various recorded noises at desired SNR level or performing speed perturbation on clean data or using room impulse responses (RIR) to simulate the desired room acoustics. These augmented data contribute in producing improved performance of ASR systems for the target domain [5–9]. Also, there are DNN-based data augmentation methods which extract acoustic information to represent acoustic environments instead of creating simulated data [10]. Also, sometimes DNN-based methods are applied, for example, variational autoencoder (VAE)-based data augmen-

tation approach which creates source-to-target and target-to-source data to increase the amount of training data [11].

Also, there are researches on adapting Gaussian-based system for dysarthric speech, when collecting data for the target domain speakers is difficult [12]. However, there are a few researches that involves domain adaptation for target domain with neural network(NN)-based ASR. For NN-based approaches, most of the examples involve feature normalization, such as cepstral mean variance normalization (CMVN), applying vocal tract length normalization (VTLN) for speaker adaptation, or feature transforms such as feature space maximum likelihood linear regression (fMLLR), etc. [13]. There are research about increasing amount of data by applying data augmentation approaches that creates large amount of data of target domain [7, 9, 15, 16]. Apart from data augmentation, domain adaptation is also important for fine-tuning models for limited target domain data where the full model fine-tuning and partial fine-tuning are introduced [17–24].

In this chapter, a novel method of data augmentation is proposed that helps to imitate real environment by using a DNN-based regression model for model-independent domain adaptation of DNN-HMM and end-to-end ASR models [25]. Similar approach of training DNN-based regression model has been adapted to perform denoising or dereverberation as pre-processing [26]. To the best of our knowledge, such small sized data have not been taken into consideration yet to perform regression-based training of feature mapping. Therefore, in this research, we train the regression model to learn on pairs of clean-rerecorded data of very small size (553 utterances). Experimental results show that it successfully transforms clean input data to real environment output data by regression-based mapping of feature called feature transformation in this dissertation. Since it learns to transform feature directly from real environment data, the computation cost is also minimal. The augmented data that prepared in this way are then used to perform fine-tuning of acoustic models. To acquire the best result, some of the original re-recorded speech are used along with the transformed features. Therefore, paired data with various

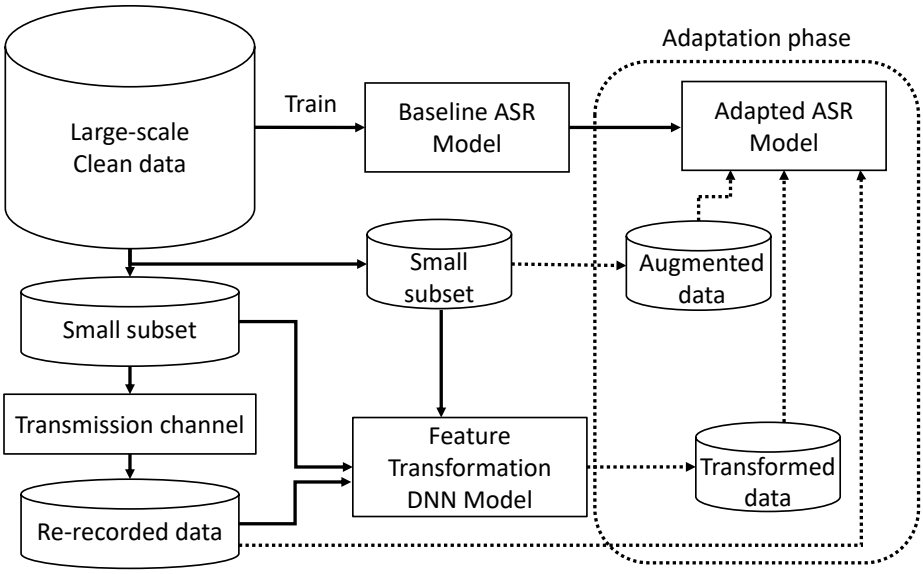


Figure 3.1: Schematic diagram of task setting for domain adaptive fine-tuning of ASR model with augmented data.

recording sources are an important part of this research. The task setting is depicted in Figure 3.1.

We focus on training the feature transformer DNN and fine-tuning the ASR models with real data as small as possible by controlling the duration of core re-recorded speech data. We observe that the fine-tuned ASR models start converging with only 30 min of core data. As a common pre-processing technique, we use CMVN for each speaker for training DNN-HMM hybrid model and for all the test scenarios. Global CMVN is used in case of end-to-end models.

3.2 Acquisition of Real Environment Speech and Pre-processing

To perform real-world speech recognition, we often face data scarcity problem due to lack of resourceful data. Therefore, to solve this problem, we propose an approach of robust automatic speech recognition (ASR) which utilizes re-recorded speech for domain adaptation of ASR model and pre-processings for re-recorded data with temporal misalignment.

In previous study, it shows that only two minutes of data per speaker is enough for feature transformation when there is a large number of speakers using statistical-based method (fMLLR) [14]. It uses over 150 hours of transformed feature supervised adaptation of DNN model for speaker identification and speech recognition. To perform acoustic domain adaptation, small amount of data is sufficient. Therefore, in this research, the duration of re-recording data is kept short and feature transformation and adaptation experiments are performed using datasets of even smaller duration (0.2h, 0.5h, 1h, 1.5h).

3.2.1 Re-recording of Clean Data in Real Environment

Training an acoustic model to adapt to a particular environment requires acquiring data from the environment as much as possible. This problem may seem to be solved by recording data at convenience. However, this simple solution does not work since it also requires for the training data to be precisely segmented and transcribed for conventional supervised learning-based acoustic models. Segmenting and transcribing manually is extremely time consuming and costly. Semi automatic way of performing such pre-processing – applying voice activity detector (VAD) for segmentation and decoding the speech using an already trained acoustic model– may help to some extent but it lacks reliability and needs additional attention by humans. Therefore, we try to solve the problem by re-recording clean data provided by a trusted speech corpus in various real environments, such as classrooms, and telephone channels as shown in Table 3.1. In this way, it is possible to acquire paired data to perform

Table 3.1: Recording devices and transmission channels for re-recorded speech.

Re-recorded dataset	**Recording device (mic)**	**Channel**	
		Caller	**Reciever**
Landline	Landline	Landline	Landline
Mobile 3G	Mobile	SoftBank 3G	SoftBank 3G
Mobile LTE[1]		SoftBank LTE	Landline
Classroom[2]	High quality pin-mic	2.4 GHz digital wireless	
(wireless pin-mic)	Low quality pin-mic	800 MHz analog wireless	

various experiments by training acoustic models using different transmission channels' conditions. We take inspiration of re-recording an existing dataset from NTIMIT [75] and CTIMIT [76] corpora. They also re-recorded from the original TIMIT acoustic-phonetic speech corpus [77] to utilize the existing transcription for speech recognition task. They are the telephone channel and cellular channel recordings of TIMIT dataset in early 1990s. While recording, they also stumble upon unknown noise or artifacts those are not easy to explain. Therefore, they take some filtering approach as pre-processing to mitigate the effect. They also pay attention to acquire perfectly aligned utterances with the original recording.

In our case, we follow the strategy to acquire data of Japanese language. In this dissertation, we discuss about telephone channels recordings[1] as well as wireless pin-mic recordings in classroom[2] environment. Even though this method of re-recording is less expensive than recording and transcribing newly recorded data, they yet need quite an attention. The recording device can cause low quality sound leading to low recognition performance as in classroom data. Also, lack of synchronization between playing and the recording devices can cause the data to have a misalignment problem as in Mobile LTE channel data. Therefore, we handle the worst performing re-recordings in this dissertation. Also, the misalignment problem needs to be fixed. There are measures to handle distortions [78] in data. We develop Euclidean distance-based alignment correction method. Temporal misalignment of re-recorded data and the method to correct it are described in following subsections.

[1]Figure 3.6 and Figure 3.7: Re-recording through telephone channel.
[2]Figure 3.2 and Figure 3.3: Re-recording through wireless pin mic channel.

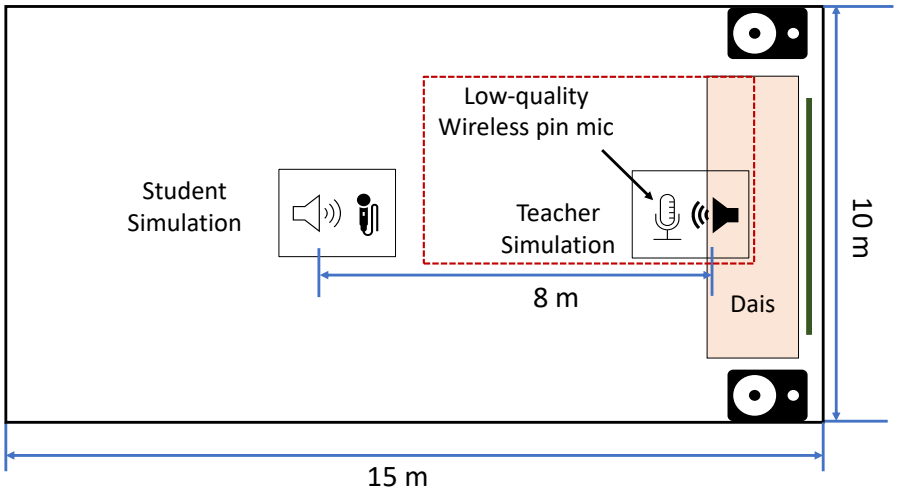


Figure 3.2: Schematic diagram of Re-recording setting for wireless pin mic channel in "Classroom" dataset recording.

Figure 3.3: Loudspeaker and wireless pin mic setting in "Classroom" dataset recording [79]. **(a)** Closeup of loudspeaker and pin mic positioning, **(b)** Recording setting in classroom 5-24, Hamamatsu campus, Shizuoka University.

3.2.2 Re-recording of Wireless Pin mic Dataset in "Classroom"

The classroom data is recorded using wireless pin-mic by playing 10 monologue speech recordings from CSJ eval3 of about 1 hour 40 minutes. Figures 3.2 and 3.3 show the setting of loudspeaker playing recordings and the positioning of the low quality wireless pin mic. This setting is considered to simulate the situation of the teacher in a classroom.

The details of recording conditions and data are explained in the masters thesis by

Y. Wakiya [79]. According to the thesis, the recording level of the playing recording for re-recording is set to a constant by measuring the loudness of read sentences by people standing at the dais (simulating lecturer) using a digital sound level meter. However, depending on the recording channel, the recording level came out differently. The recording level statistics is shown in Figure 3.4. It depicts the low sound level of low quality pin mic as opposed to hand mic and high quality pin mic, though they are supposed to be used in recording in same condition. Figure 3.5 shows the word error rate (WER %) of each recording recorded by different recording channels at the same time in the same condition. However, the wireless low quality pin mic only is used to record in two sessions (Session1: 6 recordings, Session2: 4 recordings), the wireless low quality pin mic re-recordings show incredibly higher WER (%) for all of the recordings, especially for the recordings recorded in Session2. Therefore, it is concluded that even though an effort was taken to record all the recordings in the same condition, depending on adjustment and recording channel, the recording quality vary. Since the wireless low-quality pin mic gives the worst performance than wired hand-held mic or wireless high-quality pin mic, we choose to work on the wireless low-quality pin mic, "pin mic" hereinafter to improve its recognition performance.

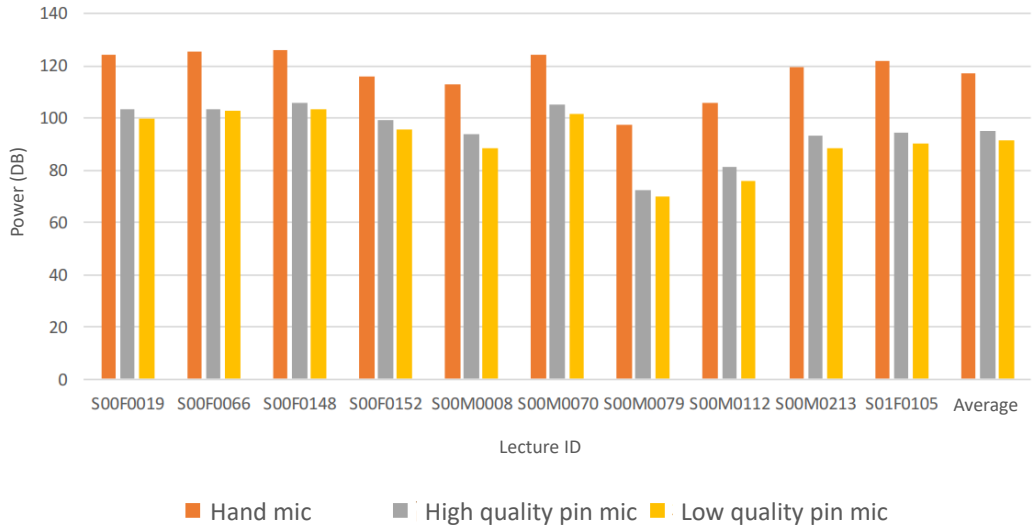


Figure 3.4: Statistic of recording level for each recording of wired hand mic, wireless high quality pin mic and wireless low quality pin mic in "Classroom" recording condition [79].

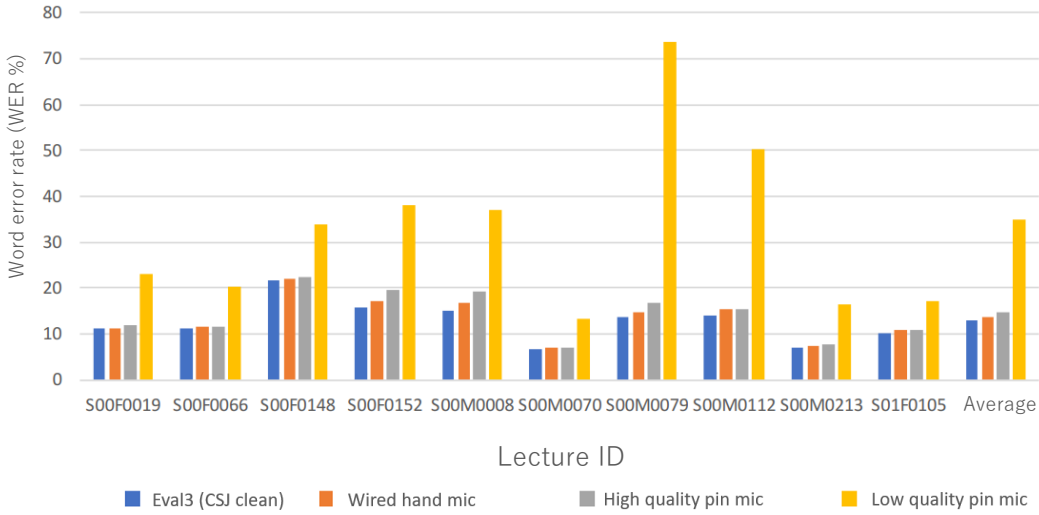


Figure 3.5: Comparison of word error rate (WER %) produced by DNN-HMM hybrid model trained with clean CSJ for Claean (CSJ eval3), wired hand mic, wireless high quality pin mic and wireless low quality pin mic in "Classroom" recording condition [79].

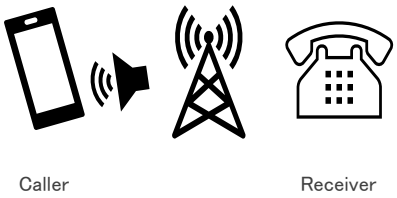


Figure 3.6: Schematic diagram of Re-recording setting for "Mobile LTE" channel dataset recording.

(a) (b)

Figure 3.7: Loudspeaker and mobile telephone handset setting in the receiver end for "Mobile LTE" dataset recording. **(a)** Closeup of loudspeaker and mobile handset positioning,**(b)** Recording setting in soundproof room.

3.2.3 Re-recording of Mobile LTE Dataset

For re-recording of dataset using telephone channel (Landline, Mobile 3G, Mobile LTE), a concatenated long recording of 2 hours 19 minutes consisting of ten monologue speech recordings from Corpus of Spontaneous Japanese (CSJ) [64], eval1 test dataset is played through speaker and recorded through different telephone channels as real-world test data. Mobile LTE data for training is also re-recorded in similar method in four sessions, each consisting of two hours of concatenated long recordings. Figures 3.6 and 3.7 show the recording setting of recordings played by loudspeaker and transmitted by a mobile telephone handset of 4th generation (4G) to a distant landline telephone. The player and loudspeaker are set inside a soundproof room. We prepared 26 recordings of about 7 hours from CSJ training set using Mobile LTE channel only for training purposes. These 26 recordings were grouped in three sessions each containing concatenated recording of about two hours. Due to failure in one session, we could use data of three sessions out of four.

Different playing and recording channels are used. Landline denotes intercom landline telephone of Shuzuoka University. Mobile 3G and Mobile LTE denote recordings using 3G and LTE mobile networks, respectively. SoftBank carrier is used to record mobile channel data. In the receiving end of the Mobile LTE recording, landline telephone is used to record the transmitted data which is located at call center in Tokyo[3]. The data is transmitted using the standard transmission method of LTE (4G) cellular network, called voice over long term evolution (VoLTE). In the receiving end a recording device embedded in landline telephone is used as recording terminal.

3.2.4 Spectral Analysis on Re-recorded Speech

Figures 3.8 and 3.9 show the long term spectra of re-recorded data through telephone and wireless microphone channels respectively as opposed to their original clean counterpart. We notice that the recording data through the mobile LTE channel has higher sensitivity at a lower frequency band than other recording channels. In this research, we focus on improving recognition performance for LTE channel (called mobile LTE hereinafter) and low-quality wireless pin-mic (called pin-mic hereinafter), two most troublesome speech to deal with.

[3]The recording is performed with the help of Mr. Takahiro Kunisaki, Nextgen Inc. Embeded recording terminal at call center is used for recording the transmitted data. Recording sample rate is 8kHz and bit-rate is 128k.

