DISCRIMINATIVE AND ADAPTIVE TRAINING FOR ROBUST SPEECH RECOGNITION AND UNDERSTANDING

A Dissertation
Presented to
The Academic Faculty

By

Zhong Meng

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

August 2018

DISCRIMINATIVE AND ADAPTIVE TRAINING FOR ROBUST SPEECH RECOGNITION AND UNDERSTANDING

Approved by:

Professor Biing-Hwang (Fred) Juang, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Chin-Hui Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor James H. McClellan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Elliot Moore II
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Yao Xie
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: June 22, 2018

To my wife and my parents,

for their boundless love and support

ACKNOWLEDGEMENTS

First, I would like to express my most sincere gratitute to my advisor Prof. Biing-Hwang (Fred) Juang for the conituous support of my Ph.D study. His extraordinary insight, wisdom and experience benifit my research greatly. I acquired a great deal of knowledge and developed strong interest, solid skills and research capabilty in my area from his excellent guidance and teaching. His great personality and wonderful accomplishment has made him the role model of my life.

I would like to thank the rest of my thesis committee members: Prof. Chin-Hui Lee, Prof. Elloit Moore II, Prof. James H. McClellan and Prof. Yao Xie for their precious time reading my thesis, insightful comments and encouragement. Special thanks to Prof. Chin-Hui Lee for serving as my proposal review committee chair and for his valuable suggestions during my Ph.D.

I also owe a debt of gratitute to my friends in Georgia Tech during my Ph.D. I would like to specially thank our group members Chao Weng and Umair Altaf for the collabaration on the project and research. I would like to thank my other colleges Antonio Mereno, Mehrez Souden, Zhen Huang, I-Fan Chen, Kehuang Li, You-Chi Cheng, Wei Li, Sicheng Wang, Ruolin Su and Santosh Gupta for valuable discussions. Thanks to Pat Dixon, Raquel Plaskett, Tasha Torrence and Daniela Staiculescu for their great administrative support.

I would like to express my appreciation to Dr. Jinyu Li and Dr. Yifan Gong whom I worked with at Microsoft AI and Research. I was motivated by their high standards in research and publications and inspired by their great insight in cutting-edge technologies. My thanks also go to the other excellent scientists and researchers in Microsoft whom I collabarated or discussed with: Dr. Zhuo Chen, Dr. Vadim Mazalov, Dr. Yong Zhao, Dr. Mehdi Aghagolzadeh, Dr. Yan Huang, Dr. Frank Seide, Dr. Jasha Droppo, Dr. Takuya Toshioka, Dr. Hakan Ergodan, Dr. Shixiong Zhang, Jacob Dewitt, Dr. Guoli Ye and Dr. Rui Zhao.

I would like to extend my appreciation to Dr. Shinji Watanabe, Dr. John Hershey and Dr. Takaaki Hori at Mistubishi Electric Research Lab. I am also grateful to Antonio Mereno and David Thomson at AT&T Labs Research. I really enjoyed working and discusssing with them during my internships. Special thanks to Antonio who accomadated me the entire summer in the beautiful townships of New Jersey where I encountered my love.

Finally, I would like to express my deepest gratitute to my wife Chun-Yu (Claire) and my parents for their boundless love, support and understanding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivations and Scientific Goals

Automatic speech understanding (ASU) aims at interpreting users' intentions from their spontaneous conversational speech utterances. An ASU system consists of an automatic speech recognition (ASR) component which transforms the speech signal into word hypotheses and a spoken language understanding (SLU) system that extracts the semantic representation (meaning) from the recognized text.

With the advent of deep neural network (DNN) acoustic model [1, 2], the ASR performance is greatly improved. However, ASR is still faced with great challenges when the size of the vocabulary is large and the speaking style is flexible. The frequent occurrences of words streams with no overt lexical marking of punctuations and disfluencies (i.e, filled pauses, repetitions, repairs and false starts) in a natural conversation drastically degrade the performance of an ASR system on spontaneous conversational speech [3]. Moreover, the spontaneous and irregular nature of the spoken language, no structure information (i.e., punctuation and sentence boundaries) from the ASR output and the recognition errors made by ASR degrade the performance of SLU.

Fortunately, the main idea of the spontaneous conversational speech highly depends on a set of keywords that are *semantically important*. Therefore, keyword spotting, with the goal of finding the instances of a particular set of spoken words in speech signals, becomes an important technique for the accurate understanding of spontaneous conversational speech [4, 5, 6]. To achieve robust keyword spotting on conversational speech, we propose to use the non-uniform error cost function [7] as the objective to discriminatively train the deep feedforward neural network (FNN) [8, 9] and deep bi-directional long short-term memory (BLSTM) - recurrent neural networks (RNN) [10] acoustic models so that the errors of the keywords out of all possible words in the vocabulary are minimized [11, 12]. With a densely connected FNN for acoustic modeling, the high correlations between frames can be well extracted and reflected in the intermediate representation and the distri-

bution of a concatenation of several consecutive speech frames within a long context window can be robustly modeled [13]. The deep LSTM-RNN is able to exploit dynamically changing contextual window over the input sequence rather than a static fixed-sized window used with the feedforward DNN. With a special gating mechanism to control the information to be added to or removed from the internal cell state, the LSTM architecture enables the capturing of *long-term* temporal context information and overcomes the diminishing gradient problem that comes along the RNN training. The BLSTM networks [14] process each input sequence in both directions so that the future context information are well exploited to assist in making the current prediction.

However, in most ASU systems, the ASR and the SLU components are isolated and are optimized towards different objectives. More specifically, in ASR, the decoded lattices (i.e., word hypotheses) are generated by the acoustic models that are discriminatively optimized to minimize the *senone* (tied tri-phone state), phoneme or word recognition errors, while in SLU, the lattices are analyzed based on their semantic meaning rather than their spellings or pronunciations. A high phoneme or state accuracy of the decoded lattices from the ASR does not guarantee a high semantic accuracy and is not suitable for the SLU. To cope with this problem, we propose the *minimum semantic error cost (MSEC)* training [15] of the BLSTM acoustic model, in which the expected semantic error cost of all possible word sequences on the lattices is minimized given the reference. The semantic error cost between a pair of words can be estimated via latent semantic analysis (LSA) [16] or RNN learned vector space word representations [17]. The semantic error cost between sentences can be obtained by accumulating the word-word semantic error costs on the lattices. We evaluate the performance of the MSEC training on a topic spotting task which aims at classifying the conversational utterances into one of a pre-defined set of topics.

To expand the application of the ASU system to various scenarios and achieve robust speech understanding under different environments and conditions, three important problems concerning the underlying ASR component remain to be addressed: First, the performance of speaker-independent (SI) acoustic models trained with speech data recorded from a large number of speakers is affected by the spectral variations in each speech unit caused by the inter-speaker variability. Such speaker variations lead to high variance in the spectral distribution of the speech signal that corresponds to each speech unit and thus large overlaps among distributions. Secondly, the ASR performance

suffers from large degradation when acoustic mismatch exists between the training and test conditions. Many factors contribute to the mismatch, such as variation in environment noises, channels and speaker characteristics. Thirdly, the ASR performance degrades dramatically when the speech signal is from a distant source under noisy conditions. This scenario is sometimes referred to as "far-field" or "distant-talking" in which the talker does not speak into a close-talking microphone located right in front of his or her mouth. In this scenario, the recorded speech is the source speech signal convolved with the room impulse responses, i.e., the so-called reverberant effect, plus whatever background noise that may be present. This reverberation effect is hard to estimate as the room geometry and the speaker-recorder positions are unpredictable and sometimes time-variant. In addition, the recorded signal is frequently mixed with other unexpected sound sources such as noises and interference of the other speakers.

One common solution to the first problem is the speaker-adaptive training (SAT) [18, 19, 20], which aims at generating a canonical acoustic model together with speaker-dependent (SD) transformations. By separately modeling the phonetic variations and the speaker variations, a compact acoustic model can be trained with reduced variance and overlap among different speech units. However, SAT requires two sets of models during testing, i.e., the SI model and the speaker canonical model, and it needs to be coupled with speaker adaptation. The SI model is used to generate the first-pass decoding transcription, and the speaker canonical model is combined with SD transformation to adapt to the test speaker. The final transcription is generated through the second-pass decoding. To realize the speaker normalization in a much simpler process, we propose speaker-invariant training (SIT) [21] to directly minimize the speaker variations by introducing a speaker classifier and optimizing an adversarial multi-task objective [22, 23]. SIT forgoes the need of estimating any additional SI bases or speaker representations during training or testing. The direct use of SIT DNN acoustic model in testing enables the generation of word transcription for unseen test speakers through *one-pass online* decoding. Moreover, it effectively suppresses the inter-speaker variability via a lightweight system with much reduced modeling parameters and computational complexity. To achieve additional gains, unsupervised speaker adaptation can also be further conducted on the SIT model with one extra pass of decoding.