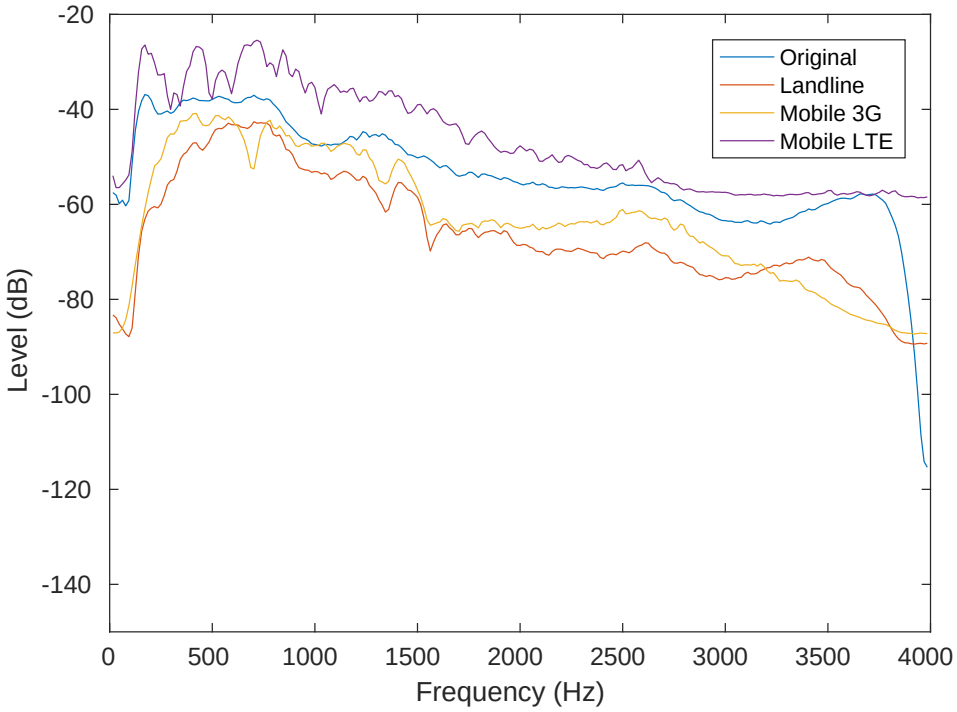


Figure 3.8: Spectral analysis of original and re-recorded speech using telephone channels (CSJ eval1 dataset).

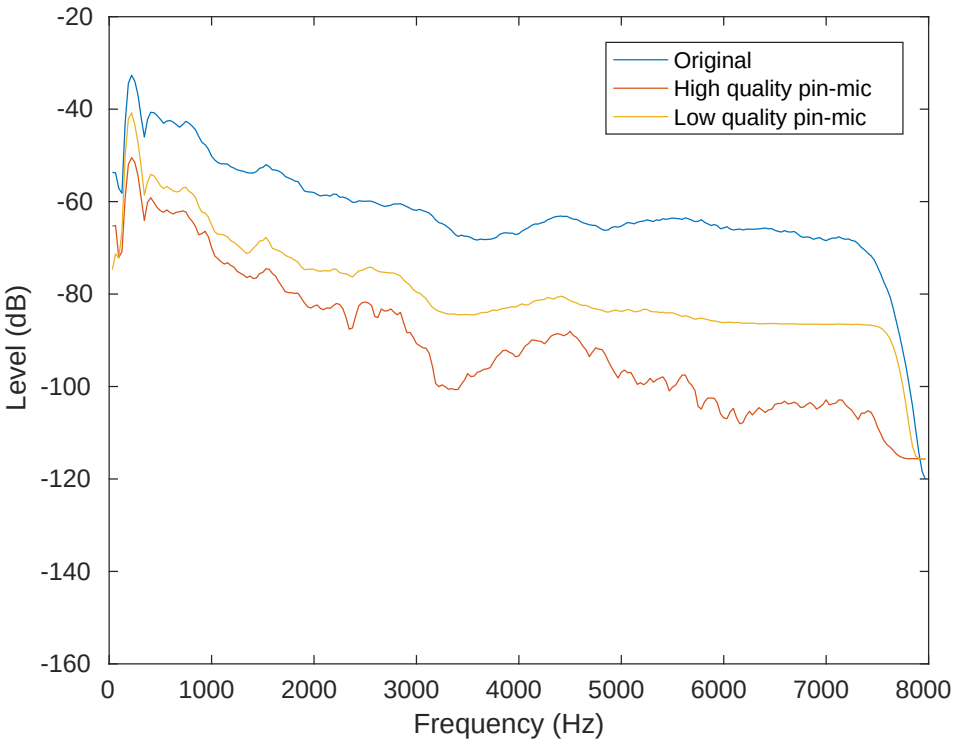


Figure 3.9: Spectral analysis of original and re-recorded speech using wireless pin-mic channels (CSJ eval3 dataset).

3.3 Problems Regarding Re-recorded Speech: Temporal Misalignment

The task of re-recording an audio is performed by playing the audio through a player device and recording the audio simultaneously using a recorder device. In case of telephone recording, transmission steps are involved between playing and recording. The start of the recording and playing time often lacks synchronization. Even though it may be possible to start playing and recording at the same time in ideal situation, there may be different reasons to cause timing mismatch between the pair of events. If the playing device and recording device lack clock synchronization, an incremental delay between estimated start and actual start with time is observed. When using cellular network (mobile phone) channel for transmission in Japan, a delay of up to about 400 ms, and also jitters can be experienced depending on the transmission network.

We performed a preliminary analysis on recordings those are re-recorded in a lecture hall using wireless pin-mic. An incremental delay of 20 ms in about every 10 minutes is noticed in Figure 3.10. An accumulated delay of 150 ms can be noticed at the starting of the 10th lecture. However, the temporal misalignment for telephone recording (CSJ-eval1) is not as simple. Figure 3.10 shows variable temporal misalignment throughout the re-recording period of 2 hours 19 minutes with the interval of duration of each lecture. Since the first estimation starts at the guide point, which is 15 seconds earlier than the starting of the first lecture, we can even see a few frames delay where the first lecture actually starts from the theoretically estimated starting in Figure 3.10. As this figure shows, time deviations sometimes exceed 200 ms, and similar time deviations were observed in terms of IPU segment units (units separated by silence greater than 200 ms in CSJ [64, 65]). Therefore, we propose a correction method in following sections.

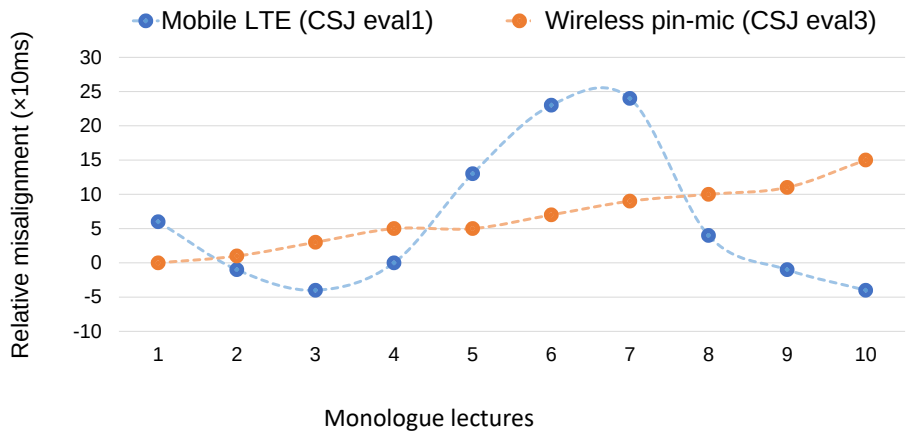


Figure 3.10: Misalignment analysis of re-recorded speech for mobile LTE and wireless pin-mic channels.

3.3.1 Misalignment Correction Method Based on Segment-level Matching with Euclidean Distance

To correct the misalignment, first, a rough starting point t'_{start} of the re-recorded speech is estimated. Then, a segment-level matching between the pair of segments $x_t ... x_{t+N-1}$ and $y_{t'} ... y_{t'+N-1}$ is performed. The optimal starting point $\hat{t}$ is estimated by finding the frame for which the average Euclidean distance is minimum,

$$\hat{t} = \underset{t' \in \{|t'-t| \leq D_{max}\}}{\arg\min} \left(\frac{1}{N} \sum_{n=0}^{N-1} d(x_{t+n}, y_{t'+n}) \right). \tag{3.1}$$

A frame consists of speech data of 10 ms.

Though there are different ways of measuring distortion between speech signals [78], we use Euclidean distance between MFCC features as distortion measurement. We calculate Euclidean distance between the feature vector of original speech at the t^{th} frame and the feature vector of re-recorded speech at the t'^{th} frame using Equation (3.2).

$$d(x_t, y_{t'}) = \sqrt{\sum_n (x_{t,n} - y_{t',n})^2}, \tag{3.2}$$

where n denotes the number of feature dimension. We assume that t' falls in the range $t - D_{max}, ..., t, ..., t + D_{max}$. D_{max} is the number of frames to search before and after each point of time.

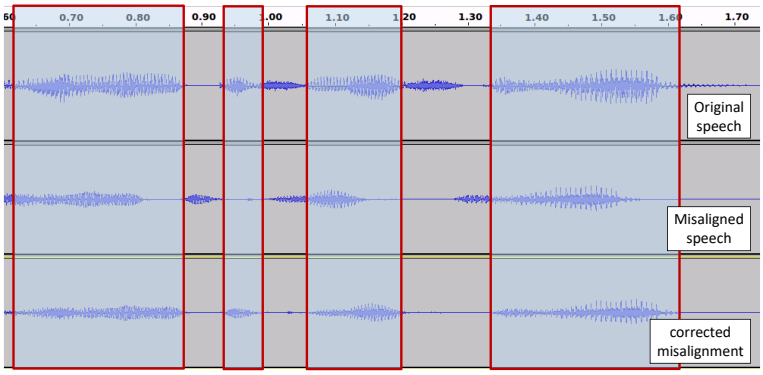


Figure 3.11: Misalignment correction with proposed Euclidean distance-based method.

We used a sine wave signal in front of each lecture to indicate starting of individual lectures at the time of concatenating them as a preparation of re-recording to ensure stable quality for the speeches as much as possible. Using the sine wave as guiding point, the starting point of a lecture is first guessed manually. Then the correct starting point is estimated automatically using the misalignment correction method described above. Then individual lectures are separated from the long re-recorded speech (segment length to compute Euclidean distance is $N = 200$ frames). In Figure 3.11, we show the misalignment correction at the starting point in visualized form for re-recorded speech.

3.3.2 Filtering out Misaligned Segments from Re-recorded Speech

We show in the Figure 3.12 that there are intra-recording misalignment in case of mobile LTE data despite correcting the initial temporal misalignment at the starting point using the Euclidean distance-based method described in Section 3.3.1. However, since we propose that we train a regression model to learn feature transformation from clean data to real environment data frame-by-frame, paired data of clean-target environment need to be prepared as much as accurately possible. Therefore, we filter out the segments from the re-recorded data, those do not match the corresponding original speech due to suffering from packet loss or delay caused by jitters. Figure 3.13 shows the flowchart of the filtering process adopted in this dissertation to filter out the misaligned utterances from the re-recorded telephone speech.

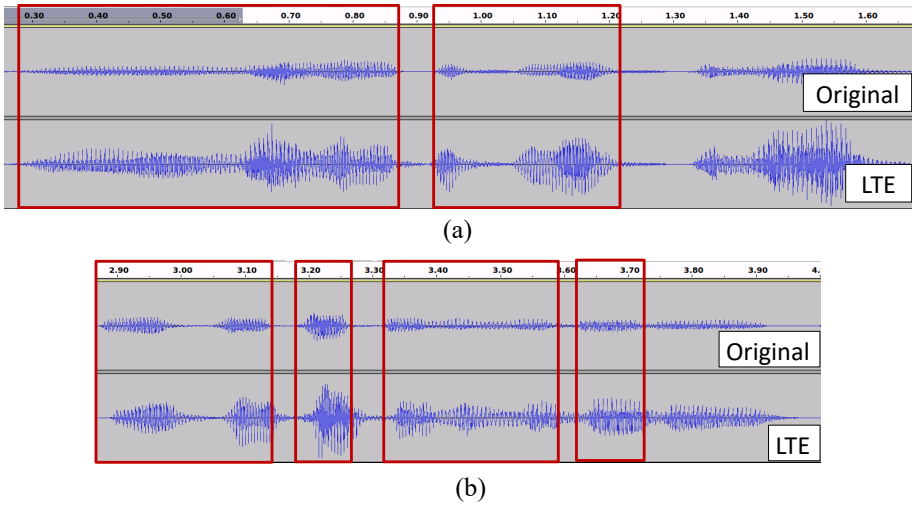


Figure 3.12: **(a)** Red rectangles represent aligned segments of original and mobile LTE re-recorded speech. **(b)** Red rectangles represent misaligned segments inside the re-recording those need to be filtered.

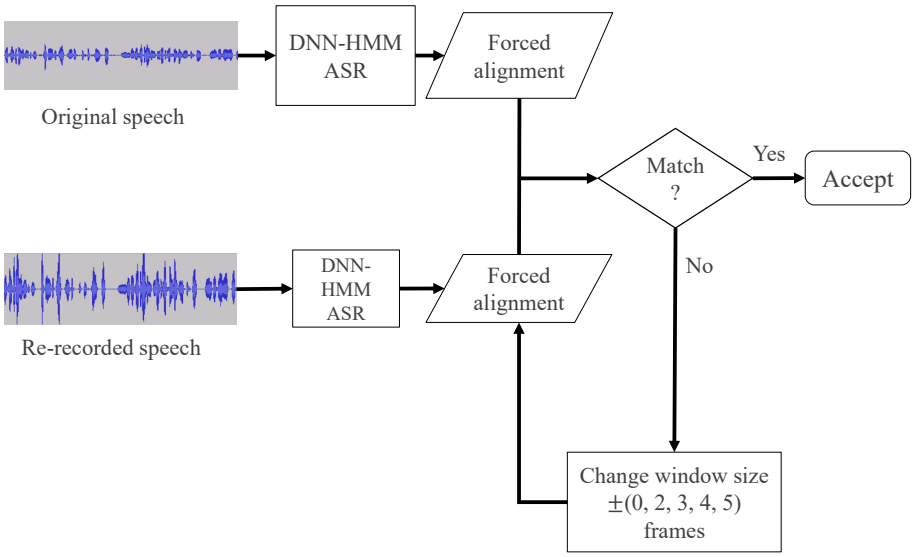


Figure 3.13: The flowchart of the filtering process of the misaligned utterances.

We first add 200 ms margin in the beginning and the end of each IPU segments [64, 65] of clean data and re-recorded data with estimated starting and ending time of IPU, then perform the forced alignment using DNN-HMM acoustic model. After acquiring forced alignments from both clean and re-recorded data, we remove silences and short-pause from the alignments. This gives us a pair of speech segments to

compare length-wise using their phoneme notations and temporal information. In our experiment, first we derive those segments which have exactly matching length for the pair. However this leaves us with a very small amount of data, about 53 minutes of data consisting of 291 utterances to be exact after being filtered out from CSJ training subset (4109 utterances of about 6 hours). Therefore, to increase the amount of data, we loosen the strictness of the filtering by allowing 2, 3, 4 and 5 frames (frame=10 ms) at the both ends of segments to find more matches in turn. In this way, we acquire 694 utterances of 2.1 h in total which is about 34% of the original recorded data.

3.3.3 Segment-level Adjustment vs. Filtering out Misaligned Utterances

In Section 3.3.1, when separating the concatenated recordings, recording-level misalignment is adjusted for the first 200 frames of each recording. The tail of the recording is adjusted using the reference length of the recording file. The utterances that have internal misalignment are filtered out using the filtering algorithm described in Section 3.3.2. In this method, the number of extracted utterances reduce dramatically.

Euclidean distance-based segment-level adjustment is also possible to adopt to correct misalignment. In this way, almost 100% of the available data may be used for further use. On the other hand, filtering out is performed to keep internally aligned utterances. Filtering out not only helps with preserving utterance length, but also helps to preserve the content, since it is based on the phoneme sequence of the utterance extracted by DNN-HMM ASR model. Euclidean distance-based adjustment method can cause data loss at the end of the utterance since it is based on physical information only, such as duration, etc. Therefore, in this research, filtering out misalignment is adopted over segment-level adjustment.

3.4 Proposed Domain Adaptation of ASR Models

3.4.1 DNN-based Data Augmentation to Tackle Data Scarcity in Target Domain

In this research, we propose a simple data augmentation method that uses simple feed-forward DNN regression model. The hypothesis is that performing non-linear feature transformation of clean data that have enough transcription data to the target domain by using a DNN model trained using clean-target domain paired data.

We train a simple architecture of feed forward DNN to perform frame-by-frame non-linear transformation for features from clean data to simulate real-world recording-like characteristics. This model is denoted as "feature transformation" hereinafter. The feature-transformation model takes d-dimensional clean feature vector at frame t with a context of c frames before and after the central frame $X_t = x_t - c, ..., x_t, ..., x_t + c$ and outputs y_t after performing feature transformation.

$$y_t = f_L(...f_l(...f_2(f_1(X_t)))), \tag{3.3}$$

where f_l is the non-linear transformation function in layer l and y_t is a d-dimensional feature vector. The training of this DNN model is performed by optimizing mean square error (MSE) objective function to predict feature vectors of corresponding re-recorded speech. Figure 3.14 shows the block diagram of feature transformation task.

In this process, we are able to prepare sufficient target domain data with proper transcription by transforming clean data. The DNN is capable of learning the differences between pairs of clean and the contents collected in other domains.

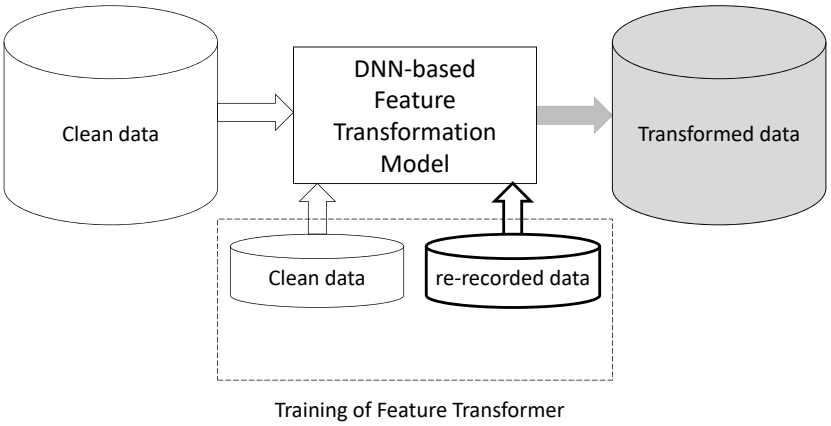


Figure 3.14: DNN-based feature transformation model for data augmentation.

3.4.2 Domain Adaptation of ASR Model using Augmented Data for Target Domain

We take various data augmentation approaches to create the most suitable dataset for training the robust baseline so that it produces the lowest character error rate (CER%) for the test data with real-world acoustic aspects. To create the datasets for training ASR model, we take general approaches adopted for data augmentation and noise-robust training as noted in Table 3.2. First, we apply μ-law encoding to the clean data to simulate landline quality telephone channel distortion. The μ-law encoding is a companding algorithm used in 8-bit pulse come modulation (PCM) digital communication systems in North America and Japan. In a later section in experimental setup, we show that the data with μ-law encoding gives better result than not using it for the target domain. Therefore, we use μ-law encoding for all of the baselines. We create another dataset to train Base-Aug3CN-ASR in Table 3.2, which contains speed and volume perturbed clean data with noisy data that contains G.712 filtering. This dataset was created to improve baseline performance of the Landline telephone speech. Because of the increased size of clean data, not only it improves performance for Landline, but also for clean test set. We create another dataset that does not contain any clean data to train Base-Aug3N-ASR

Table 3.2: Dataset for training baselines. Notations: S1 = Speed perturbation (0.9, 1, 1.1), Volume perturbation (0.7 - 1.5). The notation of "ASR" is replaced in the following section depenting on the type of model used (ASR=TDNN or E2E). Size denotes the amount (×233h) of data.

Model	Clean				Noisy				Total size (×233 h)
	μ-law encoding	Speed pert.	Vol pert.	Size	μ-law encoding	Speed pert.	Vol pert.	Size	
Base-NoAug-ASR	✓	-	-	1	-	-	-	0	1
Base-Aug3CN-ASR	✓	S1	✓	3	✓	-	-	1	4
Base-Aug3N-ASR	-	-	-	0	✓	✓	✓	3	3

baseline models showed in Table 3.2. We create this dataset considering the noisy characteristics of the re-recorded test data.

To improve the performance by fine-tuning, we needed to start from an elevated platform. Therefore, keeping real-world scenarios in mind, we perform simulation based data augmentation. Speed perturbation is used not only to increase the amount of data, but also it give us two different pitches for each speaker which somewhat simulates the effect of increasing number of speakers. Therefore, we obtain three times larger and diverse dataset with same content giving us privilege to use the same transcription for the supervised training of ASR models. Adding volume perturbation allows us to simulate different vocal levels or quality of device. By adding noise, we make the baselines robust to the common noises those can be experienced in common indoor and outdoor scenarios.

We first experiment on various combinations of data augmentation conditions for the fine-tuning dataset considering the composition of the baseline datasets. We find the combination of simulation conditions to fine-tune Base-Aug3N-ASR (Other baselines did not produce desired result) which produces the best result for most of the cases by carrying out experiments. Therefore, we propose simulation and conditions of dataset for the training adaptation-based experiments. Note that the core clean data for fine-tuning dataset used in our domain adaptation experiments is very small compared to the baseline model. Therefore, using more variations of clean data balances it when re-recorded data and transformed features are used together. We have also conducted fine-tuning experiments with speed and volume perturbed re-recorded data to see the effect of increased re-recorded data on domain adaptation.

Table 3.3: Dataset for fine-tuning of Base-Aug3N-ASR. Notations: FT= Fine-tuning of Base-Aug3N-ASR models, L = Adapted by LTE re-recordings, P = Adapted by pin-mic re-recordings, S1 = Speed perturbation (0.9, 1, 1.1), S2 = Speed perturbation (0.8, 0.9, 1, 1.1, 1.2), V = Volume perturbation (0.7 - 1.5), F = G.712 Filter, T = Transformed features. The notation of "ASR" is replaced in the following section depending on the type of model used (ASR=TDNN or E2E). Size denotes the number of seed sized dataset used for fine-tuning. seed(L)={0.2, 0.5, 1, 1.5} h; seed(P) = $\approx$ 1.2 h.

Model	Clean (μ-law, S1, V)	Noisy (μ-law, F)	Re-recorded (μ-law)			Trans.		Total size $\times$seed
	Size	Size	SP.	Vol.	Size	T	Size	
FT-L-ASR	3	1	-	-	1	-	0	5
FT-LT-ASR	3	1	-	-	1	✓	1	6
FT-Aug3L-ASR	3	1	S1	✓	3	-	0	7
FT-Aug3LT-ASR	3	1	S1	✓	3	✓	1	8
FT-Aug5L-ASR	3	1	S2	✓	5	-	0	9
FT-Aug5LT-ASR	3	1	S2	✓	5	✓	1	10
FT-P-ASR (no μ-law & F)	3	1	-	-	1	-	0	5
FT-PT-ASR (no μ-law & F)	3	1	-	-	1	✓	1	6

In Table 3.3, since the first six ASR models are intended for fine-tuning for the telephone channel, we use μ-law encoding and filtering with the noisy data. However, FT-P-ASR and FT-PT-ASR are intended for wireless pin-mic domain adaptation. Therefore, we do not apply μ-law encoding or any filtering on the fine-tuning datasets.

3.5 Experimental Setup

3.5.1 Datasets

In this section, we gradually describe the datasets and their preparation for training the baselines and fine-tuning them. First, we prepare datasets to train three baseline models for both DNN-HMM ASR and E2E ASR. Then the data prepared for fine-tuning are described. The proposed fine-tuning dataset contains an element called transformed features (Trans. in Table 3.2 and Table 3.3). The transformed features are extracted using proposed feature transformer model. Dataset for training feature transformer model is explained in Section 3.5.1. The experiments are performed using subsets of the Corpus of Spontaneous Japanese (CSJ) [64, 65] with sampling rate of 8 kHz.

Datasets for Training Baselines

We prepare datasets with clean only and multiconditional data for training three baselines for DNN-HMM ASR (Base-NoAug-TDNN, Base-Aug3CN-TDNN, Base-Aug3N-TDNN) and E2E ASR (Base-NoAug-E2E, Base-Aug3CN-E2E, Base-Aug3N-E2E). The data set used to train Base-NoAug-TDNN and Base-NoAug-E2E is 948 academic lectures of CSJ of duration 233 h is the base for all the baseline training datasets. This dataset consists of lectures from 141 female and 807 male speakers. Training dataset for Base-Aug3CN-ASR contains total 933 h of data consisting of three parts clean data with speed (0.9, 1, 1.1) and volume (factor: 0.7-1.5) perturbation and one part noisy data created with additive noises chosen from a subset of the noise database "JEIDA-NOISE" [66]. The noise types used are exhibition booth, crowd, computer room (medium), computer room (workstations), air conditioner (large), exhaust fan, and air duct. The noises are selected and added randomly with a random SNR over 5 to 20 dB with a 5 dB interval. G.712 filter [80] is used to distort the noisy data for telephone channels. Therefore, dataset used to train Base-Aug3CN-TDNN and Base-Aug3CN-E2E contains four times of the clean

Table 3.4: telephone speech: Character error rate (CER%) of TDNN and E2E ASR models trained by data with or without μ-law encoding.

Model	μ-law encoding	Test dataset (CSJ eval1)			
			Re-recorded		
		Clean	**Landline**	**Mobile 3G**	**Mobile LTE**
Base-NoAug-TDNN	×	9.5	11.1	24.4	31.5
Base-NoAug-TDNN	✓	**9.4**	**11.0**	**23.6**	**30.6**
Base-NoAug-E2E	×	**6.2**	**6.8**	15.3	20.6
Base-NoAug-E2E	✓	6.3	7.0	**14.4**	**19.5**

dataset of duration 933 h. The multiconditional dataset of 700h, used to train the Base-Aug3N-TDNN and Base-Aug3N-E2E models contain three parts noisy data prepared by applying speed and volume perturbation, noise and filtering. Therefore, this dataset does not contain any clean data. All of the above datasets are encoded using 8-bit μ-law encoding. We decide to apply μ-law encoding on every dataset by comparing the performance of Base-NoAug-TDNN and Base-NoAug-E2E trained by dataset with and without μ-law. We compare them in Table 3.4.

Base-NoAug-ASR trained with TDNN by using clean data of 233h with μ-law encoding performs better for every kind of test dataset, despite of aiming for only mobile variations. We infer that it performs better for clean and landline too because of the difference in models caused by random initialization. Since we get better results for mobile variations by E2E model trained with data containing μ-law encoding as expected, we decide to apply μ-law encoding on all of the datasets to perform further experiments.

Datasets for Domain Adaptation

We have 26 re-recorded lectures, which is a subset of CSJ training dataset (948 lectures) for LTE domain experiments in total for training purposes using training subset of CSJ corpus. We use 9 recordings from them to perform fine-tuning for LTE domain (Table A.1[4]). These nine recodings include two recordings from male seven recordings from male speakers. The fine-tuning dataset contains three parts of clean data with speed and volume perturbation, one part noise and filtering, one

[4]Appendix Section A.1

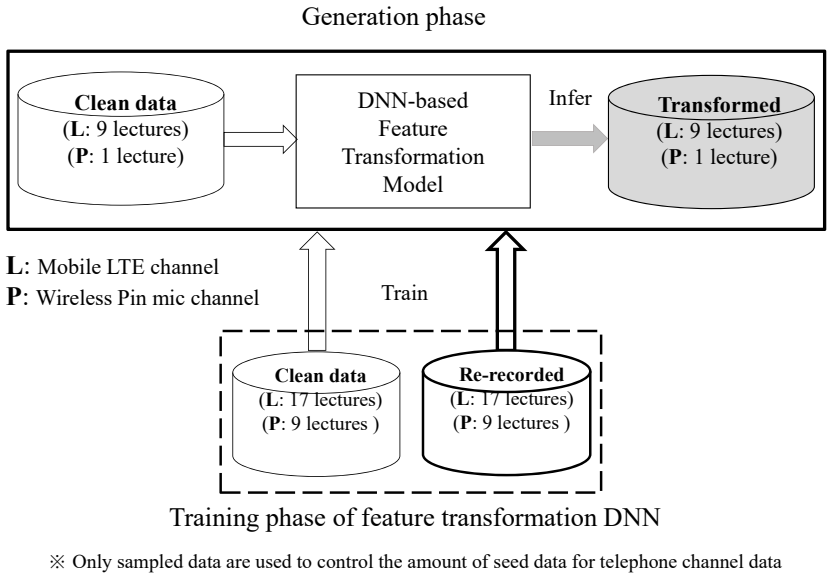


Figure 3.15: Training of DNN-based feature transformation model.

part re-recorded speech and another same combination with one part transformed features of the same clean content for the matching domain. We change the amount of seed data from 0.2 to 1.5 hours for the experiments and compare the results for validating the proposed technique. The term "seed" here denotes the core clean data that is used for data augmentation.

For the wireless pin-mic data in classroom scenario, we only use the 10 clean recordings of Eval3 test dataset of CSJ and re-record them in the said condition (Table A.4). Since we only acquire ten recordings of 10 speakers (5 male, 5 female speakers), a total of 1.32 hours, we use 9 of the recordings (≈1.2 h in average) for fine-tuning, leaving one recording to perform testing. We repeat this process 10 times to do 10-fold cross-validation for all of the speakers. We do not apply μ-law encoding or G.712 filtering representing telephone channels on simulated data of fine-tuning dataset for wireless pin-mic.

Dataset for training Feature Transformation Model

We train the feature transformation as depicted in Figure 3.15 using the 17 excluding 9 recordings those are used for fine-tuning completely (Table A.2). We use only 553 utterances at most for training the feature transformation model those we obtain by

filtering out temporally mismatched utterances from 17 pairs of clean-re-recorded data. We prepare 5 sets of training pairs with duration 0.2, 0.5, 1.0, 1.5 hours (equivalent to the size of seed mentioned in Section 3.5.1)from the set by picking a subset of utterances from 17 speakers. We train these models for validating our proposed method. For comparing the performance in general, we use the largest seed of 1.5 h.

In case of training feature transformation for wireless pin-mic, we use the same nine pairs of recordings those are explained in the paragraph above. We train 10 feature transformation models for 10-fold cross-validation.

3.5.2 Evaluation Tasks

Evaluation is performed on 10 recordings of eval1 (Table A.3)– clean and re-recorded variations of eval1– Landline, Mobile 3G and Mobile LTE for experiments related to telephone domain adaptation. "eval1" test dataset contains recordings of 10 male speakers only. Evaluation of wireless pin-mic channel speech recognition in classroom environment is performed using eval3 (Table A.4)– clean and wireless pin-mic re-recordings of it. As mentioned earlier, "eval3" dataset contains lectures from 5 male and 5 female speakers.

3.5.3 Explanation of models

DNN-based Feature Transformation Model

We train feed-forward type of DNN as feature transformation model which learns non-linear transformation for the input data to take it closer to the target data. We use DNN with different configuration for LTE and Pin-mic transformation. We decide the configuration after performing hyper parameter tuning. As described in Table 3.5, both of the models are trained using log-mel filter bank– F-bank features of 40 dimension and pitch feature of 3 dimension. We also use first derivative Δ, and second derivative $\Delta\Delta$ of the acoustic features as dynamic features. Per speaker cepstral mean variance normalization (CMVN) is performed to reduce the effect of

Table 3.5: The configuration of DNN-based feature transformation model.

Configuration	Feature transformation DNN models	
Target	Mobile LTE	Wireless Pin-mic
Input features	Filter bank (40), pitch (3), Δ, ΔΔ	Filter bank (40), pitch (3), Δ, ΔΔ
CMVN	Per recording	Per recording
Input nodes	1419 (context: ±5)	2193 (context: ±8)
Hidden layers	3	2
Hidden units	1024	1024
Output nodes	43	43

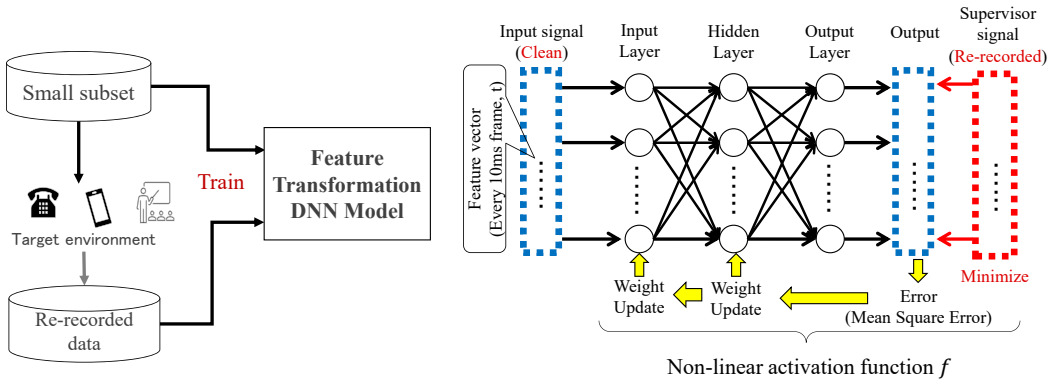


Figure 3.16: The architecture of DNN-based feature transformation model.

differences in input. For LTE feature transformation model, ±5 frames are used as context frames. Therefore, the input layer consists of 1419 nodes. Three hidden layers with 1024 hidden units in each are used. For pin-mic feature transformation model, ±8 frames are used as context frames. The input layer consists of 2193 nodes. Two hidden layers with 1024 hidden units in each are used. Both of the models give us 43 dimensions of transformed features as output. Figure 3.16 shows the architecture of DNN-based feature transformation model.

DNN-HMM ASR Model

We use a TDNN as the acoustic model of the ASR as one of the candidates. As described in Table 3.6, the baseliens are trained using 43 dimensions of F-bank-pitch features. Per speaker CMVN is performed on input features. This neural network

Table 3.6: The configuration of DNN-HMM hybrid ASR model.

Configuration	**DNN-HMM ASR Model**
Acoustic model	Time delay neural network (TDNN)
Input features	Filter bank (40), pitch (3)
CMVN	Per recording
Input nodes	473 (context: ±5)
Hidden layers	7
Hidden layer contexts	[-5,5], {-1, 2}, {-3, 3}, {-3, 3}, {-7, 2}, {0}
Output units	9225
Language model	Trigram

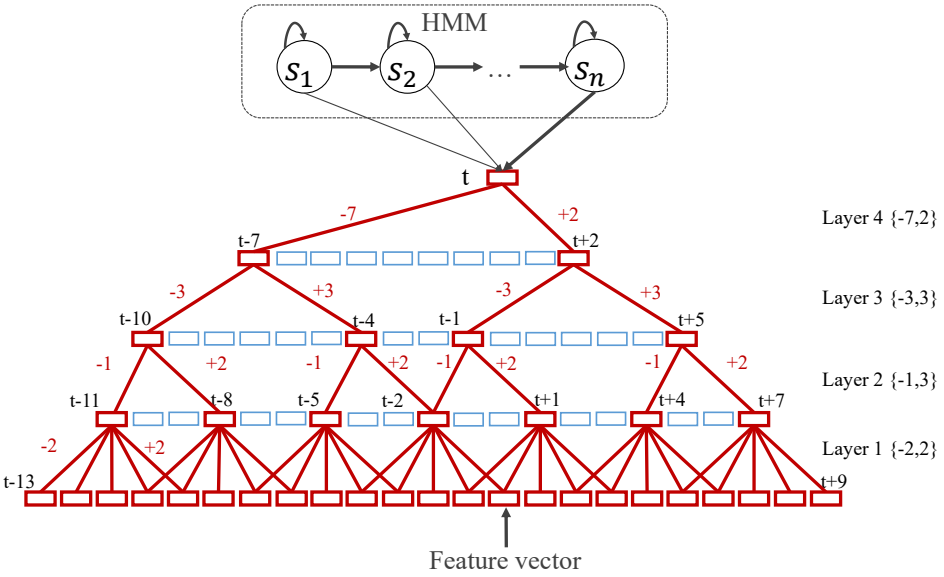


Figure 3.17: The architecture of DNN-HMM ASR model [40].

consists of seven hidden layers with following input context with sub-sampling: [-5,5], {-1,2}, {-3,3}, {-3,3}, {-7,2}, {0}. The output layer consists of 9225 units. A word trigram language model is used when decoding. Figure 3.17 shows the architecture and configuration of DNN-HMM ASR model. The baselines are then fine tuned using smaller amount of simulated and re-recorded data as well as including feature transformation-based augmented features with them. To perform experiments regarding feature transformation model and TDNN-based speech recognition, we use Kaldi toolkit for speech recognition [68, 69].

End-to-end ASR Model

Baseline hybrid CTC/Transformer-based end-to-end ASR models are trained using the baseline datasets described previously. As shown in Table 3.7, 43 dimensions of

Table 3.7: The configuration of CTC/Transformer-based end-to-end (**conv E2E**) model.

Configuration	CTC/Transformer E2E ASR Model
Input features	Filter bank (40), pitch (3)
CMVN	Global
Encoder	**Transformer layers: 12** Units: 2048 Sub-sampling unit: 2 convolutional layers Attention heads: 4; dimension: 256
Decoder	**Transformer layers: 6** Units: 2048
CTC loss weight	0.3
Output units	2865

F-bank-pitch features are used to train the transformer. Global CMVN is applied on the input features also, data augmentation method SpecAugment [82] is used on the input features. For performing SpecAugment, time warp (max 5), frequency masks (no. of masks: 2) and time masks (no. of masks: 2) parameters are used on the training data. The encoder part of the model has 12 layers, each consisting of 2048 units. The decoder consists of 6 layers with 2048 units in each of them. A sub sampling unit consisting 2 convolution layers in the encoder. It reduces the input length to one fourth. There are four attention heads with 256 dimensions. The wight of α for CTC loss is set to 0.3. Number of output unit is 2865 which corresponds to the number of different characters including Japanese characters. Figure 3.18 visualizes the architecture of CTC/Transformer-based E2E ASR model. Experiments of end-to-end ASR are performed using ESPnet, E2E speech processing toolkit [83].