To tackle the second problem, acoustic model adaptation can be effectively applied to compen-

sate for the acoustic mismatch between training and testing, in which the acoustic model parameters or the input features are adjusted according to the adaptation data. Unsupervised adaptation is necessary when labels of the target domain data are unavailable. It has become an important topic with the increasing amount of untranscribed speech data for which the human annotation is expensive. One possible way is to generate senone alignments against the unlabeled adaptation data through first pass decoding. However, the first pass decoding result is unreliable when the mismatch between the training and test conditions is significant. It is also time-consuming and can be hardly applied to huge amount of adaptation data. There are even situations when decoding adaptation data is not allowed because of the privacy agreement signed with the speakers. The goal of our study is to achieve *purely* unsupervised adaptation *without* any exposure to the labels or the decoding results of the adaptation data in the target domain.

Teacher-student (T/S) learning [24] is one possible approach to achieve purely unsupervised adaptation [25]. In T/S learning, the posteriors generated by the teacher model are used in lieu of the hard labels derived from the transcriptions to train the target-domain student model. Although T/S learning achieves large word error rate (WER) reduction in domain adaptation, it only implicitly handles the variations in each speech unit (e.g. senone) caused by the speaker and environment variability in addition to phonetic variations. Another approach achieving purely unsupervised adaptation is the adversarial multi-task learning [22] using gradient reversal layer network (GRLN) [23, 26, 27]. A deep intermediate feature is learned to be both discriminative for the main task of senone classification and invariant with respect to the shifts among different conditions (i.e., speakers or environments). As a comparison, T/S learning can achieve significantly better performance by using parallel training data, while adversarial training is the only possible solution when parallel data is not available.

To benefit from both methods, we first advance T/S learning with adversarial training to propose adversarial T/S learning [28] for condition-robust unsupervised domain adaptation, where a student acoustic model and a domain classifier are jointly trained to minimize the Kullback-Leibler (KL) divergence between the output distributions of the teacher and student models as well as to min-maximize the condition classification loss through adversarial multi-task learning. A senone-discriminative and *condition-invariant* deep feature is learned in the adapted student model through

this procedure. Based on this, we further propose the *multi-factorial adversarial (MFA)* T/S learning where the condition variabilities caused by multiple factors are minimized simultaneously.

Further, to make the adversarial learning method [23] more effective for unsupervised adaptation, we propose to use a domain separation network (DSN) [29, 30] to explicitly model the private component that is unique to each domain in addition to the shared component modeled in GRLN [23, 27] that is invariant to the domain shift. The shared component is learned through adversarial multi-task learning to be both discriminative to the main-task of senone classification on the source domain and invariant to the domain shift between source and target domains. The private component of each domain is trained to be orthogonal to the shared component to enhance its domain-invariance. The shared component extractor together with the senone classifier form the adapted acoustic model.

To deal with the third problem and achieve robust speech recognition in the far-field condition, multiple microphones can be used to enhance the speech signal, reduce the effects of noise and reverberation, and improve the ASR performance. In this scenario, an essential step of the ASR front-end processing is multichannel filtering, or *beamforming*, which steers a spatial sensitivity region, or "beam" in the direction of the target source, and inserts spatial suppression regions, or "nulls" in the directions corresponding to noise and other interference. We propose to adaptively estimate the beamforming filter coefficients at each time frame using an LSTM-RNN to deal with any possible changes of the source, noise or channel conditions [31]. The enhanced signal is generated by applying these time-variant filter coefficients to the short-time Fourier transform (STFT) of the array signals through filter-and-sum beamforming and is passed to a deep LSTM-RNN acoustic model to predict the senone posteriors. Further, we use hidden units in the deep LSTM acoustic model to assist in predicting the beamforming filter coefficients. The LSTM beamforming network and the LSTM acoustic model are jointly optimized to improve the ASR performance.

1.2 Contributions

The objective of the thesis is to build a robust speech recognition and understanding system through discriminative and adaptive training of the DNN acoustic models. The goal is achieved through the following approaches.

1. To achieve accurate keyword spotting on conversational speech, the non-uniform error cost MCE is used as the discriminative objective to train the deep FNN and deep BLSTM-RNN acoustic models so that the errors of keywords out of all possible words in the vocabulary are minimized. The proposed approach achieves 3%-6% and 6%-7% absolute FOM gains over cross-entropy training on the Switchboard-1 dataset and HKUST dataset.

2. To generate semantically accurate word lattices for topic spotting, MSEC objective function is proposed to train the deep BLSTM-RNN acoustic model, in which the expected semantic error cost of all possible word sequences on the lattices is minimized given the reference. The proposed method achieves 3.5% - 4.5% absolute accuracy improvement on Switchboard-1 dataset.

3. To suppress the effect of inter-speaker variability on speaker-independent DNN acoustic model, SIT is proposed to learn a deep representation in the DNN that is both senone-discriminative and speaker-invariant through adversarial multi-task training. The proposed method achieves 5%-6% relative WER improvements over the SI acoustic model on CHiME-3 dataset for ASR.

4. To achieve condition-robust unsupervised adaptation with parallel data, adversarial T/S learning is proposed to suppress multiple factors of condition variability in the procedure of knowledge transfer from a well-trained source domain LSTM acoustic model to the target domain. The proposed method achieves 3.5%-5.5% relative WER improvements over the T/S learning on CHiME-3 dataset for ASR.

5. To further improve the adversarial learning method for unsupervised adaptation without parallel data, DSNs are used to enhance the domain-invariance of the senone-discriminative deep representation by explicitly modeling the private component that is unique to each domain. The proposed method achieves 11.08% relative WER improvement over gradient reversal layer method on CHiME-3 dataset for ASR.

6. To achieve robust far-field ASR, beamforming is performed over speech signal acquired from multiple microphones. A deep LSTM-RNN is used to adaptively estimate the real-time beam-

forming filter coefficients to cope with non-stationary environmental noise and dynamic nature of source and microphones positions. The LSTM adaptive beamformer is jointly trained with a deep LSTM acoustic model to predict senone labels. The proposed method achieves 7.97% absolute WER gain over multi-style training on CHiME-3 dataset for ASR.

1.3 Thesis Outline

We focus on the proposed discriminative training methods for ASU in Chapter 2-3 and the proposed adaptive training methods for ASR in Chapter 4-8 to further expand the application of ASU to various environments and conditions.

The thesis is organized as the following. In Chapter 2, background knowledge related to the topics in this thesis is introduced, including the FNN-HMM and BLSTM-HMM acoustic models for ASR, conventional discriminative training of DNN acoustic models for ASR, and adaptation of DNN acoustic models. Chapter 3 presents the theory and formulation of non-uniform MCE, derives the backpropagation errors used to train the FNN and BLSTM acoustic model, and verify its effectiveness on Switchboard-1 Release 2 (English) and HKUST (Mandarin) dataset. Chapter 4 presents the theory and formulation of the proposed MSEC objective, derives the backpropagation errors used to train the BLSTM acoustic model and verify its effectiveness on Switchboard-1 Release 2 dataset. In Chapter 5, the speaker-invariant training is proposed to suppress the effect of speaker variability in ASR and is compared with conventional speaker-adaptive training approaches. We formulate SIT in the adversarial multi-task learning framework, describe its training and testing procedure and evaluate it on CHiME-3 dataset. In Chapter 6, we compare the T/S learning and adversarial learning methods for unsupervised adaptation. We advance the T/S learning with adversarial learning to proposed the adversarial T/S learning to achieve condition. The experiments are conducted on CHiME-3 dataset. In Chapter 7, we advance the adversarial learning approach with private component extractors and proposed to use DSN for unsupervised adaptation with unparallel data. DSN is evaluated on CHiME-3 dataset. In Chapter 8, we propose the adaptive LSTM beamforming network based on adaptive filter-and-sum to estimate the real-time beamforming filter coefficient for multichannel far-field ASR. We further integrate it with LSTM acoustic model and perform joint training of the integrated network. The experiments are conducted on CHiME-3

dataset. In Chapter 9, we conclude the thesis by listing the contributions.