3.5.4 Evaluation Metrices

We use Character Error Rate– CER(%) for evaluating the performance of ASR models. CER is denoted by the following equation.

$$CER = \frac{I + S + D}{N} \times 100 = \frac{I + S + D}{C + S + D} \times 100, \tag{3.4}$$

here I is the number of insertions, S is the number of substitutions, D is the number of deletions. C is the number of correct characters, N is the number of characters in the reference.

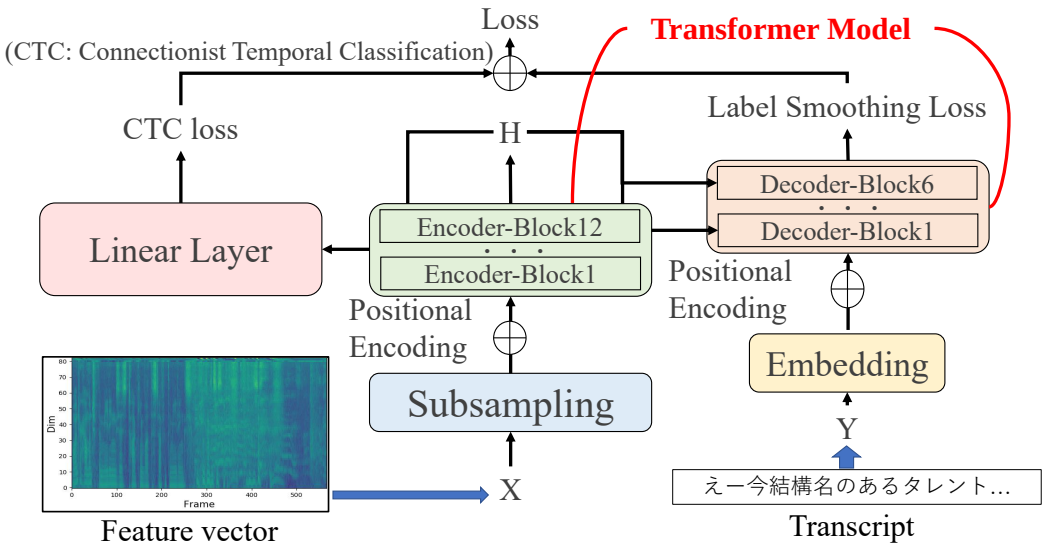


Figure 3.18: The architecture of CTC/Transformer-based end-to-end (conv E2E) model [84].

We also use character error rate reduction– CERR(%) which indicates the improvement when comparing CERs of multiple methods, CER1 and CER2. When CER2 improves from CER1, the CERR(%) of CER2 is calculated using following equation.

$$CERR = \frac{CER1 - CER2}{CER1} \times 100. \tag{3.5}$$

The smaller the CER (%), the better the performance is. On the other hand, the larger the CERR (%), the better the performance of the model is comparing to the relative model.

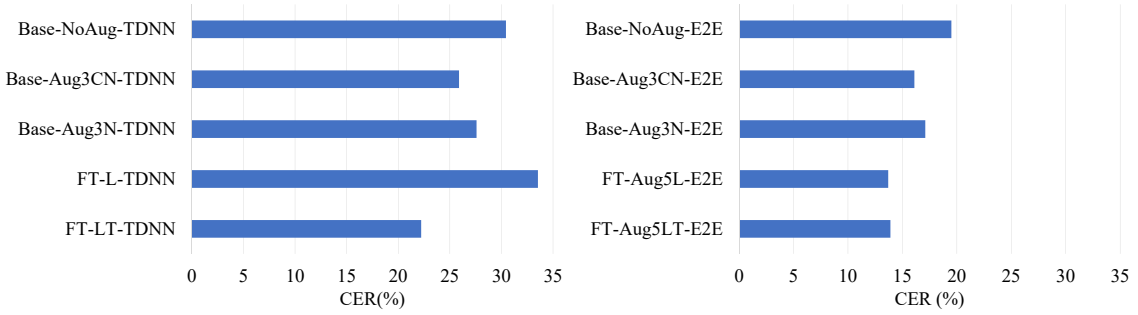
3.6 Results and Discussion

3.6.1 Results of Domain Adaptation for Mobile LTE and Pin mic Channel When the Largest Amount of Data are Used

In this section, we compare the baselines and proposed method of fine-tuning with transformed features for target domain. The starting point we consider is the Base-NoAug-TDNN and Base-NoAug-E2E for DNN-HMM ASR and end-to-end ASR, respectively. We gradually improve the performance by improving the baselines by adding various elements of real-environments by simulation. In the following results, the feature transformation model that is used is trained with the largest available data for training (1.5 hours of 17 recordings). We mainly propose this method for DNN-HMM-based system. To observe the method's performance, we apply the proposed fine-tuning techniques on end-to-end model as well. To prove its validity, we perform validation experiments and explain it in the following section with various amounts of data.

Theoretically, fine-tuning of the baseline closest to the target domain should gain better performance for the test set. Therefore, Base-Aug3N-ASR is bound to give better result when fine-tuned. However, we perform same adaptation experiments for the three baselines Base-NoAug-TDNN, Base-Aug3CN-TDNN and Base-Aug3N-TDNN. As expected, Fine-tuning of Base-NoAug-TDNN and Base-Aug3CN-TDNN don't give as satisfactory results as Base-Aug3N-TDNN, since they contain either full clean set or a three quarter of clean dataset at the primary training phase which make them more akin to clean data. Therefore, Base-Aug3N-ASRs are used for all the experiments afterwards.

In Figure 3.19, the improvements are shown with the converging character error rate for LTE channel adaptation of TDNN and E2E ASR. FT-L-TDNN and FT-L-E2E in the figures represent fine-tuning of Base-Aug3N-TDNN and Base-Aug3N-E2E respectively with speed and volume perturbed clean data, noisy data and accompanying re-recorded speech of 9 re-recorded lectures. For FT-LT-TDNN and



(a) TDNN: Domain adaptation for LTE

(b) E2E: Domain adaptation for LTE

Figure 3.19: Performance of Data augmentation on Telephone channel speech. Base-Aug3N-ASR models are used as the base for all the fine-tuned models (FT). Seed=1.5h is used as the seed amount of data for fine-tuning.

Table 3.8: telephone speech: Character error rate (CER%) of different TDNN ASR models and character error reduction (CERR%) from Base-NoAug-TDNN. Base-Aug3N-TDNN model is used as the base for all the fine-tuned models (FT). Seed=1.5h is used as the seed amount of data for fine-tuning.

Model	Data size for training / adaptation (seed) (h)	Test dataset (CSJ eval1)							
		Clean		Re-recorded					
				Landline		Mobile 3G		Mobile LTE	
		CER	CERR	CER	CERR	CER	CERR	CER	CERR
Base-NoAug-TDNN	233 (233)	9.4	-	11.0	-	23.6	-	30.4	-
Base-Aug3CN-TDNN	933 (233)	**8.8**	**6.2**	**9.7**	**11.5**	18.5	21.6	25.9	27.8
Base-Aug3N-TDNN	700 (233)	9.6	-1.8	9.9	10.2	**17.1**	**27.8**	27.6	9.4
FT-L-TDNN	7.5 (1.5)	-	-	-	-	-	-	33.6	-10.2
FT-LT-TDNN	9 (1.5)	-	-	-	-	-	-	**22.2**	**27.0**
FT-Aug3L-TDNN	10.5 (1.5)	-	-	-	-	-	-	30.8	-1.1
FT-Aug3LT-TDNN	12 (1.5)	-	-	-	-	-	-	28.1	7.6
FT-Aug5L-TDNN	13.5 (1.5)	-	-	-	-	-	-	30.9	-1.5
FT-Aug5LT-TDNN	15 (1.5)	-	-	-	-	-	-	28.2	7.5

FT-LT-E2E, when fine-tuning Base-Aug3N-TDNN and Base-Aug3N-E2E respectively, transformed features are added with the data combination of FT-L-TDNN and FT-L-E2E.

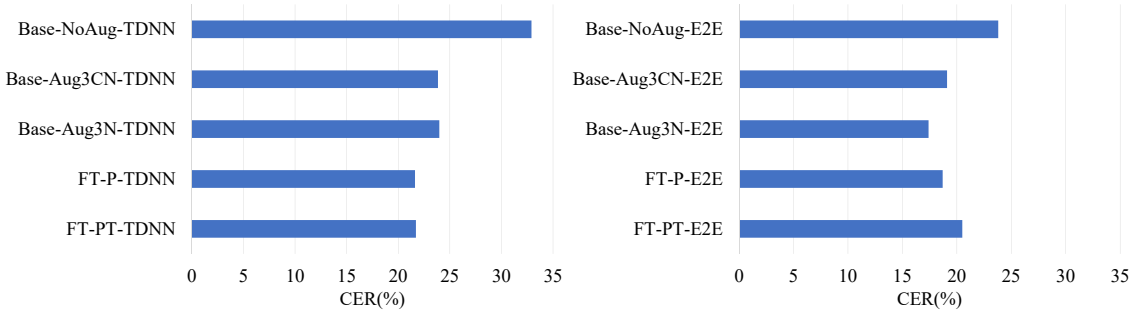
In Table 3.8, we show the performance in detail. The character error rate reduction (27.0%) for the Mobile LTE channel is the best for the proposed fine-tuning method with DNN-based data augmentation, prepared with our proposed method of data augmentation for domain adaptation of DNN-HMM ASR. Table 3.9 shows that end-to-end model-based speech recognition performs better with adaptation proposed that uses a larger set of augmented re-recorded data. Therefore, we get 34.4% CERR for FT-Aug3LT-E2E, which is the best improvement found in this research.

Table 3.9: telephone speech: Character error rate (CER%) of different E2E ASR models and character error reduction (CERR%) from Base-NoAug-E2E. Base-Aug3N-E2E model is used as the base for all the fine-tuned models (FT). Seed=1.5h is used as the seed amount of data for fine-tuning.

Model	Data size for training / adaptation (seed) (h)	Test dataset (CSJ eval1)							
		Clean		Re-recorded					
				Landline		Mobile 3G		Mobile LTE	
		CER	CERR	CER	CERR	CER	CERR	CER	CERR
Base-NoAug-E2E	233 (233)	6.3	-	7.0	-	14.4	-	19.5	-
Base-Aug3CN-E2E	933 (233)	**5.8**	**6.2**	**6.2**	**11.5**	11.7	18.8	16.1	15.1
Base-Aug3N-E2E	700 (233)	6.5	-3.2	6.7	-4.5	**11.5**	**20.1**	17.1	12.3
FT-L-E2E	7.5 (1.5)	-	-	-	-	-	-	13.7	29.7
FT-LT-E2E	9 (1.5)	-	-	-	-	-	-	13.9	28.7
FT-Aug3L-E2E	10.5 (1.5)	-	-	-	-	-	-	13.0	33.3
FT-Aug3LT-E2E	12 (1.5)	-	-	-	-	-	-	12.8	34.4
FT-Aug5L-E2E	13.5 (1.5)	-	-	-	-	-	-	12.5	35.9
FT-Aug5LT-E2E	15 (1.5)	-	-	-	-	-	-	**12.4**	**36.4**

Though this research focuses on the purpose of improving recognition performance of real-environment data, we find improvement on the recognition of clean data also when the amount of clean data is used the most. For Base-Aug3CN-ASRs, the CERR is 6.2% for both DNN-HMM ASR and E2E ASR. Though the relative improvement for Mobile 3G is better for DNN-HMM ASR, it is the same for both of the methods in case of Landline. E2E-based methods start at lower CER to begin with. Though we did not perform training adaptation for Landline and Mobile 3G, we get improvement by considering different real-world conditions, such as, distortions, noises while preparing a better baseline. The improvement strategy reflects on the CER for those channels.

In case of wireless pin-mic recordings in classroom environment, we observe interesting behaviour when performing fine-tuning using the proposed feature transformation-based method of data augmentation. The method is first developed to improve telephone channels and is then applied to the classroom recordings to note its generalization ability. The feature transformation model is trained using less data than the feature transformation model used for the telephone channel. Also, need to keep in mind that the nature of test dataset is completely different than that of Eval1 dataset. As expected, it does not work up to the expectation. We can see the convergence and divergence in Figure 3.20. However, we find impressive improvement



(a) TDNN: Domain adaptation for pin-mic

(b) E2E: Domain adaptation for pin-mic

Figure 3.20: Performance of Data augmentation on wireless pin-mic speech in classroom environment. Base-Aug3N-ASR models are used as the base for all the fine-tuned models (FT). Seed≈1.2h is used as the seed amount of data for fine-tuning.

with CERR of 29.7% for when we use re-recorded data with the simulated data for training adaptation in case of DNN-HMM ASR in Table 3.10. For E2E-based approach, though we fail to achieve the expected result from the proposed method, the best performance is achieved for the Base-Aug3N-E2E, where the data does not contain any clean data.

The results indicate that the adaptation dataset contents fall into the mismatched domain along with data size issue. Also, our investigation shows the variability in recording quality of data in terms of volume and so on than the telephone speech. Therefore, consistent feature transformation could not been achieved. Though the CER is the largest to begin with, for clean test dataset, the best CER is achieved consistently with all the other experiments for Base-Aug3CN-TDNN and Base-Aug3CN-E2E in Table 3.10 and Table 3.11, respectively. Because of the restriction in usability of data, we do not perform validation experiments for classroom wireless pin-mic tasks.

In addition to the explanation above, we would like to state that though directly incomparable due to the differences in model configuration, we get better result for eval1 clean (CER 8.4% for Task1 in [52]) by changing CTC weight α from 0.1 to 0.3 than the state of the art method. Though we use smaller training data (233 h comparing to 581 h), SpecAugment data augmentation technique helps to give it a jump start. Moreover, the data augmentation method adopted for baseline 2,

Table 3.10: Wireless pin-mic speech: Character error rate (CER%) of different TDNN ASR models and character error reduction (CERR%) from Base-NoAug-TDNN. Base-Aug3N-TDNN model is used as the base for all the fine-tuned models (FT). Seed≈1.2h is used as the seed amount of data for fine-tuning.

Model	Data Size for training/ adaptation (seed) (h)	Test dataset (CSJ eval3)			
		Clean		Re-recorded Wireless pin-mic	
		CER	CERR	CER	CERR
Base-NoAug-TDNN	233 (233)	10.6	-	30.8	-
Base-Aug3CN-TDNN	933 (233)	**10.1**	**6.2**	22.1	28.2
Base-Aug3N-TDNN	700 (233)	11.2	-9.4	22.5	26.7
FT-P-TDNN	6 (1.2)	-	-	**21.6**	**29.7**
FT-PT-TDNN	≈7 (1.2)	-	-	21.7	29.4

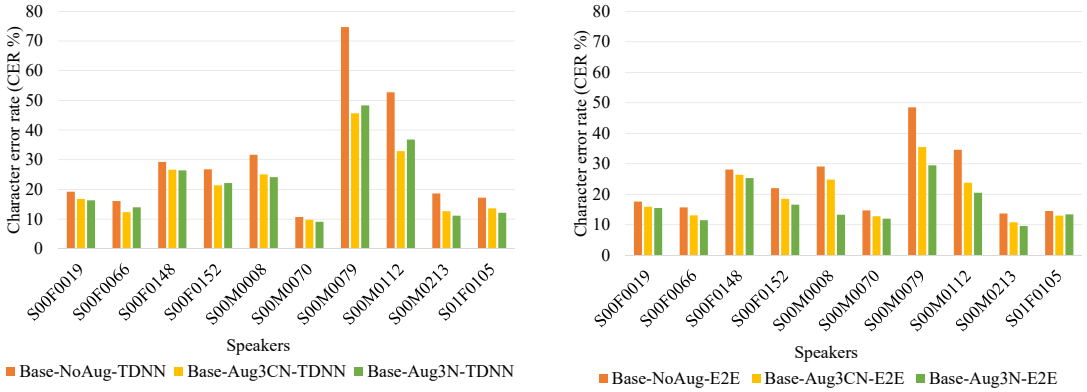
Table 3.11: Wireless pin-mic speech: Character error rate (CER%) of different E2E ASR models and character error reduction (CERR%) from Base-NoAug-E2E. Base-Aug3N-E2E model is used as the base for all the fine-tuned models (FT). Seed≈1.2h is used as the seed amount of data for fine-tuning.

Model	Data Size for training/ adaptation (seed) (h)	Test dataset (CSJ eval3)			
		Clean		Re-recorded Wireless pin-mic	
		CER	CERR	CER	CERR
Base-NoAug-E2E	233 (233)	10.8	-	23.8	-
Base-Aug3CN-E2E	933 (233)	**10.0**	**7.4**	19.1	19.7
Base-Aug3N-E2E	700 (233)	11.4	-5.3	**17.4**	**26.9**
FT-P-E2E	6 (1.2)	-	-	18.7	21.4
FT-PT-E2E	≈7 (1.2)	-	-	20.5	13.9

provides variations in different aspects for the same data and helps to produce 5.8% CER for eval1 clean.

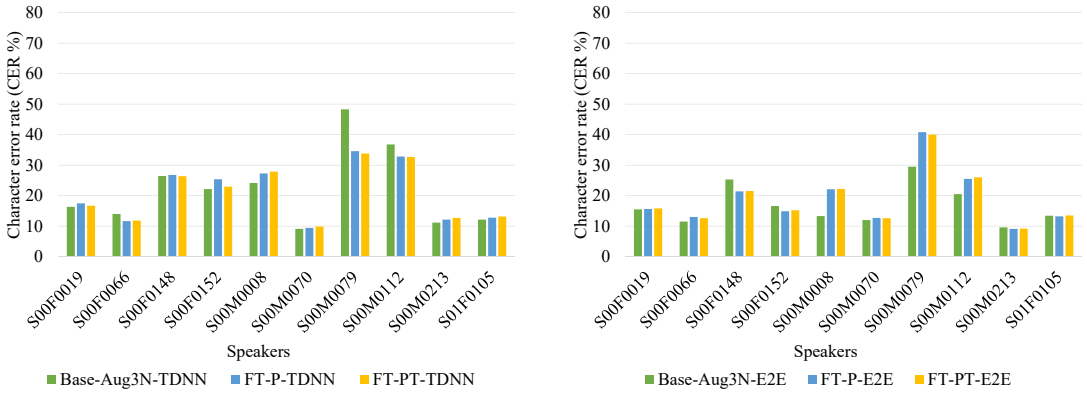
3.6.2 Effect of Variability in Recording Quality

Figure 3.21 shows performance of baseline DNN-HMM hybrid (TDNN) and E2E ASR models for recordings of each speaker using wireless pin mic channel. The performance differences between TDNN-based baselines and E2E-based is noticeable at a glance. However, we find certain speakers in last four recordings perform much worse than the rest of the six recordings. It shows the difference of performance in recordings that are recorded in sessions with different sound level adjustments. Figure 3.22 shows that the fine-tuned Base-Aug3N-E2E models yield worse CER (%) for all the speakers. It is because of the mixture of rather good quality recordings and bad quality recordings. The mismatched condition causes confusion in case of training of



(a) (b)

Figure 3.21: Speaker-wise performance analysis of ASR models for wireless pin mic classroom dataset. (**a**) TDNN: CER(%) of baseline models for each speaker, (**b**) E2E: CER(%) of baseline models for each speaker.



(a) (b)

Figure 3.22: Speaker-wise performance analysis of fine-tuned ASR models for wireless pin mic classroom dataset. (**a**) TDNN: CER(%) comparison of baseline Base-Aug3N-TDNN and fine-tuned models for each speaker, (**b**) E2E: CER(%) comparison of baseline Base-Aug3N-E2E and fine-tuned models for each speaker.

feature transformer as well as in the time of fine-tuning. Due to the small number of speakers available a large variation of speakers could not have been provided. Therefore, there is a possibility of occurring speaker adaptation when about the half of the speakers sit in opposite spectrum. This can cause poor performance for the data which have less matching condition data. Especially, in case of test data ID S00M0079, since this recording has the worst audibility, the most of the data used are different in the sound quality than the data itself, causing worsened adaptation for this data.

As we know that the recording quality of pin mic is different from other recording

channels in "Classroom" environment, we perform further analysis. We perform spectral analysis of pin-mic re-recordings opposed to their original counterparts. It shows that the spectrum of the recordings are divided in two groups according to the sessions they were recorded in. It is showed in details in Chapter 4. In conclusion, some re-recordings, for example, the re-recordings of Speaker A has an amplitude of reasonable audibility. On the other hand, the re-recording set for Speaker B has an amplitude which is hardly audible as opposed to its clean counter part. In the data set of 10 speakers, re-recordings of six speakers have rather good audibility but are noisy. Rest of the four speakers suffer from audibility problem. Therefore, when we adapt for Speaker A, Speaker B and other poorly audible recordings affect the overall performance and it continues for 10-fold cross validation. We suspect that since recording time and settings are different between the group of Speaker A and the group of Speaker B, the variability occurred. The details of this problem and extended research to solve this problem is described in the following Chapter 4.

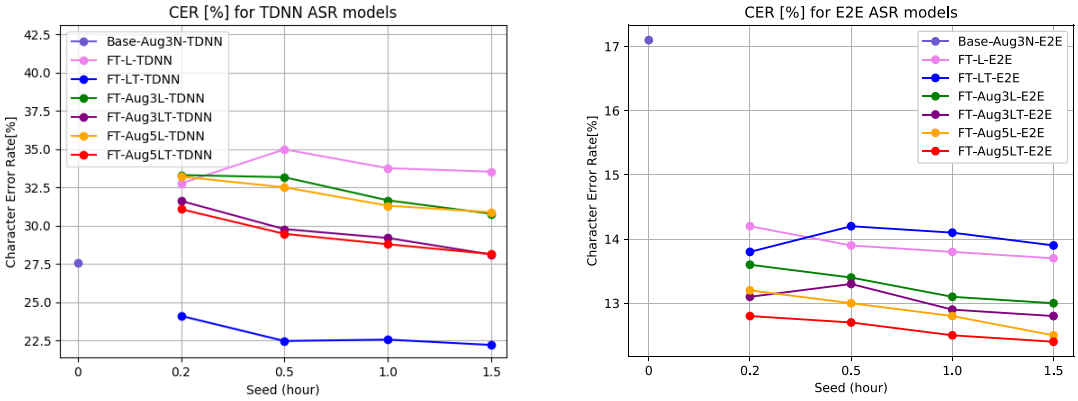
3.6.3 Validation Experiments for Mobile LTE channel with Limited Re-recorded Data

We perform additional experiments to find out the minimum optimal amount of data that need to be prepared for training adaptation, as well as to validate the proposed method, proving its consistency. We acquire the seed amounts of clean data of 0.2, 0.5 and 1.0, and the most is 1.5 h (same condition as Tables 3.8 and 3.9). We perform the detailed experiment only on telephone speech for the DNN-HMM model. The proposed datasets of FT-L-ASR, FT-LT-ASR, FT-Aug3L-ASR, FT-Aug3LT-ASR, FT-Aug5L-ASR and FT-Aug5LT-ASR are compared in Figure 3.23. Experiments are performed to observe the effect of re-recorded data only on the fine-tuning by increasing the amount of LTE channel re-recorded data in "FT-Aug3L-TDNN" and "FT-Aug3LT-TDNN" by adding speed (0.9, 1 and 1.1) and volume (factor: 0.7–1.5) perturbation to the LTE data. We increase the amount of LTE data even more in "FT-Aug5L-TDNN" and "FT-Aug5LT-TDNN" by adding more speed

perturbation (0.8, 0.9, 1, 1.1 and 1.2). Additionally, for each of the combinations, the amount of data used to train the feature transformer model also matches with the size of the seed.

In Figure 3.23, the effect of re-recorded data itself for fine-tuning is proved. Though adding more re-recorded data helps reduce the distance between fine-tuned TDNN with augmented LTE re-recordings and fine-tuned TDNN with augmented LTE re-recordings along with transformed features (pink–blue, green–purple and orange–red curve pairs in Figure 3.23a), it does not necessarily improve the whole performance, rather, it represses the models from converging to the smallest character error rate possible. We do not have more data to observe if they are going to decrease drastically or gradually. However, we clearly see the effectiveness of fine-tuning with transformed features in each case. Moreover, in Figure 3.23b, the models show interesting behavior while the dataset is the smallest and the largest for every model in our task. We notice gradual decrement of the FT-LT-E2E after the point 0.5. We increase the amount of re-recorded data by applying speed and volume perturbation on it, too. In this way, we can observe the effectiveness of the proposed feature transformation method (FT-Aug3LT-E2E and FT-Aug5LT-E2E) with the support of larger simulated re-recorded data. Moreover, with data augmentation for re-recorded speech, the model converges faster even with a smaller seed. The character error rate reduced to 26.1% for the DNN-HMM-based approach (FT-LT-TDNN) and to 34.9% for the end-to-end-based approach (FT-Aug5LT-E2E) by using a seed of 30 min only.

We have also performed experiment on FT-Aug5L-E2E and FT-Aug5LT-E2E by freezing the decoder since it is recommended to only fine-tune the ASR partially when the amount of target domain data is low [23]. However, the experiment gives 17.4% and 17.6% CER for FT-Aug5L-E2E and FT-Aug5LT-E2E respectively with only encoder parameters updated. Since both of the results are way worse than the proposed method, we do not perform further investigations in this directions for end-to-end-based ASR.



(a) (b)

Figure 3.23: Performance of data augmentation for LTE telephone speech by reducing data size. (**a**) TDNN: domain adaptation for mobile LTE, (**b**) E2E: domain adaptation for mobile LTE.

3.7 Summary

By adopting the approach of data augmentation, we can achieve the best result. Also, with data augmentation for re-recorded speech, the model converges faster even with a smaller seed. It gives character error rate reduction of 26.1% for DNN-HMM acoustic model and 31.8% for end-to-end-based ASR model by using a seed of 30 minutes only for training the feature transformation model and fine-tuning the ASRs. However, it also shows some limitation in case of data with variability, such as pin-mic recordings, where the recording condition vary over sessions. In Chapter 4, this shortcoming is handled by taking session dependent feature transformation approach into consideration as well as another approach of end-to-end ASR model.

CHAPTER 4

DOMAIN ADAPTATION OF SELF-SUPERVISED LEARNING MODEL-BASED ASR FOR LIMITED TARGET DOMAIN DATA

This chapter describes domain adaptation approach for self-supervised learning model-based ASR for limited target data. In Section 4.1, the introduction of limited data and pre-trained self-supervised learning-based audio encoder is provided. In Section 4.2,the task settings of domain adaptation method using conventional end-to-end model and self-supervised learning model-based end-to-end are described. In Section 4.3, the experimental setup and in Section 4.4, the results are discussed. Finally, Section 4.5 provides a summary of this chapter.

4.1 Introduction

In this research, data augmentation and domain adaptation both are introduced for conventional end-to-end (**conv E2E**) ASR approach [42,47,52,53] and self-supervised learning end-to-end model-based (**SSL-based E2E**) ASR approach [59, 63] and investigated to find a better solution for addressing acoustic aspect related issue for a specific classroom domain of data which contains acoustic variability. In this chapter, we discuss low-quality wireless pin mic (lapel mic) recording of eval3 test dataset of corpus of spontaneous Japanese (CSJ) [64] in a classroom environment. 10 monologue lectures of eval3 test dataset are concatenated to create a long recording of about 1 h

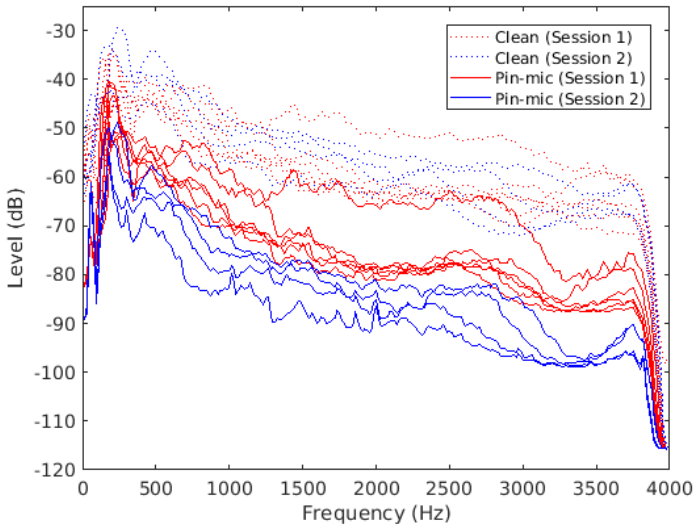


Figure 4.1: Spectral analysis of wireless original clean data and re-recordings through pin-mic channels after down sampling.

40 min for convenience of recording. However, the long recording is recorded in two sessions in two different workdays causing variability in recording condition between the sessions. As a result, we acquire 6 re-recorded monologue speech recordings for the first session, Session1 hereinafter and 4 re-recorded monologue speech recordings for the second session, Session2 hereinafter. The acoustic quality among the recordings varies greatly, especially, from session to session.

In Figure 4.1, we show the spectral analysis of pin mic re-recordings opposed to their original counterparts. In the figure, we show the spectrum of the recordings divided in two groups according to the sessions they were recorded in. The first session is depicted by red lines and the second session is depicted by blue lines. This analysis shows us the problem of significant difference in level (dB) of re-recorded speech between two recording sessions. Some re-recordings, for example, the re-recordings of Speaker A has an amplitude of reasonable audibility. On the other hand, the re-recording set for Speaker B has an amplitude which is hardly audible as opposed to its clean counter part. In the data set of 10 speakers, re-recordings of six speakers have rather good audibility but are noisy. Rest of the four speakers suffer from audibility problem. Therefore, when we adapt for Speaker A, Speaker B and other poorly audible recordings affect the overall performance and it continues

for 10-fold cross validation. We suspect that since recording time and settings are different between Speaker A and Speaker B, the variability occurred. Therefore, the research is conducted addressing this problem by applying data augmentation with session-dependent fine-tuning of ASR models.

In Chapter 3, the effectiveness of using small fine-tuning data has been discussed for two state-of-the-art supervised ASR models. The models are trained in fully supervised manner. Between the two models, the end-to-end (E2E) ASR model has better performance even for small amount of fine-tuning data with proposed augmentation since it can encode acoustic domain information than DNN acoustic model of DNN-HMM ASR model. However, because of its fully supervised nature, at the training phase, the model trains the parameters for the training data with target label (transcription). Therefore, another state-of-the-art, the SSL model based E2E is introduced. Since the encoder of this model is pre-trained with large amount of unlabeled data of different languages from different sources, it is capable of encoding and adapting effectively to the target domain data by fine-tuning with small amount of data without any augmentation.

Self-supervised learning-based models act as audio encoder for the down-stream task (e.g. ASR). The audio encoder learns the feature representation on a large amount of data in unsupervised fashion. Figure 4.2 and Table 4.1 show the audio encoding technique of XLSR model (wav2vec2.0 [59]) and its effectiveness on multi-lingual domain using large amount of data [63]. In this case, the model is pre-trained using 56k hours of data for 53 languages. Because this pre-trained model contains knowledge from multiple domain aspects, such as, language, speaking style, speakers, and so on, the acoustic domain aspects for each recording condition also vary. Since it shows its effectiveness on language domain, it is also expected to be able to encode acoustic aspect of the speech. Therefore, the XLSR model is used as the pre-trained audio encoder for domain adaptation in this research.

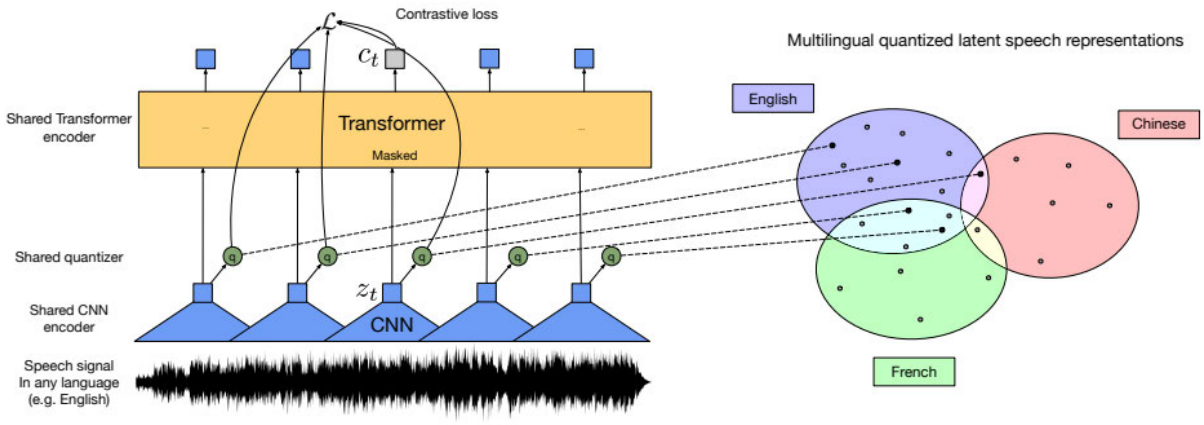


Figure 4.2: Self-supervised learning (SSL)-based audio encoding for with pre-trained XLSR model [63].

Table 4.1: Example performance (phoneme error rate %) of XLSR model for cross-ligual domain speech recognition dataset CommonVoice as stated in [63].

Model	**D**	**#pt**	**#ft**	**es**	**fr**	**it**	**ky**	**nl**	**ru**	**sv**	**tr**	**tt**	**zh**	**Avg**
Number of pretraining hours per language				168h	353h	90h	17h	29h	55h	3h	11h	17h	50h	793h
Number of fine-tuning hours per language				1h	1h	1h	1h	1h	1h	1h	1h	1h	1h	10h
Baselines from previous work														
m-CPC† (Rivière et al., 2020)	LS_{100h}	10	1	38.7	49.3	42.1	40.7	44.4	45.2	48.8	49.7	44.0	55.5	45.8
m-CPC† (Rivière et al., 2020)	LS_{360h}	10	1	38.0	47.1	40.5	41.2	42.5	43.7	47.5	47.3	42.0	55.0	44.5
Fer et al.† (Fer et al., 2017)	BBL_{all}	10	1	36.6	48.3	39.0	38.7	47.9	45.2	52.6	43.4	42.5	54.3	44.9
Our monolingual models														
XLSR-English	CV_{en}	1	1	13.7	20.0	19.1	13.2	19.4	18.6	21.1	15.5	11.5	27.1	17.9
XLSR-Monolingual	CV_{mo}	1	1	6.8	10.4	10.9	29.6	37.4	11.6	63.6	44.0	21.4	31.4	26.7
Our multilingual models														
XLSR-10 (unbalanced)	CV_{all}	10	1	9.7	13.6	15.2	11.1	18.1	13.7	21.4	14.2	9.7	25.8	15.3
XLSR-10	CV_{all}	10	1	9.4	14.2	14.1	8.4	16.1	11.0	20.7	11.2	7.6	24.0	13.6
XLSR-10 (separate vocab)	CV_{all}	10	10	10.0	13.8	14.0	8.8	16.5	11.6	21.4	12.0	8.7	24.5	14.1
XLSR-10 (shared vocab)	CV_{all}	10	10	9.4	13.4	13.8	8.6	16.3	11.2	21.0	11.7	8.3	24.5	13.8
Our multilingual models (Large)														
XLSR-10	CV_{all}	10	1	7.9	12.6	11.7	7.0	14.0	9.3	20.6	9.7	7.2	22.8	12.3
XLSR-10 (separate vocab)	CV_{all}	10	10	8.1	12.1	11.9	7.1	13.9	9.8	21.0	10.4	7.6	22.3	12.4
XLSR-10 (shared vocab)	CV_{all}	10	10	7.7	12.2	11.6	7.0	13.8	9.3	20.8	10.1	7.3	22.3	12.2
Our Large XLSR-53 model pretrained on 56k hours														
XLSR-53	D_{53}	53	1	**2.9**	**5.0**	**5.7**	**6.1**	**5.8**	**8.1**	**12.2**	**7.1**	**5.1**	**18.3**	**7.6**

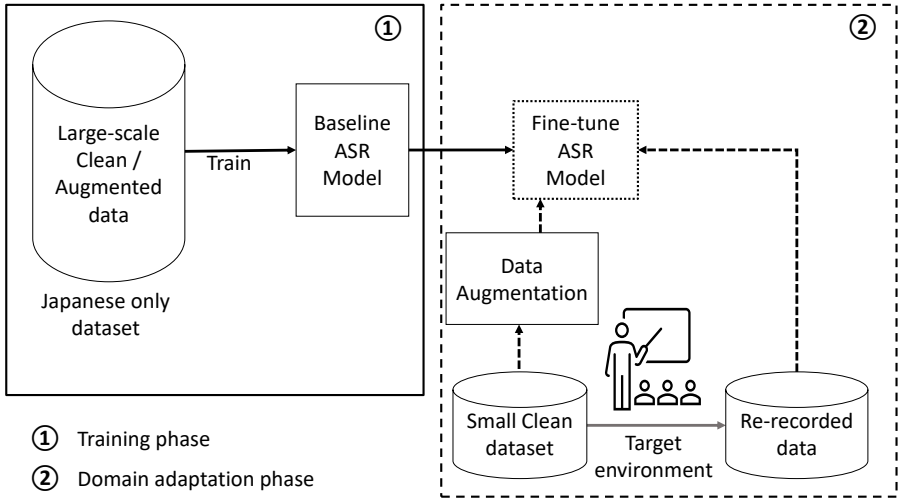


Figure 4.3: Task setting for domain adaptation using CTC/Transformer end-to-end (conv E2E) ASR model for classroom re-recorded speech.