CHAPTER 2

BACKGROUND

2.1 Automatic Speech Recognition System

ASR is the task of automatically converting speech signal X into a text transcription of spoken words W. As shown in Fig. 2.1, the architecture of a typical ASR system consists of five component: feature extraction, acoustic model (AM), pronunciation model, language model (LM) and decoder. The output of an ASR system is the word sequence $\hat{W}$ that maximizes the posterior prob-

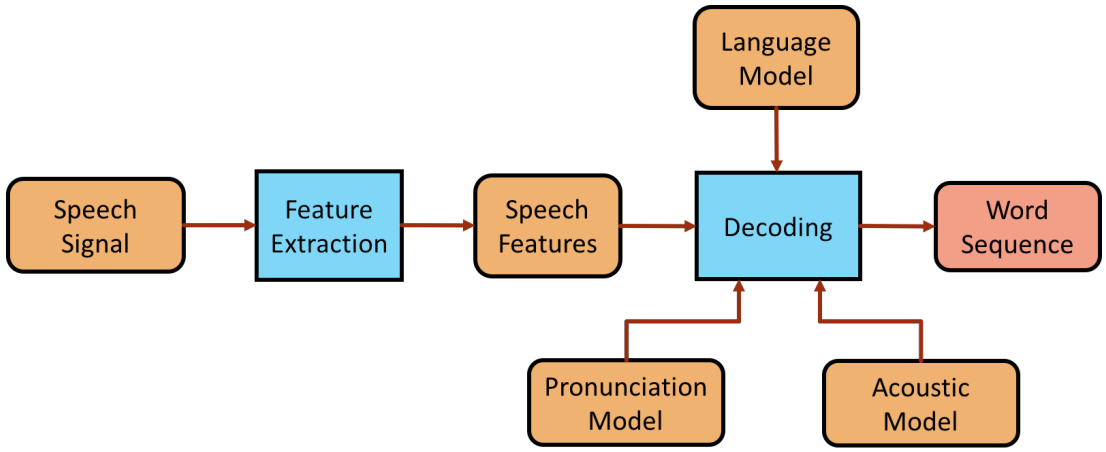


Figure 2.1: The architecture of an ASR system.

ability $P(W|X)$ given the input speech data X. ASR can be formulated as a maximum *a posterior* decision problem below:

$$\hat{W} = \arg\max_{W} P(W|X) = \arg\max_{W} \frac{P_{\Lambda}(X|W)^{\kappa} P_{\Theta}(W)}{P(X)} = \arg\max_{W} P_{\Lambda}(X|W)^{\kappa} P_{\Theta}(W) \quad (2.1)$$

where $P_{\Lambda}(X|W)$ is the acoustic model likelihood with parameters Λ, κ is the scaling factor for the acoustic model likelihood and $P_{\Theta}(W)$ is the language model probability with parameters Θ. In the training stage, we want to estimate the optimal acoustic and language model parameters $\hat{\Lambda}$ and $\hat{\Theta}$ given the training data. During testing, the decoder will find the optimal word sequence $\hat{W}$ given $\hat{\Lambda}$ and $\hat{\Theta}$ through proper search, often involving some form of dynamic programming.

2.1.1 Feature Extraction

The speech signal is produced by passing an excitation signal through a slowly time-varying linear system. For the voiced speech, the excitation signal is in the form of a quasi-periodic glottal wave and for the unvoiced speech, it is represented by random noise. In the speech production procedure, the linear system is essentially a vocal tract that reforms the spectrum of the speech coming out of the lips. For the ASR task, the short-time spectrum is the most fundamental feature representation. The short-time spectrum may undergo further transformation to become feature and can be augmented with auxiliary features for better performance (e.g., pitch feature for tonal languages).

The most popular acoustic features have been mel-frequency cepstral coefficients (MFCC) [32]. To generate MFCC features, a pre-emphasis filter is first applied on the signal to amplify the high-frequency components since they normally have smaller magnitudes than the low-frequency ones. The pre-emphasis filter can be implemented as a first-order high-pass filter and the filtered signal in time domain is represented as the following:

$$y_t = x_t - \alpha x_{t-1} \tag{2.2}$$

where the values for the filter coefficient α are typically around 0.95.

After pre-emphasis, short but overlapping segments of speech are successively extracted and the frequencies in a speech signal are assumed to be stationary over a very short period of time. The frame size is typically set at 25 ms with a stride of 10 ms (15 ms overlap). After framing the speech signal, a window function (e.g. Hamming window) is applied to each frame as follows:

$$w_n = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) \tag{2.3}$$

where N is the window length and $0 \leq n \leq N-1$.

Then an N-point fast Fourier transform (FFT) is conducted on each frame to calculate the frequency spectrum and this is called a short-time Fourier transform (STFT). A power spectrum is

computed from the discrete STFT coefficients afterwards.

$$Y_{i,k} = \sum_{n=-\infty}^{+\infty} y_n w_{n-i} e^{-j\frac{2\pi}{N}kn} \tag{2.4}$$

$$P_{i,k} = \frac{1}{N}||Y_{i,k}||_2 \tag{2.5}$$

where $Y_{i,k}$ and $P_{i,k}$ are the complex STFT coefficient and power spectrum for the discrete time-frequency index $(i,k), k = 0, \ldots, N-1$ respectively.

Then we apply triangular filters (typically 40 filters) evenly distributed on a Mel-scale to the power spectrum to extract frequency bands. The Mel-scale aims to mimic the non-linear human perception of sound which is more discriminative at lower frequencies and less discriminative at higher frequencies. The relationship between f in Hertz and m in Mel is the following:

$$m = 2595 \log_{10}(1 + \frac{f}{700}) \tag{2.6}$$

$$f = 700(10^{\frac{m}{2595}} - 1) \tag{2.7}$$

The frequency response of each triangular filter in the filter bank is 1 at the central frequency and decreases linearly to 0 until it reaches the central frequencies of the two adjacent filters where the response is 0. Normally the triangular filters are spread over the whole frequency range from zero up to the Nyquist frequency. Band-limiting is often useful to reject unwanted frequencies or avoid allocating filters to frequency regions in which there is no useful signal energy. The log Mel filter bank coefficients $Z_{i,m}$ at time i are

$$Z_{i,m} = \log\left(\sum_{k=0}^{N-1} P_{i,k} H_{m,k}\right), \quad 0 \leq m \leq M-1 \tag{2.8}$$

where $H_{m,k}$ is the m th triangular filter in the filter bank.

So far, the filter bank coefficients computed in the previous steps are highly correlated. However, when using Gaussian mixture model (GMM) - HMM system for acoustic modeling, the covariance matrix is diagonal with assumption that feature coefficients (dimensions) are independent from each other. Therefore, we need to apply discrete cosine transform (DCT) to decorrelate the filter bank

coefficients as follows:

$$c_{i,p} = \sum_{m=0}^{M-1} Z_{i,m} \cos\left(\frac{\pi p}{M}(m+0.5)\right), \quad 0 \leq p \leq M-1 \tag{2.9}$$

where $c_{i,p}$ is the MFCC coefficients at time i. For ASR, the resulting cepstral coefficients 2-13 are retained and the rest are discarded because they represent fast changes in the filter bank coefficients and do not contribute so much to ASR performance.

MFCC features are widely used in both GMM-HMM and DNN-HMM systems. In real application, the speech signal goes through a transmission channel before reaching the receiver and the received speech signal is equivalent to multiplying the speech spectrum by the channel transfer function which is assumed constant during the utterance. In the log cepstral domain, this multiplication becomes an addition of a constant cepstrum vector which represents the channel effect and can be removed by subtracting the cepstral mean from all the feature frames. This cepstral mean normalization (CMN) technique is very effective in compensating for the long-term cepstral effect caused by different microphones, audio channels and so on.

With a strong capability of modeling the high correlations among different dimensions of the input features [33], the DNN-HMM systems appear to work better with log Mel filter-bank features in Eq. 2.8 without because as a linear transformation, DCT discards some information in the speech signal which is highly non-linear.