4.2 Domain Adaptation Methods

4.2.1 Domain Adaptation of SSL-based ASR Model by Fine-tuning for Limited Target Domain Data

The neural networks are ideally trained using a large-scale clean database where they perform reasonably well for the clean data of matched condition. However, they do not perform up to expectation when applied for real environment speech. Therefore, we adopt some domain adaptation approaches such as increasing variability in training data by applying general data augmentation approaches or fine-tuning of the ASR model by using generally augmented data along with specially augmented data by DNN-based feature mapping. We also attempt to fine-tune the SSL-based ASR model by using the re-recorded data only and observe the effect for different amounts of data. The SSL-based model XLSR is pre-trained using large amount of variable domain data. The hypothesis is that it contains the knowledge of different domains already, a small amount of target domain data only should help it converge at the time of fine-tuning. Figures 4.3 and 4.4 depict the schematic diagram of task setting for both E2E and SSL-based domain adaptation approaches.

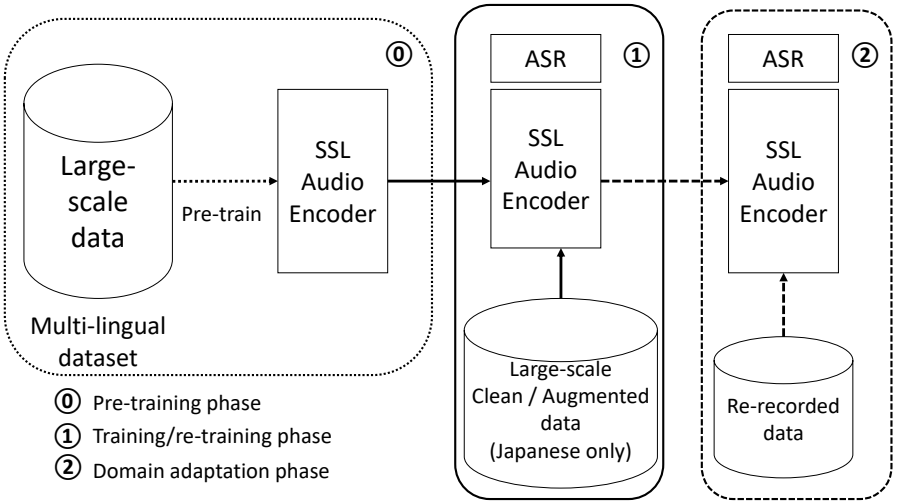


Figure 4.4: Task setting for domain adaptation using self-supervised learning model-based end-to-end (SSL-based E2E) ASR model for classroom re-recorded speech.

4.2.2 Data Augmentation Approaches as Baselines

We take various data augmentation approaches into account of different stages of training the ASR. The most important part is to create an elevated platform for the fine-tuning to action. Therefore, we train a baseline model with simulation-based augmented data apart from the baseline where only clean data are used to train the model.

The baseline model trained using dataset without augmentation is called Base-NoAug-ASR. It consists of clean data from the subset of CSJ corpus with μ-law encoding. The baseline model trained using dataset with augmentation is called Base-Aug3N-ASR. It spins off from the dataset of Base-NoAug-ASR model. The dataset of Base-Aug3N-ASR consists of μ-law encoded clean data being speed perturbed to 0.9, 1.0 and 1.1 times of the original speed. After speed perturbation, volume perturbation with the range 0.7–1.5 is applied to the data. Then noises are added to the perturbed data with a random signal to noise ratio (SNR) of 5, 10, 15 and 20. Finally, G.712 filter is applied on the three times larger dataset than the original data. μ-law encoding and G.712 filter is exclusive to telephony data. However, we use them for conducting research on wireless pin mic data for classroom condition, since this research is an extension of our previous research [25] that also involve telephone

Table 4.2: Dataset for training baselines. Notations: S1 = Speed perturbation (0.9, 1, 1.1), Volume perturbation (0.7 - 1.5). The notation of "ASR" is replaced in the following section depenting on the type of model used (ASR=E2E or SSL). Size denotes the amount (×233h) of data.

Model	Clean				Noisy				Total size (×233 h)
	μ-law encoding	Speed pert.	Vol pert.	Size	μ-law encoding	Speed pert.	Vol pert.	Size	
Base-NoAug-ASR	✓	-	-	1	-	-	-	0	1
Base-Aug3N-ASR	-	-	-	0	✓	✓	✓	3	3

speech. Therefore, in this dissertation, we share the baselines that include telephony characteristics as a part of generalization. ASR in the names of the baselines denotes the type of the training approach, conventional E2E or SSL-based E2E in this case. Table 4.2 shows details of dataset for training baseline models for both E2E models.

Speed perturbation is used not only to increase the amount of data, but it also gives us two different pitches (fundamental frequencies) for each speaker, which somewhat simulates the effect of increasing the number of speakers with the same content of speech. In this way, we obtain three times larger diverse dataset with the same speed adjusted transcription. Volume perturbation helps us to simulate different vocal levels or quality of devices. By applying noise, we make the baseline robust to common noises, such as computer or air duct noises experienced mostly in indoor and a very few outdoor scenarios like crowd or exhibition booth.

4.2.3 Data Augmentation Based Fine-tuning of Conv E2E ASR Model

Data augmentation is also performed while performing fine-tuning of conventional E2E model. For fine-tuning, we adopt session dependent and general approach. For both of the approaches, following augmentation methods are applied. Table 4.3 shows the details of fine-tuning datasets for conv. E2E models. The eval3 dataset of CSJ is the base for the re-recorded dataset called pin mic. Therefore, we apply three-way speed and volume perturbation on clean CSJ eval3 data. Fine-tuning is exclusively intended for the target domain of classroom data. Hence, we do not apply μ-law encoding or G.712 filtering while preparing the fine-tune dataset. We add noise to un-altered eval3 data to get noisy data of the same size as the eval3 dataset. This noisy dataset is used along with speed and volume perturbed data

Table 4.3: Dataset for fine-tuning of Base-Aug3N-ASR. Notations: FT= Fine-tuning of Base-Aug3N-E2E model, P = Adapted by pin-mic re-recordings, S1 = Speed perturbation (0.9, 1, 1.1), V = Volume perturbation (0.7 - 1.5), T = Transformed features. The notation of "E2E" is for the conventional E2E models. Size denotes the number of seed sized dataset used for fine-tuning. seed(P) = ≈ 1.2 h for general, seed(P) = ≈ 37 min for Session1 and seed(P) = ≈ 25 min for Session2.

Model	Clean (S1, V)	Noisy	Re-recorded			Trans.		Total size ×seed
	Size	Size	SP.	Vol.	Size	T	Size	
FT-Aug3CNP-E2E	3	1	-	-	1	-	0	5
FT-Aug3CNPT-E2E	3	1	-	-	1	✓	1	6
FT-Aug3CNPS*-E2E	3	1	-	-	1	-	0	5
FT-Aug3CNPTS*-E2E	3	1	-	-	1	✓	1	6

as well as the pin mic data itself. The whole dataset used to fine-tune E2E model is FT-Aug3CNP-E2E. We have another dataset that includes transformed features created by a DNN-based regression model trained to map pin mic features from clean features [25]. This extended dataset is used to fine-tune FT-Aug3CNPT-E2E. Here, FT stands for fine-tuning, Aug3C means augmented clean data with speed and volume perturbation, N denotes noisy data, P is for re-recorded pin mic data, and for another set, T stands for transformed features.

For session-dependent training, we create datasets separating the data to two datasets naming them with inclusion of initials for their belonging session, FT-Aug3CNPS*-E2E and FT-Aug3CNPTS*-E2E. Here S* denotes session number.

4.2.4 ASR Fine-tuning of Self-supervised Learning-based E2E ASR Model

For self-supervised learning model-based fine-tuning approach of end-to-end ASR model (SSL-based E2E ASR model), we first use XLSR pre-trained model and fine-tune it using the dataset of Base-NoAug-ASR and Base-Aug3N-ASR separately using the CSJ transcription to create Base-NoAug-SSL and Base-Aug3N-SSL. Both of them are then fine-tuned using full set of pin mic for and another attempt is to reduce the data size (3 min of each speaker, 15 min for Session1 and 9 min for Session2) of pin mic in a session-dependent manner with six-fold and four-fold cross validation for each session respectively. Therefore, we use full set of pin mic

for fine-tuning FT-PS1-SSL for Session1 and FT-PS2-SSL for Session2. Reduced datasets are used to train FT-PS1-15min-SSL and FT-PS2-9min-SSL.

4.3 Experimental Setup

4.3.1 Datasets

Baselines

The most basic baseline model Base-NoAug-ASRs is trained using subset of training dataset of corpus of spontaneous Japanese (CSJ) [64]. This dataset contains 948 monologue speech recordings of 233h, consisting of 141 female and 807 male speakers. The dataset is down sampled to 8kHz to achieve generalization along with telephone speech for CTC/Transformer E2E ASR. To fine-tune the SSL-based E2E ASR, data with original 16kHz sampling rate is used. The augmented baseline dataset for Base-Aug3N-ASRs contains speed and volume perturbed noisy filtered data. Hence, creating three times larger data from the dataset of Base-NoAug-ASR. The additive noise applied on the data are noises chosen from a subset of the noise database "JEIDA-NOISE" [66]. The noise types used are exhibition booth, crowd, computer room (medium), computer room (workstations), air conditioner (large), exhaust fan and air duct. The noises are selected and added randomly with a random SNR over the range of 5 to 20dB with 5dB interval. The dataset is distorted using G.712 filter to make it adaptable for telephone speech.

Fine-tuning CTC/Transformer E2E ASR models

We use 10 clean Eval3 recordings of CSJ test dataset and re-record them in the classroom scenario using a general low-quality analog wireless pin mic with an 800 MHz band radio transmitter. Since we only acquire 10 recordings of 10 speakers, a total of 1.32 h of pin mic data are available in this research. Since quality wise, the data are divided in two sessions, we get 45 min of data of 6 recordings for Session1 and 34.28 min of 4 recordings for Session2. To keep the open condition, we perform

six-fold cross validation for Session1 and four-fold cross-validation for Session2. For each session, recording of 1 speaker is kept for testing and the rest are used for fine-tuning in rotation.

Fine-tuning SSL-based E2E ASR models

The pre-trained SSL-based model of Wav2vec2.0 (XLSR [63]) is first re-trained with the acoustics and the character dictionary of CSJ dataset (Base-NoAug-SSL and Base-Aug3N-SSL). Then they are trained using the full available dataset of pin mic for each session with cross validation (FT-PS1-SSL and FT-PS2-SSL). Another variation of fine-tuning involves reducing the data size by taking only 3min from each recording. FT-PS1-15min-SSL contains 5 speakers from about 37 min and FT-PS2-9min-SSL has of 3 speakers from a total of about 25 min in rotation.

Evaluation tasks

Evaluations are performed on 10 recordings of eval3 (5 male, 5 female speakers), which is clean dataset, called "Clean" along with re-recorded data in classroom environment using wireless pin mic, called "pin mic".

4.3.2 CTC/Transformer End-to-end ASR Model

E2E ASR model described in this dissertation is a hybrid CTC/Transformer end-to-end model [84]. This model is trained using baseline datasets Base-NoAug-ASR and Base-Aug3N-ASR described previously in Section 4.3.1. Table 4.4 shows the configuration of CTC/Transformer E2E ASR Model. 43 dimensions of F-bank and pitch features are used to train the ASR model. Global CMVN is applied to the input along with SpecAugment [82] data augmentation method. For performing SpecAugment, time warp (max 5), frequency masks (no. of masks: 2) and time masks (no. of masks: 2) parameters are used on the training data. The encoder consists of 12 layers, each consisting of 2048 units. The decoder consists of 6 layers with 2048 units in each of them. A subsampling unit consisting of 2 convolution layers is used in the encoder. It reduces the input length to one-fourth. There are four attention

Table 4.4: The configuration of CTC/Transformer-based end-to-end (conv E2E) model.

Configuration	CTC/Transformer E2E ASR Model
Input features	Filter bank (40), pitch (3)
CMVN	Global
Encoder	**Transformer layers: 12** Units: 2048 Sub-sampling unit: 2 convolutional layers Attention heads: 4; dimension: 256
Decoder	**Transformer layers: 6** Units: 2048
CTC loss weight	0.3
Output units	2865

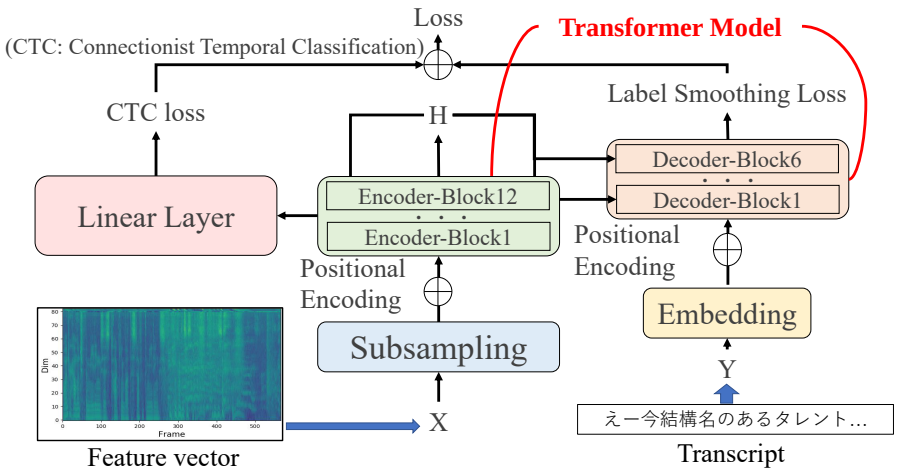


Figure 4.5: The architecture of CTC/Transformer-based end-to-end (conv E2E) model [84].

heads with 256 dimensions. The weight of α for CTC loss is set to 0.3. In conv E2E model, the label for each speech segment is trained from the beginning. The number of output units is 2865, corresponding to the number of characters, including Japanese characters. The experiments regarding E2E are performed using ESPnet, the E2E speech recognition toolkit [83]. Figure 4.5 depicts the block diagram of network architecture of CTC/Transformer E2E model.

4.3.3 SSL-based End-to-end ASR Model

As shown in Table 4.5, the pretrained model used is a Wav2vec2.0 model trained using XLSR [63] dataset, which consists of 56kh speech data of 53 languages. It is called SSL-based E2E model. This dataset originally contains 2h of Japanese speech data. The feature encoder consists of 7 convolutional layers. The transformer

Table 4.5: The configuration of SSL-based (XLSR) end-to-end ASR model.

Configuration	Wav2vec2.0 (XLSR) [Conneau+ 21]
Pre-trining data	53 languages, 56k hours ※Japanese: 2 hours
Input	Raw speech
Convolutional layer	7 layers
Encoder	**Transformer layers: 24** Units: 4096 hidden units Attention heads: 16; dimension: 1024
Decoder Objective	**Linear layer: 1** CTC

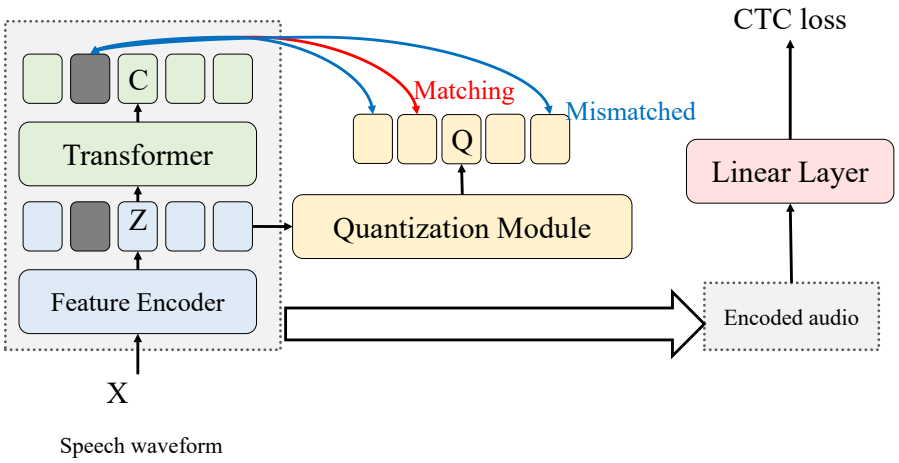


Figure 4.6: The architecture of self-supervised learning model-based end-to-end (SSL-based E2E) ASR model [63].

encoder consists of 24 layers. Learning rate is $5 \times 10^{(-5)}$. When training the SSL-based audio encoder, it generates pseudo labels, which are then used in training the ASR. Therefore, it trains itself with a method between unsupervised and supervised methods. However, when fine-tuning an SSL-based E2E ASR model, the linear layer of decoder part is trained using the segment labels available for the given speech signal. In this particular case, the ASR decoder is trained to output 2865 characters, corresponding to the number of characters, including Japanese characters. Training is optimized with Adam and a tri-state rate scheduler where the learning rate is warmed up for the first 10% of updates, held constant for next 40% and then linearly decayed. Training is updated for 3000 times. fairseq tool [85] is used to train and fine-tune the SSL models. Figure 4.6 depicts block diagram of network architecture of SSL-based E2E model.

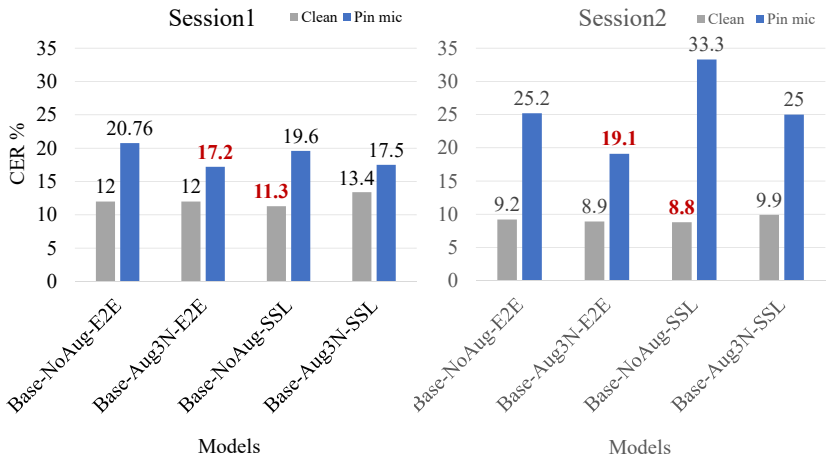


Figure 4.7: Character error rate (CER%) of Baseline models for both end-to-end approaches for session 1 and session 2.

4.4 Results and Discussion

4.4.1 Baselines

All the four baseline variations of ASR for each session are discussed in the light of character error rate (CER%) of Clean and Pin mic data. Figure 4.7 shows that observe consistent performance for Clean and Pin mic domain data for both of the sessions. We understand from the result that the results for in-domain case (Clean) converge for SSL-based baseline whereas E2E-based baseline with data augmentation works better for out-of-domain case (Pin mic).

4.4.2 Fine-tuning CTC/Transformer (conv E2E) ASR Model

In case of fine-tuning the E2E model, we adopt two approaches. One is overall approach, which involves both of the sessions together while fine tuning. Figure 4.8 shows that it improves the performance for Session1 (CER: 16.6%), which is better than the baselines. This is expected result in case of data with proper audibility, which proves that the knowledge is transferred for the target domain with smaller data than baseline. This helps us realize that domain adaptation with fine-tuning of a pre-trained model is effective for any kind of data with credible quality. However, in case of Session2, we see that session dependent fine-tuning helps converge towards

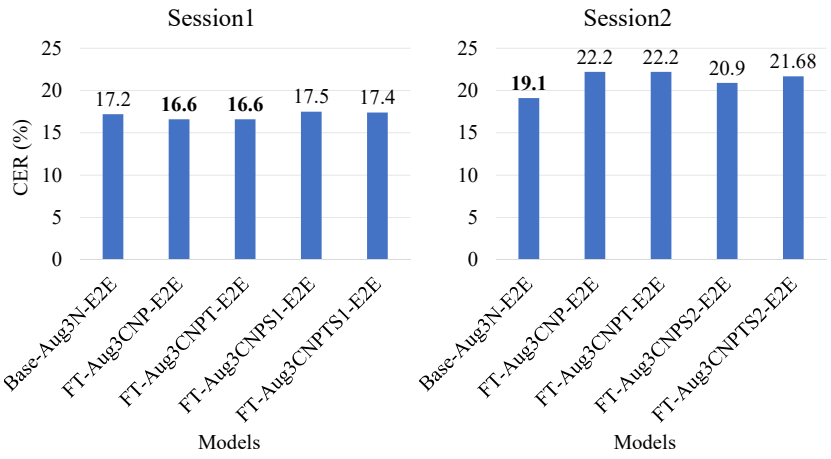


Figure 4.8: Character error rate (CER%) of different fine-tuned CTC/Transformer-based end-to-end (conv E2E) ASR models for session 1 and session 2. The CER(%) of pin mic for Base-Aug3N-E2E for session 1 and session 2 are 17.2% and 19.1%, respectively.

better performance, though it cannot exceed the performance of Base-Aug3N-E2E (CER: 19.1%). Therefore, the possible explanation is that the fine-tune dataset and the general training process is not suitable for this specific domain.

4.4.3 Fine-tuning Self-supervised Learning model-based (SSL-based E2E) ASR Model

Fine-tuning is performed on SSL-based model in two different ways for each session. One way is to use overall data of pin mic recordings only for each speaker in a session (FT-PS1-SSL and FT-PS2-SSL). In Figure 4.9, we observe overall improvement for session 1 and 2 even with only 15 and 9 min of data respectively than the baselines. For Session1, FT-PS1-15min-SSL gives CER of 17.1% (17.2% in Figure 4.7) whereas for Session2, FT-PS2-9min-SSL gives 18.9% (19.1% in Figure 4.7). We observe the best result when we use the full data available for each speaker for the session. FT-PS1-SSL gives 16.5% and FT-PS2-SSL gives 17.9% CER.

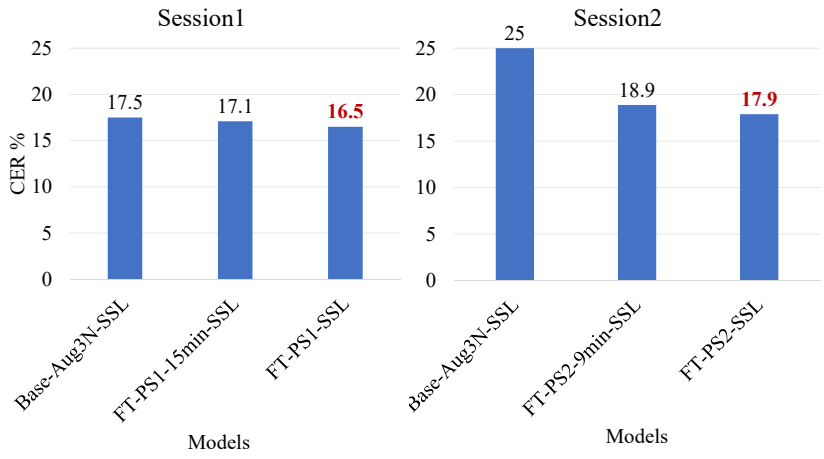


Figure 4.9: Character error rate (CER%) of different fine-tuned self-supervised learning-based end-to-end ASR models (SSL-based ASRs) for session 1 and session 2.

4.4.4 Analysis of the Worst Performing Recording

The performance is the worst for the recording ID S00M0079 in every case of baselines and fine-tuned models. Table 4.6 shows the CER for the comparable cases for the worst recording from Session2. It shows that the CER improves to 26.1% for FT-PS2-SSL consistently along with other recordings, showing the superiority of SSL-based fine-tuning method for out-of-domain with moderately small amount of data of the target domain data only. We also perform central kernel alignment [86] analysis for before and after performing fine-tuning with the target domain data. The CKA value ranges from 0.0 to 1.0 for the previous and current state of each layer of the model. The larger the CKA value, the similar the layers are comparing to before parameters update, which indicates that they do not learn new knowledge. Figure 4.10 shows how the alignment of each layer change after fine-tuning with the target re-recorded data for Session2. This happens due to the mismatch between previous and target domain data whereas the layers do not change much for Session1 data. with the change in alignment, we understand that the self-supervise learning-based audio encoder is unable to preserve the knowledge gained from the previous fine-tuning stage. However, this model is able to perform adjustment to learn the differences. Therefore, the worst performing recording with the ID S00M0079 can achieve improvement. This is a new phenomenon from the acoustic perspective of speech recognition using

Table 4.6: Character error rate (CER%) of classroom wireless pin mic data with worst performance.

Approaches	Models	Pin mic: **S00M0079**
Baselines	Base-Aug3N-E2E	29.5
	Base-Aug3N-SSL	43.4
Fine-tuning of conv E2E ASR	FT-Aug3CNP-E2E	37.5
	FT-Aug3CNPS2-E2E	40.8
Fine-tuning of SSL-based E2E ASR	FT-PS2-SSL	**26.1**
	FT-PS2-9min-SSL	27.2

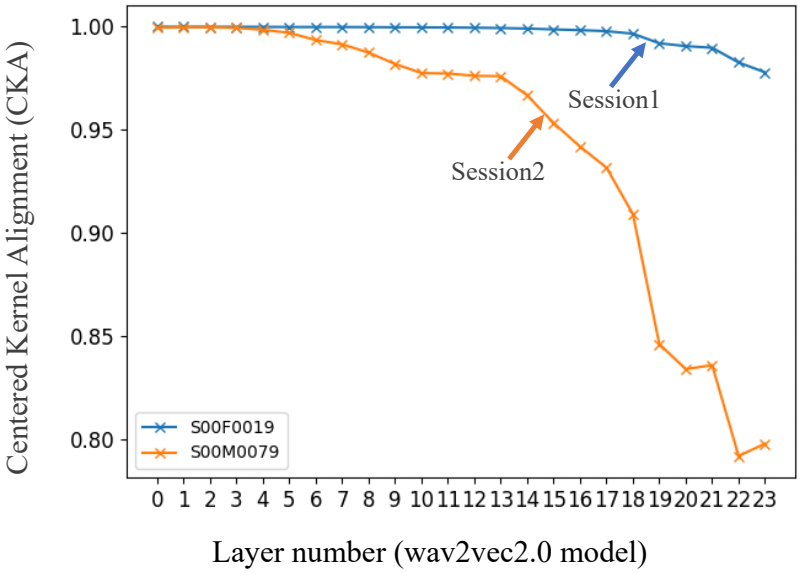


Figure 4.10: Central kernel alignment analysis of before and after fine-tuning audio encoder of Wav2vec2.0 for session 1 and session 2. X-axis: layer number in encoder block, y-axis: CKA value (the larger the similar).

SSL-based ASR models.

4.4.5 Discussion of Both End-to-End Approaches

Two end-to-end models, CTC/Transformer (conv E2E) and SSL model-based (SSL-based E2E) end-to-end models are used in this research. As listed in Table 4.7, both of the models have unique characteristics. Such as, one of them takes filter-bank feature as input and another starts with wave form of speech directly. They also have relative pros and cons, such as, the large training computation cost of SSL-based E2E approach (315 M parameters) against rather modest conv E2E (27 M parameters), being able to run on central processing unit (CPU) or only using graphical processing unit (GPU), etc. On the other hand, the real time factor to decode is much smaller for

Table 4.7: Difference between two end-to-end architecture-based models.

Categories	**CTC/Transformer E2E (conv E2E)**	**SSL Model-based E2E (SSL-based E2E)**
Type	Encoder-decoder	Encoder-decoder
Encoder-decoder	Encoder: 12 Transformer blocks Decoder: 6 Transformer blocks	Encoder: 24 Transformer blocks Decoder: 1 linear layer
Input	F-bank feature	Raw wave
Computation cost	CPU or GPU can be used for training and decoding	Using GPU is compulsory
Training approach	Encoder: Fully supervised Decoder: Fully supervised	Encoder: Self-supervised Decoder: Supervised
Number of parameters	27 M	315 M
Real time factor	0.188	0.0032

SSL-based E2E approach (0.0032) than the conv E2E (0.188). Real time factor (RTF) is the ratio of time taken to process the input and the duration of input itself. Also, the performance also differs for variable data. Though the baseline shows better result (low CER) in case of conv E2E for both sessions (Session1: 17.2%, Session2: 19.1%) than SSL-based E2E (Session1: 17.5%, Session2: 25.0%). However, fine-tuning them shows different behavior. Fine-tuning SSL-based E2E with re-recorded data only yields gain for both sessions (Session1: 16.5%, Session2: 17.9%), whereas fine-tuning conv E2E produces worst performance for Session2. From above discussion, it is difficult to draw clear conclusion on which model architecture is better in one word. Rather, it is understood that both of the models have their merits and demerits, and conv E2E performs well for invariable data similar to pre-training data, whereas SSL-based E2E is recommended to fine-tune for variable data (Session2) different from pre-training data. If considering mobility, conv E2E can be used on mobile devices. However, methods like knowledge distillation can be adopted to downsize and extract the knowledge of large SSL-based model and run it on mobile device.

In future, it is planned to perform feature-based experiments on SSL-based E2E including using augmented data using feature transformation for fine-tuning.

4.5 Summary

In this chapter, conventional end-to-end ASR models and contemporary self-supervised learning-based ASR models are observed for domain adaptation of pin mic recording in classroom environment. With investigation, it is observed that though with the larger data with a general augmented data, the conventional E2E-based models work better for out-of-domain data, the audio encoding capability helps SSL-based models perform even better for target domain when they are adapted by fine-tuning using moderately small amount of target data (Session1 CER: 16.5% and Session2 CER: 17.9%). The reason is that the SSL-based audio encoder contains knowledge from pre-training with large multi-linguistic and acoustic domain dataset. They even converge with only 9min of data for problematic Session2. In future, we plan to observe the effect of speaker adaptive fine-tuning approach on SSL-based ASR models.

CHAPTER 5

CONCLUSIONS AND FUTURE DIRECTIONS

In this chapter, the research presented in this dissertation is concluded and the insights for the future directions are described based on the knowledge and experience gained the research. Section 5.1 states the conclusion and Section 5.2 provides future directions.

5.1 Conclusions

Automatic Speech Recognition (ASR) has been an active field of artificial intelligence (AI) for over six decades. Throughout this time the approaches taken for performing task vary from hand-crafted approach to the very modern artificial neural network (ANN)-based approach. No matter which approach is used, for a system to be able to perform ASR, it needs to be trained using speech samples along with its corresponding label also known as transcription. There are various data corpora that provide speech data for various tasks. Up until now, the most popular task has been to recognise speech acquired in relatively quiet (clean) environment. Also, for a model to be able to perform such a task, it needs to be trained with a fairly large amount data. Therefore, the research on real world speech recognition is still in advancing phase.

The preceding research on the topic of real environment domain involve primitive

approaches of adaptation, such as cepstral mean variance normalization (CMVN), vocal tract length normalization (VTLN), feature space maximum likelihood linear regression (fMLLR), etc. There are other approaches of speech recognition for limited environment data which involve generating real-world like data by applying perturbations, room impulse responses (RIR) to simulate environment acoustics etc. There is even DNN-based data generation approach using feature mapping method like variational autoencoder (VAE) to generate source or target domain data and vice versa. Also, there are research for domain adaptation to target domain in case of target domain data scarcity. However, most of the domain adaptation approaches are applicable to hand-crafted approaches. The research on domain adaptation DNN-HMM ASR models or end-to-end are very small comparing to other topics. These research suggest fine-tuning pre-trained model using target domain data. However, it is evident that ample amount of target domain data is required. Therefore, we focus on domain adaptation of ASR models for limited target domain data.

In this research, to solve the problem of domain adaptation with limited data, the fine-tuning approach of domain adaptation is adopted. In this research, two recording environments are considered as target domain (mobile telephone, classroom) for validating the proposed method. By re-recording clean data in target domain, recording and transcription cost could be reduced significantly. However, the re-recorded data suffer from some unintentional problems such as, temporal misalignment from packet loss and restoration in case of mobile telephone, and variability in recording in case of long-term re-recording condition. Therefore, though re-recording transcribed clean data may seem to be an easy solution, depending on real-world problems, it could be difficult to achieve desired performance just by it.

One of the goals in this research is to augment data for target domain by using clean and re-recorded data pairs to train a DNN-based regression model to map clean features to target domain features and generate features with target domain characteristics (feature transformation). In this thesis, a frame-by-frame approach

of training feed-forward DNN is proposed. Due to the frame-by-frame computation policy, the misaligned telephone speech could not have been used. Therefore, a geometric approach of correcting misalignment and a filtering method to filter out internally misaligned utterances are adopted. However, it results in reduction of usable utterances significantly (34% of the re-recorded utterances could be used). The DNN for feature transformation is trained using a small portion of the filtered and aligned speech features as target with the clean counterpart as input. This DNN is used to perform data augmentation by generating re-recording-like data. The generated features are used with the re-recorded and augmented clean features to perform domain adaptation by fine-tuning ASR models with them. Before fine-tuning with the limited amount of target domain data, in the training or pre-training phase, we use the traditional approaches of augmentations line speed and volume perturbation, adding noise, applying distortions through filtering, etc. along with speaker-level or global CMVN depending on the ASR models used. These fundamental approaches help us achieve higher baseline models which then converge better than not performing any pre-processing. Also, experiment results show that using simulated augmented data closer to the target domain as much as possible for pre-training the models are more effective than just training the models using larger amount of clean data.

The effect of domain adaptation is observed independently to the ASR models such as, two state-of-the-art models, time delay neural network (TDNN)-based DNN-HMM acoustic model and hybrid CTC/Transformer-based E2E ASR model. Both of the model structures encompass mechanism to handle temporal aspect of speech data. By training the ASRs with augmented large clean dataset, it is possible to get closer to the target domain. By fine-tuning them with even small amount of target domain data along with DNN-based augmented features, it has been possible to achieve 27% character error rate reduction (CERR) for telephone speech using LTE network for DNN-HMM hybrid acoustic model and 36.4% for hybrid CTC/Transformer E2E ASR model.

Despite the success of the method mentioned above with telephone speech, the performance for classroom wireless pin mic recording was not satisfactory since the recording levels (dB) are different for different recording sessions. Also, the number of recordings is very small, which makes it difficult to train the DNN-based feature transformation model with mixed condition. Therefore, another state-of the-art, a self-supervised learning (SSL)-based E2E approach is investigated for domain adaptation that incorporates multi-lingual pre-trained self-supervised model and fine-tune it with large Japanese data followed by small amount of session-dependent target domain data. This approach helps us achieve character error rate of 16.5% for the first session and 17.9% for the second session (16.6% and 22.2% respectively for the hybrid CTC/Transformer end-to-end approach mentioned in the paragraph above) for the classroom wireless pin mic recordings. It shows that the SSL E2E model is robust for domain adaptation by fine-tuning.

5.2 Future Directions

Though we achieve improvements for the two target domain real environment data we considered in this research, there is still room for improvements. In this research, we discuss single-domain approaches only. However, in future, it may lead to multi-domain adaptation. Also, the multi-speaker aspects can be investigated to make this research more versatile. The thought regarding future directions are addressed in following sections.

Multi-domain Aspects of Domain Adaptation

To perform multi-domain adaptation in more agile way, the system needs to be aware of the current domain to give recommendations about the data augmentation or fine-tuning approach. Therefore, domain identification may be introduced in the pre-processing stage. Another approach of multi-task training with domain classification as a feedback to the system and optimize it together with the ASR task can be helpful in case of end-to-end ASR models.

Multi-speaker Aspects of Domain Adaptation

In a natural situation, it is only natural to have multi-speaker speech in real environment. When there is un reliable segment information, it is difficult to separate single speaker speech to perform any further processing. Therefore, multi-speaker voice activity detection (VAD) is a promising candidate to perform speech separation and segmentation together as pre-processing. It will be more effective if the VAD task is performed in end-to-end manner by optimizing it together with the target ASR task.

BIBLIOGRAPHY

[1] Mohamed, A.; Dahl, G. E.; Hinton, G. Acoustic Modeling using Deep Belief Networks. IEEE Transactions on Audio, Speech and Language Processing, vol. 20, pp. 12–22, 2012.

[2] Dahl, G. E.; Yu, G.; Deng, L.; Acero, A. Context-dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition. IEEE Transactions on Audio, Speech and Language Processing. vol. 20, no. 1, pp, 30–42, 2012.

[3] Hinton, G.; Yu D.; Dahl, G. E.; Mohamed, A.; Jaitly, N.; et al. Deep Neural Network 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.

[4] Hinton, G.; Osindero, S.; Teh, Y. W. A Fast Learning Algorithm for Deep Belief Nets. Neural Computation. vol. 18, no. 7, pp. 1527–1557, 2006.