2.1.2 Acoustic Model and Pronunciation Model

The acoustic model integrates knowledge about acoustics and phonetics. It takes the features generated from the feature extraction component as the input and generates the likelihood (acoustic model score) $P(X|W)$ of the variable-length feature sequence. The goal of acoustic modeling is to establish the statistical representations for the feature vector sequences computed from the speech waveform. Acoustic modeling plays a critical role in improving the recognition accuracy and it is the key component of an ASR system.

Conventionally, GMM-HMM acoustic model is the most popular acoustic model in which an HMM characterizes the temporal variability of speech and a GMM models the emitting probability

of each HMM state which quantifies how well each state of an HMM fits a feature frame that represents the acoustic input. HMM changes state once every time frame, and at each time frame when a state j is entered, an observation feature vector x_t is generated from the emitting probability distribution. In the HMM, the transition probability a_{ij} represents the probability of entering state j given the previous state j. Assume s_t is the state index at time t. For an N-state HMM, we have

$$a_{ij} = P(s_t = j | s_{t-1} = i) \tag{2.10}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{2.11}$$

The emitting probability $b_j(x)$ describes the distribution of the observation vectors x at the state j. In a continuous-density HMM, $b_j(x)$ is represented by a GMM as follows:

$$b_j(x) = \sum_{m=1}^{M} c_{j,m} \mathcal{N}(x; \mu_{jm}, \Sigma_{jm}) \tag{2.12}$$

where c_{jm}, μ_{jm} and Σ_{jm} are the weight, mean and covariance of the m th Gaussian component of the mixture distribution $\mathcal{N}(x; \mu_{jm}, \Sigma_{jm})$ at state j.

With the advent of deep learning, deep neural networks have replaced the GMM in acoustic modeling and the DNN-HMM hybrid systems have achieved the state-of-the-art performance for ASR. We will elaborate DNN-HMM acoustic models in Sections 2.2 and 2.3.

Pronunciation model translates each word or phrase into a sequence or multiple sequences of fundamental speech units (e.g., phonemes, phonetic features), which is an important link between the acoustic model and the language model in an ASR system. In general, a pronunciation model is built from a knowledge-based lexicon which specifies the mapping relationships between each word and its phonetic transcription in the vocabulary. We perform grapheme-to-phoneme conversions for the out-of-vocabulary (OOV) words.

2.1.3 Language Model

A statistical language model is a probability distribution $P(W)$ over a sequence of words W which reflects how frequently the string W occurs. The most widely used language model for ASR is

the n-gram language model which gives an estimate of the probability of a certain word given its history. The language model $P(W)$ can be factorized as follows.

$$\begin{aligned} P(W) &= P(w_1, w_2, \ldots, w_n) \\ &= \prod_{i=1}^{n} P(w_i | w_1, w_2, \ldots, w_{i-1}) \end{aligned} \tag{2.13}$$

As a simplification, the n-gram model assumes that $P(w_i|w_1, w_2, \ldots, w_{i-1})$ depends only on $n-1$ previous words $w_{i-(n-1)}, w_{i-(n-2)}, \ldots, w_{i-1}$. Therefore, the n-gram language model can be decomposed as

$$P(W) \approx \prod_{i=1}^{n} P(w_i | w_{i-(n-1)}, w_{i-(n-2)}, \ldots, w_{i-1}) \tag{2.14}$$

The conditional probability can be computed from n-gram frequency counts:

$$P(w_i | w_{i-(n-1)}, w_{i-(n-2)}, \ldots, w_{i-1}) = \frac{C(w_{i-(n-1)}, w_{i-(n-2)}, \ldots, w_{i-1}, w_i)}{C(w_{i-(n-1)}, w_{i-(n-2)}, \ldots, w_{i-1})} \tag{2.15}$$

where $C(\cdot)$ is the count of an n-gram term.

Smoothing is widely used to deal with the data sparseness problem in language modeling for ASR. The popular smoothing algorithms are Jelinek-Mercer [34] smoothing, Katz backoff [35], Witten-Bell smoothing [36], and Kneser-Ney smoothing [37]. It has been shown in [38] that Kneser-Ney smoothing works better than other smoothing methods because of its unique back-off distribution where a fixed discount was subtracted from each nonzero count. Recently, neural network language models [39, 40] are introduced and have achieved the state-of-the-art performance by using RNN.

2.1.4 Decoder

In ASR, the goal of a decoder is to find most probable sequence of words given the acoustic model, language model and the test input features. Weighted finite state transducer (WFST) [41, 42] which compiles major components of ASR including HMMs, context-dependency models, pronunciation dictionaries, language model is commonly used for LVCSR task. With WFST decoder, an $HCLG$

decoding graph is first constructed as follows:

$$HCLG = min(det(H \circ C \circ L \circ G)) \tag{2.16}$$

where H, C, L and G represent the HMM structure, the phonetic context-dependency, the lexicon and the grammar, respectively and $\circ$ is WFST composition operation. In the $HCLG$ graph, the input labels are the context-dependent HMM states, and the output labels represent words.

When we want to decode an utterance of T frames, our goal is to find the most likely word sequence and its corresponding state-level alignment. We construct an weight finite state acceptor (WFSA) U with $T + 1$ states and an arc for each combination of time and context-dependent HMM states. The costs on the arcs correspond to negated and scaled acoustic log-likelihoods. A search graph S of the test utterance is created by composing U with $HCLG$, i.e.,

$$S = U \circ HCLG \tag{2.17}$$

The search graph S has approximately $T + 1$ times more states than $HCLG$ decoding graph. The decoding problem becomes finding the best path in S. The input label sequence on the best path corresponds to the state-level alignment and the output label sequence represents the sentence. Instead of a full search of S, a more practice way is to perform beam pruning and conduct Viterbi decoding on the searched subset of S. The search subset includes a subset of the states and arcs of S generated by heuristic pruning.

2.2 Deep FNN Acoustic Models for ASR

As shown in Fig. 2.2, in DNN-HMM acoustic models, DNN are directly used to model the emitting probability of each HMM state which generates the acoustic score. As a type of DNN, deep feedforward neural networks (FNN) with multiple hidden layers are trained to model the multi-frame distributions over senones (tied tri-phone states) as its output and have achieved remarkable performance improvement on almost all challenging LVCSR tasks [1, 2][1].

[1] In [13], it is reported that if DNN only uses one frame at a time, the performance is not as good as traditional GMM models.

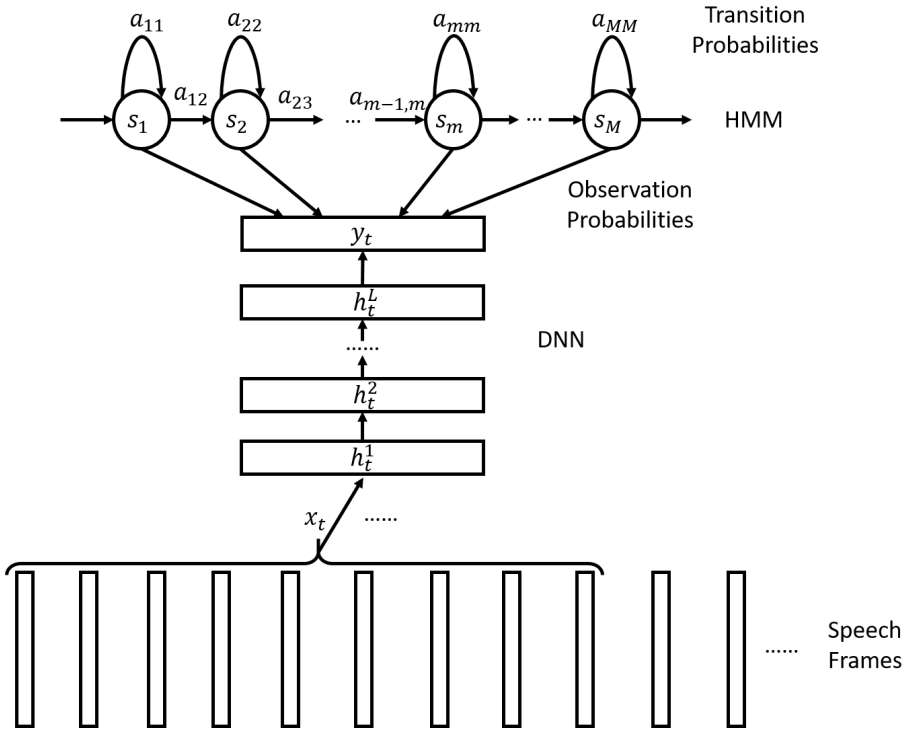


Figure 2.2: The architecture of a DNN-HMM hybrid acoustic model for ASR.