[5] Ko, T.; Peddinti, V.; Povey, D.; Khudanpur, S. Audio augmentation for speech recognition. In proc. INTERSPEECH, pp. 3586–3589, 2015.

[6] Hsiao, R.; Ma, J.; Hartmann, W.; Karafiát, M.; František, G.; Burget, L.; et al. Robust speech recognition in unknown reverberant and noisy conditions. In proc. Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 533–538, 2015.

[7] Ko, T.; Peddinti, V.; Povey, D.; Seltzer, M. L.; Khudanpur, S. A study on data augmentation of reverberant speech for robust speech recognition. In proc. ICASSP, pp. 5220–5224, 2017.

[8] Cui, X.; Goel, V.; Kingsbury, B. Data augmentation for deep neural network acoustic modeling. IEEE/ACM Transactions on Audio, Speech and Language Processing. Vol. 23, No. 9, pp. 1469–1477, 2015.

[9] J. B. Allen and D. A. Berkley, Image method for efficiently simulating small-room acoustics. Acoust. Amer., vol. 65, no. 4, pp. 943–950, 1979.

[10] Khokhlov, Y.; Zatvornitskiy, A.; Medennikov, I.; Sorokin, I.; Prisyach, T.; Romanenko, A.; et al. R-vectors: New Technique for Adaptation to Room Acoustics. In proc. INTERSPEECH, 1243–1247, 2019.

[11] Hsu, W.-N.; Zhang, Y.; Glass, J. Unsupervised domain adaptation for robust speech recognition via autoencoder-based data augmentation. In proc. IEEE Autom. Speech Recognit. Understanding Workshop. pp. 16–23, 2017.

[12] Christensen, H.; Cunningham, S.; Fox, C.; Green, P.; Hain, T.; A comparative study of adaptive, automatic recognition of disordered speech. In proc. INTERSPEECH, pp. 1776–1779, 2012.

[13] K. C. Sim, Y. Qian, G. Mantena, L. Samarakoon, S. Kundu, and T. Tan, Adaptation of deep neural network acoustic models for robust automatic speech recognition. in New Era for Robust Speech: Exploiting Deep, S. Watanabe, M. Delcroix, F. Metze, and J. R. Hershey, Eds. Berlin, Germany: Springer, pp. 219–243, 2017.

[14] S. H. K. Parthasarathi, B. Hoffmeister, S. Matsoukas, A. Mandal, N. Strom, and S. Garimella. fMLLR based feature-space speaker adaptation of DNN acoustic models. In Proc. Interspeech, 2015, pp. 3630–3634.

[15] C. Kim et al., Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home. In Proc. INTERSPEECH, pp. 379–383, 2017.

[16] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, An overview of noiserobust automatic speech recognition. IEEE/ACM Audio, Speech, Lang. Process., vol. 22, pp. 745–777, Apr. 2014.

[17] Y. Huang and Y. Gong, Regularized sequence-level deep neural network model adaptation. In proc. INTERSPEECH, pp. 1081–1085, 2015.

[18] Y. Long, Y. Li, H. Ye, and H. Mao, Domain adaptation of latticefree MMI based TDNN models for speech recognition. Int. J. Speech Technol. pp. 171–178, 2017.

[19] J. Fainberg, S. Renals, and P. Bell, Factorised representations for neural network adaptation to diverse acoustic environments. In proc. INTERSPEECH, pp. 749–753, 2017.

[20] K. C. Sim et al., Domain adaptation using factorized hidden layer for robust automatic speech recognition. In proc. Interspeech, pp. 892–896, 2018.

[21] J. Huang et al., Cross-language transfer learning, continuous learning, and domain adaptation for end-to-end automatic speech recognition. 2020, arXiv:2005.04290.

[22] S. Ueno et al., Encoder transfer for attention-based acoustic-to-word speech recognition. In proc. Interspeech, pp. 96–108, 2018.

[23] T. Moriya et al., Progressive neural network-based knowledge transfer in acoustic models. In IEEE Asia-Pacific Signal Inf. Process. Assoc. pp. 998–1002, 2018.

[24] P. Bell, J. Fainberg, O. Klejch, J. Li, S. Renals and P. Swietojanski, Adaptation Algorithms for Neural Network-Based Speech Recognition: An Overview. In IEEE Open Journal of Signal Processing, vol. 2, pp. 33-66, 2021.

[25] Nahar, R.; Miwa, S.; Kai, A. Domain Adaptation with Augmented Data by Deep Neural Network Based Method Using Re-Recorded Speech for Automatic Speech Recognition in Real Environment, Sensors, 2022, 22, 9945.

[26] Ueda, Y.; Wang, L.; Kai, A.; Ren, B. Environment-dependent denoising autoencoder for distant-talking speech recognition. EURASIP J. Adv. Signal Process. 92 (2015), 2015.

[27] D. Povey and K. Yao. A basis representation of constrained MLLR transforms for robust adaptation. Computer Speech & Language, Vol. 26, No. 1, pp. 35-51, 2012.

[28] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In Proceedings of ICSLP96. Vol. 2, pp. 1137–1140, 1996.

[29] R. Haeb-Umbach and H. Ney. Linear discriminant analysis for improved large vocabulary continuous speech recognition. In Proceedings of ICASSP. Vol. 1, pp. 13–16, 1992.

[30] R. A. Gopinath. Maximum likelihood modeling with Gaussian distributions for classification. In Proceedings of ICASSP. Vol. 2, pp. 661–664, 1998.

[31] N. Morgan and H. A. Bourlard, Neural networks for statistical recognition of continuous speech. In proc. IEEE, vol. 83, no. 5, pp. 742–772, May 1995.

[32] Y. Chauvin, and D.E. Rumlhart. Backpropagation: Theory, Architectures, and Applications. Psychology Press, 1995.

[33] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, Vol. 313, No. 5786, pp. 504–507, 2006.

[34] G. Hinton, L. Deng, Y. Dong, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine, Vol. 29, pp. 82–97, 2012.

[35] S. Boll. Suppression of Acoustic Noise in Speech Using Spectral Subtraction. IEEE Trans. Acoust.Speech Signal Process. Vol. 27, No. 2, 1979.

[36] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal and S. Khudanpur. A pitch extraction algorithm tuned for automatic speech recognition. Proc. ICASSP, pp. 2494-2498, 2014.

[37] B. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identifications and verifications. J. Acoust. Soc. Am., Vol. 55, No. 6, 1974.

[38] Waibel, A.; Hanazawa, T.; Hinton, G.; Shikano, K.; Lang, k. Phoneme recognition using time-delay neural networks. IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 3, pp. 328–339, 1989.

[39] Waibel, A. Modular construction of time-delay neural networks for speech recognition. Neural computation, vol. 1, no. 1, pp. 39–46, 1989.

[40] Peddinti, V.; Povey, D.; Khudanpur, S. A time delay neural network architecture for efficient modeling of long temporal contexts. Proc. INTERSPEECH, pp. 3214–3218, 2015.

[41] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473, pp. 1–15, 2014.

[42] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio. End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results. arXiv:1412.1602, pp. 1–10, 2014.

[43] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. AttentionBased Models for Speech Recognition. arXiv:1506.07503, pp. 1–19, 2015.

[44] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio. End-to-end attention-based large vocabulary speech recognition. In Proceedings of ICASSP, pp. 4945–4949, 2016.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. In Advances in Neural Information Processing Systems, pp. 5999–6009, 2017.

[46] 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.

[47] A. Graves, S. Fern´andez, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of ICML, pp. 369–376, 2006.

[48] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In Proceedings of ICML, pp. 1764–1772, 2014.

[49] G. Alex. Supervised Sequence Labelling with Recurrent Neural Networks. Springer, 2012.

[50] 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.

[51] S. Kim, T. Hori, and S. Watanabe. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In Proceedings of ICASSP, pp. 4835–4839, 2017.

[52] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi. Hybrid CTC/Attention Architecture for End-to-End Speech Recognition. IEEE Journal on Selected Topics in Signal Processing, Vol. 11, No. 8, pp. 1240–1253, 2017.

[53] 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.

[54] G. E. Hinton and R. R. ShalaKhutdinov. Reducing the Dimensionality of Data with Neural Networks. Vol 313, Issue 5786, pp. 504-507, 2006.

[55] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In Proceedings of CSCCVPR, pp. 770–778, 2015.

[56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, Vol. 15, pp. 1929–1958, 2014.

[57] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer Normalization. arXiv:1607.06450, pp. 1–14, 2016.

[58] S. Schneider, A. Baevski, R. Collobert, and M. Auli, wav2vec: Unsupervised Pre-Training for Speech Recognition. In Proc. INTERSPEECH, pp.3465–3469, 2019.

[59] A. Baevski, H. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: a framework for self-supervised learning of speech representations. Proc. 34th Int. Conf. on Neural Information Processing Systems, Article No.: 1044, pp. 12449-12460, 2020.

[60] W. Hsu, B. Bolte, Y. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, Hubert: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 29, pp. 3451–3460, 2021.

[61] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, and F. Wei, Wavlm: Large-scale selfsupervised pre-training for full stack speech processing. IEEE Journal of Selected Topics in Signal Processing, vol. 16, pp. 1–14, 2022.

[62] J. Zhao and W.-Q. Zhang, Improving automatic speech recognition performance for low-resource languages with self-supervised models. IEEE Journal of Selected Topics in Signal Processing, vol. 16, no. 6, pp. 1227–1241, 2022.

[63] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, M. Auli, Unsupervised Cross-lingual Representation Learning for Speech Recognition. In Proc. INTERSPEECH, pp. 2426-2430, 2021.

[64] Corpus of Spontaneous Japanese. Available online: https://clrd.ninjal.ac.jp/csj/en/index.html (accessed on 01/09/2022)

[65] Report: Construction of the Corpus of Spontaneous Japanese, Chapter 2: Transcriptions.
Available online: https://clrd.ninjal.ac.jp/csj/en/document.html (accessed on 01/09/2022)

[66] Electronic noise database (in Japanese). Available online: http://www.sunrisemusic.co.jp/database/fl/noisedata01_fl.html (accessed on 01/09/2022)

[67] T. Akiba, et al. Overview of the NTCIR-10 SpokenDoc-2 Task. Proc. of the 10th NTCIR Conference, Tokyo, Japan, 2013.

[68] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlcek, Yanmin Qian, Petr Schwarz, et al. The Kaldi speech recognition toolkit. In Proc. IEEE 2011 Workshop, 2011.

[69] Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, o.; Goel, N.; et al. The Kaldi speech recognition toolkit. In Proc. IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, 2011.

[70] D. Yu, F. Seide and G. Li. Conversational Speech Transcription Using Context-dependent Deep Neural Networks. Proc. 29th Int. Conf. Mach. Learn., pp. 1–2, 2012.

[71] G. E. Hinton and R. R. Salakhutdinov. Reducing Dimensionality of Data with Neural Networks. Science, Vol. 313, No. 5786, pp. 504–507, 2006.

[72] Zhang, Y.; Qin, J.; Park, S. D.; Han, W.; Chiu, C. C.; Pang, R.; Le,V. Q.; Wu, Y. Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition. arXiv 2020, arXiv:2010.10504.

[73] Graves, A.; Fernandez, S.; Gomez, F.; Huber, J. S. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. Proc. 23rd International Conference on Machine Learning, pp. 25–29, 2006.

[74] Nakamura, A.; Saito, T.; Ikeda, D.; Ohta, K.; Mineno, H.; Nishimura, M. Automatic Detection of Chewing and Swallowing. Sensors 2021, 21, 3378.

[75] Jankowski, C.; Kalyanswamy, A.; Basson S.; and Spitz, J. NTIMIT: a phonetically balanced, continuous speech, telephone bandwidth speech database. Proc. ICASSP, vol.1, pp. 109–112, 1990.

[76] Brown, K. L. and George, E. B. CTIMIT: a speech corpus for the cellular environment with applications to automatic speech recognition. Proc. ICASSP, pp. vol.1, 105–108, 1995.

[77] Garofolo, J.; Lamel, L.; Fisher, W.; Fiscus, J.; Pallett, D.; Dahlgren, N. DARPA, TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM. National Institute of Standards and Technology. 1990.

[78] Gray, R. M.; Buzo, A.; Gray, A.; Matsuyama, Y. Distortion measures for speech processing. IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol. ASSP-28, No. 4, pp. 367–376, 1980.

[79] Y. Wakiya. 講義音声認識のための遠隔混入音声の影響分析とDNN音声分離モデルによる改善. Masters thesis. Mathematical and Systems Engineering course, Engineering Faculty, Integrated School of Science and Technology, Shizuoka University, 2019.

[80] ITU recommendation G.712. Transmission performance characteristics of pulse code modulation channels. 1996.

[81] Geng, M.; Xie, X.; Liu, S.; Yu, J.; Hu, S.; Liu, X.; Meng, H. Investigation of Data Augmentation Techniques for Disordered Speech Recognition. Proc. INTERSPEECH, pp. 696–700, 2020.

[82] Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; Le, Q. V. SpecAugment: A Simple Data Augmentation Method for Automatic speech Recognition. Proc. INTERSPEECH, pp. 2613–2617, 2019.

[83] Watanabe, S.; Hori, T.; Karita, S.; Hayashi, T.; Nishitoba, J.; Unno, Y.; Enrique Yalta Soplin, N.; Heymann, J.; Wiesner, M.; Chen, N.; Renduchintala, A.; Ochiai, T. ESPnet: End-to-End Speech Processing Toolkit. Proc. INTERSPEECH, pp. 2207–2211, 2018.

[84] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, T. Nakatani. Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration, Proc. INTERSPEECH, pp. 1408–1412, 2019.

[85] fairseq tool. https://github.com/facebookresearch/fairseq (accessed on 31 January 2023)

[86] S. Kornblinth, M. Nourouzi, H. Lee, G. Hinton. Similarity of Neural Network Representations Revisited, Proc. Int. Conf. on Machine Learning, pp. 3519-3529, 2019.

APPENDIX A

APPENDIX

A.1 Re-recorded Datasets used in Chapter 3 and Chapter 4

The datasets are re-recorded by playing corpus of spontaneous Japanese (CSJ) through loud speaker and recording using channels such as various telephone channels or wireless pin mic in classroom environment.

Table A.1: Speaker ID for recordings used to fine-tune baseline models for Mobile LTE channel. Female: 7, Male: 2; Total: 9 speakers

Speaker ID	**Gender**
A01F0261	F
A01F0655	F
A01F0861	F
A01F0876	F
A01F0931	F
A01F0949	F
A01M0027	M
A02F0799	F
A02M0222	M

Table A.2: Speaker ID for recordings used to train feature transformation DNN for Mobile LTE channel. Female: 8, Male: 9; Total: 17 speakers

Speaker ID	**Gender**
A03F0412	F
A03M0100	M
A04F0616	F
A04M0105	M
A05F0424	F
A05M0217	M
A06F0135	F
A06M0134	M
A07F0399	F
A07M0265	M
A08F0742	F
A08M0257	M
A09F0837	F
A09M0172	M
A10F0429	F
A11M0335	M
A13M0979	M

Table A.3: Speaker ID for recordings used to test for Mobile LTE channel (CSJ eval1). Female: 0, Male: 10; Total: 10 speakers

Speaker ID	**Gender**
A01M0097	M
A01M0110	M
A01M0137	M
A03M0106	M
A03M0112	M
A03M0156	M
A04M0051	M
A04M0121	M
A04M0123	M
A05M0011	M

Table A.4: Speaker ID for recordings used to train feature transformation DNN, fine-tuning of ASRs and test for wireless pin mic channel (CSJ eval3). Female: 5, Male: 5; Total: 10 speakers

Speaker ID	**Gender**
S00F0019	F
S00F0066	F
S00F0148	F
S00F0152	F
S00M0008	M
S00M0070	M
S00M0079	M
S00M0112	M
S00M0213	M
S01F0105	F

A.2 List of Abbreviations

1. AI: Artificial Intelligence

2. ASR: Automatic Speech Recognition
3. ANN: Artificial neural network
4. CER: Character error rate
5. CERR: Character error rate reduction
6. CMVN: cepstral mean variance normalization
7. CNN: Convolutional neural network
8. CSJ: Corpus of spontaneous Japanese
9. CTC: Connectionist temporal classification
10. DNN: Deep neural network
11. E2E: End-to-end
12. FFNN: Feed-forward neural network
13. fMLLR: Feature space maximum likelihood linear regression
14. GMM: Gaussian mixture model
15. GPU: Graphical processing unit
16. HMM: Hidden Markov model
17. LSTM: Long short-term memory
18. LTE: Long term evolution
19. MLP: Multi layer perceptron
20. RBM: Restricted Boltzmann machine
21. RIR: Room impulse response
22. RNN: Recurrent neural network
23. SNR: Signal to noise ratio

24. SSL: Self-supervised learning
25. TDNN: Time delay neural network
26. VAE: Variational autoencoder
27. VoLTE: Voice over LTE
28. VTLN: Vocal tract length normalization
29. 3G: 3^{rd} generation
30. 4G: 4^{th} generation

Related to Methods (Parts of ASR model name)

31. Base: Baseline
32. Aug: Augmented
33. C: Clean
34. N: Noisy
35. L: LTE
36. P: Pin mic
37. T: Transformed feature
38. FT: Fine-tune
39. S: Session

LIST OF PUBLICATIONS

Academic article

1. R. Nahar, S. Miwa and A. Kai, "Domain Adaptation with Augmented Data by Deep Neural Network Based Method using Re-recorded Speech for Automatic Speech Recognition in Real Environment", Sensors 22, no. 24: 9945, 2022.

Proceedings with peer review (International Conferences)

1. R. Nahar and A. Kai, "Effect of Data Augmentation on DNN-Based VAD for Automatic Speech Recognition in Noisy Environment", 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE), pp. 368-372, 2020.

2. R. Nahar, T. Kawai and A. Kai, "Multi-Condition Training of Denoising Autoencoder by Augmenting Simulated Reverberant Speech Data", 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE), pp. 334-338, 2018.

3. S. M. R. Nahar and A. Kai, "Robust Voice Activity Detector by combining sequentially trained Deep Neural Networks", 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA), pp. 1-5, 2016.

Proceedings (Domestic Conferences)

1. R. Nahar, R. Suzuki and A. Kai, "Domain Adaptation for Improving End-to-end ASR Performance of Classroom Speech with Variable Recording Condition," IEICE Technical Report, SP2022-65, pp. 153-158, Okinawa, Japan, 2023/3/1

2. R. Nahar and A. Kai, "Efficient channel adaptation of ASR by DNN-based data augmentation using re-recorded paired data with automatic alignment correction", 2021 Spring Meeting (Online) Acoustical Society of Japan 10-12 March, 1-2P-3, 2021.