The middle layers of the deep FNN are a hierarchy of nonlinear intermediate representations that capture the complex statistical characteristics in data while the upper layers are multilayer perceptron (MLP) at the upper layers classifies the intermediate representation to different senones [33]. These intermediate representations are first generated through the generative pre-training of a stack of restricted Boltzmann machine (RBM) [9] and are then discriminatively fine-tuned to predict the senones with a certain objective through backpropagation [8]. The densely connected deep FNN is able to well extract the high correlations between speech frames via the intermediate representation and accurately model the probabilistic distribution of a splice of several consecutive speech frames within a long context window.

Assume that $X = \{x_1, \ldots, x_T\}$ is the sequence of training speech features to the input of the FNN where $x_t, 1 \leq t \leq T$ is typically a concatenation of 11 frames of acoustic features. FNN takes observation vectors X_r as the input and pass it through many layers of linear and non-linear

transformations as follows:

$$a_t^1 = W_1 x_t + b_1 \tag{2.18}$$

$$h_t^1 = \sigma(a_t^1) \tag{2.19}$$

$$a_t^l = W_l h_t^{l-1} + b_l, \quad 2 \leq l \leq L \tag{2.20}$$

$$h_t^l = \sigma(a_t^l) \tag{2.21}$$

where W_l and b_l are the weight matrix and bias vector for the l th hidden layer and h_t^l is the vector of hidden units in the l th hidden layer at time t. $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function applied element-wise. a_t^l is the vector of activations at the l th hidden layer before sigmoid.

The output of the deep FNN for senone s is the posterior probability $p(s|x_t)$ obtained by a softmax function.

$$p(s|x_t) = y_t(s) = \frac{\exp[a_t^y(s)]}{\Sigma_{s'=1}^{S} \exp[a_t^y(s')]} \tag{2.22}$$

where $a_t^y(s)$ is the activation for senone s at the output layer, $s \in \{1, \ldots, S\}$ and S is the total number of senones. y_t is the vector of output units of the FNN at time t with a dimension of S. The pseudo log-likelihood of observation x_{rt} given senone s is

$$\log p(x_t|s) = \log p(s|x_t) - \log p(s) + \log p(x_t) \tag{2.23}$$

where $p(s)$ is the prior probability of senone s estimated from the training set and $p(x_t)$ is the probability of observation x_t which is independent of the word sequence and can be ignored.

The parameters Λ of the deep FNN are trained to minimize the cross entropy (CE) objective below

$$\begin{aligned} \mathcal{L}_{\text{CE}}(\Lambda) &= -\frac{1}{T} \sum_{t=1}^{T} \sum_{s=1}^{S} \hat{y}_t(s) \log p(s|x_t) \\ &= -\frac{1}{T} \sum_{t=1}^{T} \sum_{s=1}^{S} \hat{y}_t(s) \log y_t(s) \end{aligned} \tag{2.24}$$

where $\hat{y}_t$ is the S dimensional one-hot target vector obtained from a hard alignment.

$$\hat{y}_t(s) = \begin{cases} 1, & s = s_t \\ 0, & s \neq s_t \end{cases} \tag{2.25}$$

where s_t is the senone label the input feature x_t is aligned with.

The parameters of the deep FNN is optimized using backpropagation. The backpropagation error at the output layer of the network is the derivative of $\mathcal{L}_{\text{CE}}(\Lambda)$ with respect to each dimension of the output layer activation vector a_t^y as follows:

$$\frac{\partial \mathcal{L}_{\text{CE}}(\Lambda)}{\partial a_t^y(s)} = y_t(s) - \hat{y}_t(s) \tag{2.26}$$

The gradients for all the parameters of the deep FNN can be computed by backpropagating the error in Eq. 2.26. The optimization is conducted using minibatch based stochastic gradient descent (SGD) [43].

2.3 Deep BLSTM Acoustic Models for ASR

The other type of DNN widely used for the acoustic modeling is deep LSTM-RNN. The LSTM network, a special kind of RNN with purpose-built memory cells to store information, have been successfully applied to many sequence modeling tasks. Recently, LSTM-based acoustic modeling has achieved improved performance over FNNs [1, 2] and conventional RNNs [44, 45] for LVCSR as they are able to model temporal sequences and long-range dependencies more accurately than others especially when the amount of training data is large. LSTM has been successfully applied in both the LSTM-HMM hybrid systems [46, 47, 48, 49] and the end-to-end system [50, 51, 52]. In LSTM-HMM hybrid system, the LSTM is directly used to model the emitting probability for each HMM state which generates the acoustic score.

For acoustic modeling, the LSTM takes in a sequence of input speech frames $X = \{x_1, \ldots, x_T\}$

and computes the hidden vector sequence $H = \{h_1, \ldots, h_T\}$ by iterating the equation below

$$h_t = \text{LSTM}(x_t, h_{t-1}) \tag{2.27}$$

where LSTM$(\cdot)$ denote the hidden layer function of the LSTM. In this paper, we implement Eq. (2.27) with the LSTM introduced in [53] as follows

$$i_t = \sigma(W_{x,i}x_t + W_{h,i}h_{t-1} + W_{c,i}c_{t-1} + b_i) \tag{2.28}$$

$$f_t = \sigma(W_{x,f}x_t + W_{h,f}h_{t-1} + W_{c,f}c_{t-1} + b_f) \tag{2.29}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{x,c}x_t + W_{h,c}h_{t-1} + b_c) \tag{2.30}$$

$$o_t = \sigma(W_{x,o}x_t + W_{h,o}h_{t-1} + W_{c,o}c_t + b_o) \tag{2.31}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.32}$$

where i, f, o, c are the input gate, forget gate, output gate and cell state respectively, all of which are the same dimension as the hidden units vector h_t, σ is the logistic sigmoid function and $\odot$ stands for point-wise product. The weight matrix subscripts indicates the input and the gate (e.g., $W_{h,i}$ is the hidden-input gate matrix, etc.) The weight matrices from the cell to gate vectors (e.g. $W_{c,i}$, etc.) are diagonal.

A deep LSTM stacks multiple LSTM hidden layers on top of each other, with the output sequence of one layer forming the input sequence for the next hidden layer. With a deep architecture, we are able to progressively learn higher level representations of the acoustic data and capture the high correlations between speech frames within a dynamic size of context window. The deep BLSTM processes the sequence of speech frames from both directions. It computes the forward hidden vector sequence $\overrightarrow{H} = \{\overrightarrow{h}_1, \ldots, \overrightarrow{h}_T\}$, the backward hidden vector sequence $\overleftarrow{H} = \{\overleftarrow{h}_1, \ldots, \overleftarrow{h}_T\}$ and the output sequence of senone posterior $Y = \{y_1, \ldots, y_T\}$ by iterating the backward layer from $t = T$ to 1, the forward layer from $t = 1$ to T and then updating the output layer. For each time, the output from both the forward and backward hidden layers are concatenated and then fed as the input of the next forward and backward hidden layers or the output layer.

To reduce the number of trainable parameters and alleviate the computational complexity, we

introduce a separate linear projection layer after each BLSTM layer as in [47]. We connect each hidden layer to a recurrent projection layer with a reduced number of units before recurrently feeding the projection layer back to the BLSTM input. The deep BLSTM acoustic model in this work is formulated as follow.

$$\overrightarrow{h}_t^1 = \text{LSTM}_1^{\text{forward}}(x_t, \overrightarrow{h}_{t-1}^1) \tag{2.33}$$

$$\overleftarrow{h}_t^1 = \text{LSTM}_1^{\text{backward}}(x_t, \overleftarrow{h}_{t+1}^1) \tag{2.34}$$

$$\overrightarrow{h}_t^n = \text{LSTM}_n^{\text{forward}}(p_t^{n-1}, \overrightarrow{h}_{t-1}^n), \quad n = 2, \ldots, N \tag{2.35}$$

$$\overrightarrow{p}_t^n = W_{\overrightarrow{h}^n, \overrightarrow{p}^n} \overrightarrow{h}_t^n, \quad n = 1, \ldots, N \tag{2.36}$$

$$\overleftarrow{h}_t^n = \text{LSTM}_n^{\text{backward}}(p_t^{n-1}, \overleftarrow{h}_{t+1}^n), \quad n = 2, \ldots, N \tag{2.37}$$

$$\overleftarrow{p}_t^n = W_{\overleftarrow{h}^n, \overleftarrow{p}^n} \overleftarrow{h}_t^n, \quad n = 1, \ldots, N \tag{2.38}$$

$$p_t^n = (\overleftarrow{p}_t^n, \overrightarrow{p}_t^n), \quad n = 1, \ldots, N \tag{2.39}$$

$$y_t = \text{softmax}(W_{p^N y} \tanh(p_t^N) + b_y) \tag{2.40}$$

where $\text{LSTM}_n^{\text{forward}}(\cdot)$ and $\text{LSTM}_n^{\text{backward}}(\cdot)$ denote the forward and backward n^{th} hidden layer functions of the LSTM respectively. $\overrightarrow{p}_t^n$ and $\overleftarrow{p}_t^n$ are the projection vectors of forward and backward hidden vectors $\overrightarrow{h}_t^n$ and $\overleftarrow{h}_t^n$ respectively at the n^{th} layer. $W_{\overrightarrow{h}^n, \overrightarrow{p}^n}$ and $W_{\overleftarrow{h}^n, \overleftarrow{p}^n}$ are the projection matrices. p_t^n is the concatenation of forward and backward projection vectors $\overrightarrow{p}_t^n$ and $\overleftarrow{p}_t^n$. y_t is the senone posterior output vector given input speech frame x_t. The parameters of the deep BLSTM is optimized using backpropagation through time (BPTT) with SGD.

2.4 Discriminative Training of DNN Acoustic Models

Conventionally, DNNs are trained to model the distribution of the senones based on a cross-entropy criterion in LVCSR tasks. A senone-level alignment on the training set is used as the labels for training the DNN. However, the DNNs trained through distribution estimation do not necessarily lead to the minimization of the recognition error rate. Therefore, many discriminative training criteria are proposed to directly optimizes the performance metric, e.g., WER, and are used to train the

acoustic models according to the new objective. The most popular discriminative training methods include minimum classification error (MCE) [54, 55], maximum mutual information (MMI) [56, 57], minimum phone error (MPE) [58, 59], state-level minimum Bayes risk (sMBR) [60, 61, 62], minimum word error (MWE) [63, 64] and boosted MMI [65] are used as the objective for DNN acoustic model training [66].

MCE is the first dicriminative training objective that directly minimizes the recognition errors. Assume that the training utterances $X_r, r = \{1, \ldots, R\}$ and W_r is the word transcription for X_r. W'_r represents one of the hypothesized word sequences for X_r including W_r. The discriminative function is defined as:

$$g(X_r, W'_r; \Lambda) = \log[P_\Lambda(X_r|W'_r)^\kappa P(W'_r)] \tag{2.41}$$

where Λ is the parameters of the acoustic model.

The misclassification measure is thus given by

$$d(X_r, W_r; \Lambda) = -g(X_r, W_r; \Lambda) + \log \left\{ \frac{1}{C(W'_r) - 1} \sum_{W'_r \neq W_r} \exp[g(X_r, W'_r; \Lambda)\eta] \right\}^{\frac{1}{\eta}} \tag{2.42}$$

The misclassification measure is then embeded into a sigmoid function

$$l(d(X_r, W_r; \Lambda)) = \frac{1}{1 + \exp(-\alpha d(X_r, W_r; \Lambda) + \beta)} \tag{2.43}$$

The slope of the sigmoid curve can be adjusted by α and β is normally set to 0. The MCE loss function then becomes

$$\mathcal{L}_{\text{MCE}}(\Lambda) = \sum_{r=1}^{R} l(d(X_r, W_r; \Lambda)) \tag{2.44}$$

The $\mathcal{L}_{\text{MCE}}(\Lambda)$ is essentially a smoothed approximation of the *empirical error rate*.

The MMI criterion maximize the posterior probability of the correct sentence W_r given the

utterance X_r. The objective function of the MMI criterion is

$$\begin{aligned}\mathcal{L}_{\text{MMI}}(\Lambda) &= \sum_{r=1}^{R} \log P_\Lambda(W_r|X_r) \\ &= \sum_{r=1}^{R} \log \frac{P_\Lambda(X_r|W_r)^\kappa P(W_r)}{P(X_r)} \\ &= \sum_{r=1}^{R} \log \frac{P_\Lambda(X_r|W_r)^\kappa P(W_r)}{\sum_{W_r'} P_\Lambda(X_r|W_r')^\kappa P(W_r')}\end{aligned} \tag{2.45}$$

The MPE directly maximizes the smoothed phoneme transcription accuracy. The objective function of MPE is the expected phoneme transcription accuracy as follows:

$$\begin{aligned}\mathcal{L}_{\text{MPE}}(\Lambda) &= \sum_{r=1}^{R} \sum_{W_r'} P_\Lambda(W_r'|X_r) A(W_r', W_r) \\ &= \sum_{r=1}^{R} \sum_{W_r'} \frac{P_\Lambda(X_r|W_r')^\kappa P(W_r') A(W_r', W_r)}{P(X_r)} \\ &= \sum_{r=1}^{R} \sum_{W_r'} \frac{P_\Lambda(X_r|W_r')^\kappa P(W_r') A(W_r', W_r)}{\sum_{W_r''} P_\Lambda(X_r|W_r'')^\kappa P(W_r'')}\end{aligned} \tag{2.46}$$

where $P_\Lambda(W_r'|X_r)$ is the posterior of the hypothesized word sequence W_r'. $A(W_r', W_r)$ represents the phoneme accuracy which equals the number of phonemes in the reference transcription W_r minus the number of phoneme errors made in W_r'. The objective function in Eq. 2.46 becomes sMBR and MWE when $A(W_r', W_r)$ represents the state-level and word-level accuracies respectively of the hypothesized word sequence W_r'.

To train a DNN acoustic model with discriminative objective, the gradients with respect to the activations at the output layer are first computed. The gradients for all the parameters of the network can be derived from this quantity based on the back-propagation procedure[66].

2.5 Adaptation of DNN Acoustic Models

ASR still suffers from large performance degradation when acoustic mismatch exists between the training and test conditions [67]. Many factors contribute to the mismatch, such as variation in environment noise, channels and speaker characteristics. Acoustic model adaptation is an effective

way to address this limitation, in which the acoustic model parameters or input features are adjusted to compensate for the mismatch.

The biggest challenge for the adaptation of DNN acoustic model is the limited adaptation data from the target domain. In this scenario, the acoustic model can be easily overfitted to the small amount of adaptation data. To address this issue, regularization-based approaches are proposed to regularize the neuron output distributions or the DNN model parameters. In [68], the senone distribution estimated from the adapted model is forced to be close to that from the unadapted model by adding Kullback-Leibler divergence regularization to the adaptation criterion. In [69, 70], the affine transformations of the DNN acoustic model are regarded as random Gaussian variables and are learned through maximum *a posteriori* estimation (MAP) [71] by incorporating prior knowledge into the adaptation process. More recently, Huang et.al [72] proposed the multi-task learning approach for the rapid adaptation of the DNN acoustic model, in which the senone classification is performed as the primary task and the mono-phone/senone-cluster classification is conducted simultaneously as the secondary task to alleviate the effect of senone sparseness in the limited adaptation data. In [73], teacher-student (T/S) learning [74] is proposed to adapt the DNN acoustic model without any transcription. With T/S learning, the posterior probabilities generated by a well-trained source-domain teacher network are used in lieu of labels to train the target-domain student network so that the student network can mimic the behavior of the teacher via knowledge distillation.

The second class of DNN adaptation methods is a transformation-based approach which aims at reducing the number of learnable parameters. In [75, 76], a linear transformation network (LIN) is inserted into the input, hidden or output of a well-trained unadapted DNN and the parameters of the LIN are trained to minimize the errors at the output of the DNN while the parameters of the original DNN are fixed. In [77, 78], the weight matrix of an unadapted DNN is factorized as the product of two low-rank matrices using singular value decomposition (SVD). The adaptation is then performed by updating a small-footprint square matrix inserted in between the two low-rank matrices.

In addition, auxiliary features are used to learn a canonical DNN acoustic model to achieve a better adaptation performance. In [79], an i-vector [80] that represents the speaker identity is used as an input feature in parallel with the regular acoustic features. The i-vector for a speaker is concatenated to every frame that belongs to that speaker and changes across different speakers for

both training and testing. In [81, 82], speaker codes are connected to the unadapted DNN through a set affine transformations, which are estimated together with the speaker codes during training. During adaptation, only the speaker code is re-estimated using the adaptation data.

Recently, adversarial training has become a very hot topic in deep learning because of its great success in estimating generative models [22]. It was first applied to the area of unsupervised domain adaptation by Ganin et al. in [23] in a form of multi-task learning. In their work, the unsupervised adaptation is achieved by learning deep intermediate representations that are both discriminative for the main task on the source domain and invariant with respect to mismatch between source and target domains. The domain invariance is achieved by the adversarial training of the domain classification objective functions. This can be easily implemented by augmenting any feed-forward models with a few standard layers and a *gradient reversal layer (GRL)*. This GRL approach has been applied to acoustic models for unsupervised adaptation in [27] and for increasing noise robustness in [26, 83]. Improved ASR performance is achieved in both scenarios.

CHAPTER 3
NON-UNIFORM MINIMUM CLASSIFICATION ERROR TRAINING FOR KEYWORD SPOTTING

3.1 Introduction

Large vocabulary continuous speech recognition (LVCSR) has achieved extraordinary performance when the speech is read or dictated. For instance, a word accuracy higher than 90% can be expected on the Wall Street Journal task. However, this performance decreases tremendously on a spontaneous conversational speech recognition task [1] as it consists of a stream of words with no overt lexical marking of punctuations and disfluencies (i.e, filled pauses, repetitions, repairs and false starts) may occur frequently in a natural conversation [3]. However, in real applications, it is more important to semantically understand a spontaneous speech rather than to recognize its word transcription. Moreover, the semantic meaning generally resides in a set of keywords in the spoken utterances. For instance, in the automatic topic classification task, each topic could have strong associations with a certain group of keywords and the occurrences of these words in the input speech may lead to its correct topic label. In the utterance "Exactly, it wouldn't be nice if it started raining. It's too hot.", the keywords "raining" and "hot" are strong indicators of the topic "weather", while in the utterance "Would you prefer waffle or pancake for breakfast?", the keywords "waffle" and "pancake" suggest the topic label of "food". Therefore, keyword spotting techniques become crucial for spontaneous conversational speech recognition tasks. Therefore, keyword spotting techniques become crucial for spontaneous conversational speech recognition tasks.

Many techniques have been proposed for the keyword spotting task. In [84], an optimum dynamic programing (DP) based time-normalization algorithm is proposed for spoken word recognition. In 1990s, a hidden Markov model (HMM) based keyword spotting system is proposed within the framework of hypothesis testing [85]. In [86], a set of hypothesized word transcriptions are first generated by the LVCSR decoder and the keywords are then detected and verified. Although good performance is achieved, the two stages in this approach are isolated and optimized based on

different criteria. To circumvent this problem, the keyword spotting is formulated as a non-uniform error LVCSR task and the method of *non-uniform minimum classification error (MCE)* is proposed in [7]. In conventional LVCSR, discriminative training (DT) is applied to refine the models with the objective of minimizing the recognition errors without any emphasis on the keywords. However, with non-uniform error LVCSR, the non-uniform error cost is embedded in the DT process to minimize the errors of some words (i.e., keywords) out of all possible words in the vocabulary. This idea is implemented efficiently in the weighted finite state transducer (WFST) framework and has shown some improvement over the baseline system. Moreover, this work is built upon a GMM-HMM system where a GMM is used to model the probability distribution of input features that are associated with a state of an HMM. With an adequate number of mixture components, GMMs are able to accurately model an arbitrary distribution. The parameters of a GMM can be fine-tuned discriminatively to minimize the non-uniform MCE objective specially designed for keyword spotting.

However, GMMs with diagonal covariance matrices are not good at handling highly correlated frames and the concatenation of neighboring frames will inevitably bring about the curse of dimensionality issue during model training. Recently, deep feedforward neural networks (FNN) with multiple hidden layers are trained to model the multi-frame distributions over *senones* (tied triphone states) as its output and have achieved remarkable performance improvement on almost all challenging LVCSR tasks [1, 2]. The resulting deep FNNs learn a hierarchy of nonlinear intermediate representations at the middle layers that capture the complex statistical characteristics in data and the multilayer perceptron (MLP) at the upper layers classifies the intermediate representation to different senones [33]. These intermediate representations are first generated through the generative pre-training of a stack of restricted Boltzmann machine (RBM) [9] and are then discriminatively fine-tuned to predict the senones with a certain objective through backpropagation [8]. By using densely connected FNN for acoustic modeling, the high correlations between frames can be well extracted and reflected in the intermediate representation and the distribution of a concatenation of several consecutive speech frames within a long context window can be robustly modeled [13].

Therefore, we propose a *non-uniform MCE training of a deep FNN* for keyword spotting, in which a deep FNN is discriminatively trained to minimize the empirical error cost. The backpropagation error based on non-uniform MCE is derived for updating the parameters in deep FNNs.

When applying this to LVCSR, a sequence of decoded words will be produced similar to the usual word error rate (WER) based LVCSR, except that the keywords will have fewer recognition errors. To further improve the performance, we boost the likelihood of the hypothesized word sequences in proportion to their phone error rate, which is equivalent to generating more confusable data for the discriminative training. Therefore, a *non-uniform boosted MCE (BMCE) training of a deep FNN* is proposed to incorporate this data augmentation strategy in training. Experiments are conducted on a large-scale spontaneous conversational telephone speech (CTS) dataset. The proposed method achieves 3.65% absolute figure of merit (FOM) gain over the baseline system using cross entropy as the objective on "Credit Card Use" topic of Switchboard-1 Release 2.

However, deep FNNs can only make use of limited contextual information by taking a fixed-size window of speech frames as the input to make the prediction. Although they successfully model the high correlations between frames within a fixed and short time interval, they fail to capture the long-term dependencies within the entire speech signal and are not able to handle dynamic speaking rates. By using recurrent nueral networks (RNN), the network activations of the previous time step are fed as the input to the network to assist in making predictions at the current time step. The cycles in a RNN allow it to store and update the context information about the past inputs in its internal state for an amount of time that is not fixed a priori, but rather depends on its weights and on the input data [87]. Therefore, RNNs are able to exploit a dynamically changing contextual window over the input sequence rather than a static one as in the fixed-sized window used with a conventional deep FNN. The long short-term memory (LSTM) network [10] is a kind of RNN specially designed for capturing *long-term* temporal context information. It overcomes the diminishing gradient problem that comes along the RNN training with a special gating mechanism to control the information to be added or removed to the internal cell state. To also exploit future context information to assist in making current prediction, bidirectional LSTM (BLSTM) networks [14] are introduced to process the input sequence in both directions with two separate hidden layers which are then fed forward together to the same output layer.

Therefore, we further propose the *non-uniform boosted minimum classification error (BMCE) training of a deep BLSTM* acoustic model for keyword spotting in spontaneous conversational speech. We define the empirical error cost for non-uniform MCE and derive the backpropagation

error for the BLSTM. The BLSTM is optimized using backpropagation through time and stochastic gradient descent. With the non-uniform BMCE trained BLSTM acoustic model, the LVCSR decoder is able to generate word transcription with significantly reduced recognition errors on the keywords. To further improve the performance, we boost the likelihood of the hypothesized word sequences in proportion to their phone error rate, which is equivalent to generating more confusable data. Experiments are performed on Switchboard-1 Release 2 dataset, which is a large-scale spontaneous conversational telephone speech (CTS) dataset. The proposed method achieves 5.49% and 7.37% absolute figure-of-merit (FOM) improvements respectively over the BLSTM and FNN baseline systems trained with the cross-entropy criterion for the keyword spotting task on "Credit Card Use" topic of Switchboard-1 Release 2 dataset.

In Section 3.2, we discuss how the non-uniform BMCE criterion is embedded in the DNN training for keyword spotting. In Section 3.3, we show how the non-uniform BMCE is implemented in the WFST framework. In Section 3.4, experimental results on Switchboard dataset are shown and discussed. We draw our conclusion in Section 3.5.

3.2 Non-Uniform BMCE Training of DNN Acoustic Models for Keyword Spotting

Conventionally, DNNs (deep FNNs, deep RNNs and deep BLSTMs are all special types of DNNs) are trained to model the distribution of the senones based on a cross-entropy criterion in LVCSR tasks. A senone-level alignment on the training set is used as the labels for training the DNN. However, the DNNs trained through distribution estimation do not necessarily lead to the minimization of the recognition error rate. In [66], maximum mutual information (MMI) [56, 57], minimum phone error (MPE) [58, 59], state-level minimum Bayes risk (sMBR) [60, 61, 62] and boosted MMI [65] are used as the objective for DNN training. Although these discriminative traing methods are able to improve the performance over the traditional cross-entropy based methods, they do not directly minimize an objective function which is related to the recognition error rate. To circumvent this problem, MCE was proposed to directly minimize the empirical error rate and is widely used in GMM-based LVCSR systems.

Keyword spotting can be formulated as an LVCSR task in which some recognition units (i.e., keywords) are more significant than others. More specifically, the LVCSR designed for keyword

spotting should be able to generate a decoded word sequence in which keywords have fewer recognition errors than the normal LVCSR system. To satisfy this requirement, we introduce the non-uniform MCE objective for the training of DNNs in the LVCSR task. Instead of minimizing the empirical error rate for conventional MCE, the non-uniform MCE training of DNN is aimed at minimizing the empirical error cost. This can be realized by embedding the non-uniform error cost function into the MCE objective on the frame level to emphasize both the miss detection errors and the false alarm errors on the keywords. Strictly speaking, the error cost function should be individually assigned to each pair of words in the vocabulary to take care of all kinds of recognition errors [88]. Our formulation is a simplified version of the general non-uniform MCE for fast and easy implementation. Introducing nonuniform error cost at the frame level is justified based on the general assumption that word-level errors are proportional to their frame-level errors and minimizing frame-level non-uniform error costs will accomplish similar results as minimizing word-level non-uniform costs.

A two-stage training approach based on the standard error backpropagation procedure is applied to optimize the non-uniform MCE objective. In the first stage, the gradients of the non-uniform MCE objective with respect to the activations at the output layer are calculated and then backpropagated to derive the gradients for all the parameters of the DNN in the second stage. We will derive this important gradient below.

Assume that the training data is given by training utterances $r = \{1, \ldots, R\}$. The sequence $X_r = \{x_{r1}, \ldots, x_{rT_r}\}$ is observations for utterance r. W_r is the word sequence in the reference (label transcription) for utterance r. W is a word sequence in the hypothesis set encapsulated in the decoded speech unit lattice for utterance r. $S_W = \{s_{W1}, \ldots, s_{WT}\}$ is the senone sequence corresponding to W, where s_{Wt} is the senone which frame x_{rt} is aligned with.

The output of the DNN for senone s is the posterior probability $p(s|x_{rt})$ obtained by a softmax function.

$$p(s|x_{rt}) = \frac{\exp[a_{rt}(s)]}{\Sigma_{s'} \exp[a_{rt}(s')]} \tag{3.1}$$

where $a_{rt}(s)$ is the activation for senone s at the output layer. The pseudo log-likelihood of obser-

vation x_{rt} given senone s is

$$\log p(x_{rt}|s) = \log p(s|x_{rt}) - \log p(s) + \log p(x_{rt}) \tag{3.2}$$

where $p(s)$ is the prior probability of senone s estimated from the training set and $p(x_{rt})$ is the probability of observation x_{rt} which is independent of the word sequence and can be ignored.

The frame-level discriminative function for W and misclassification measure is given by

$$g(x_{rt}, s_{Wt}; \Lambda) = \log[p(x_{rt}|s_{Wt})^{\kappa} p(s_{Wt})] \tag{3.3}$$

where $p(x_{rt}|s_{Wt})$ and $p(s_{Wt})$ denote the acoustic and language models respectively, κ is the acoustic model scaling factor and Λ is a set of model parameters.

$$d(x_{rt}; \Lambda) = -g(x_{rt}, s_{W_r t}; \Lambda) + \log \left\{ \frac{1}{N-1} \sum_{W \neq W_r} \exp[g(x_{rt}, s_{Wt}; \Lambda)\eta] \right\}^{\frac{1}{\eta}} \tag{3.4}$$

where N is the total number of hypothesized word sequences. By varying the positive number η, the significance of the competing classes can be adjusted.

By embedding the misclassification measure Eq. (3.4) into a sigmoid function for smoothing, the objective function of the non-uniform MCE training of DNN is given by

$$\mathcal{L}_{\text{NUMCE}}(\Lambda) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \epsilon_r(t) l(d(x_{rt}; \Lambda)) \tag{3.5}$$

where $\epsilon_r(t)$ is the error cost function at the frame level, $l(\cdot)$ is the sigmoid which takes the form

$$l(d) = \frac{1}{1 + \exp(-\alpha d + \beta)} \tag{3.6}$$

The slope of the sigmoid curve can be adjusted by α and β is normally set to 0. The objective function in Eq. (3.5) is essentially a smoothed approximation of the *empirical error cost*. Note that when the error cost function is fixed to 1 for all t (i.e., $\epsilon_r(t) = 1$), Eq. (3.5) degrades to the objective

function of MCE, which is a smoothed approximation of the *empirical error rate* on the training set.

The derivative of Eq. (3.5) with respect to the activation $a_{rt}(s)$ at the output layer is

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{NUMCE}}(\Lambda)}{\partial a_{rt}(s)} &= \sum_{q} \frac{\partial \mathcal{L}_{\text{NUMCE}}(\Lambda)}{\partial \log p(x_{rt}|q)} \frac{\partial \log p(x_{rt}|q)}{\partial a_{rt}(s)} \\ &= \alpha \epsilon_r(t) l\left(d(x_{rt};\Lambda)\right)\left[1 - l\left(d(x_{rt};\Lambda)\right)\right] \kappa \left[\delta_{s_{W_r t}:s} - \gamma_{rt}^{W \neq W_r}(s)\right] \end{aligned} \tag{3.7}$$

where $\gamma_{rt}^{W \neq W_r}(s)$ is the posterior of being in senone s at time t, computed over the denominator lattice of the utterance r, and the lattice of utterance r excluding the path corresponding to the word sequence W_r, $\log p(x_{rt}|q)$ is the log-likelihood of x_{rt} given senone q, and $\delta_{s_{Wt}:s}$ is the Kronecker delta function defined as

$$\delta_{s_{Wt}:s} = \begin{cases} 1, & s_{Wt} = s \\ 0, & s_{Wt} \neq s \end{cases} \tag{3.8}$$

For easy implementation, $d(X_{rt};\Lambda)$ is used as an approximation of $d(x_{rt};\Lambda)$. Eq. (3.7) is the error to be backpropagated to derive the gradients for all the parameters of DNN.

To minimize the recognition errors on the keywords, the error cost function $\epsilon_r(t)$ should be designed in such a way that all the recognition error cost associated with the keywords are emphasized. More specifically, as in Eq. (3.9), the initial $\epsilon_r(t)$ for the frames labeled as keywords in the label transcription (denoted by K_1) should be greater than 1 to reduce the miss detection errors. Also the initial $\epsilon_r(t)$ for the frames aligned with keywords on the hypothesized word sequences (denoted by K_2) other than the label transcription should be greater than 1 to prevent the false alarm errors. The $\epsilon_r(t)$ for the frames aligned with non-keywords in all the word sequences in the decoded speech lattice for utterance r should be 1.

$$\epsilon_r(t) = \begin{cases} K_1, & t \in \{t | W_r(t) \text{ is a keyword}\} \\ K_2, & t \in \{t | W(t) \text{ is a keyword}, W \neq W_r\} \\ 1, & \text{otherwise} \end{cases} \tag{3.9}$$

where $W_r(t)$ is the word which x_{rt} is aligned with in the label transcription and $W(t)$ is the word

which x_{rt} is aligned with in the hypothesis word sequences and $K_1 > 1, K_2 > 1$. The error cost function can be adjusted adaptively through iterations using a AdaBoost-like scheme as is proposed in [89]. We multiply $\epsilon_r(t)$ with a decay factor β if a frame x_{rt} is correctly classified at the current training iteration.

To achieve a better performance for keyword spotting, we boost the likelihood of the hypothesized word sequences that have a higher phone error rate relative to the label transcription, which is equivalent to generating more data from the more confusable hypothesized word sequences. For non-uniform boosted MCE, the misclassification measure can be re-written as

$$d(x_{rt};\Lambda) = -g(x_{rt}, s_{W_r t};\Lambda) + \log\left\{\frac{1}{N-1}\sum_{W\neq W_r}\exp\{g(x_{rt}, s_{Wt};\Lambda) - bA(p_{Wt}, p_{W_r t})]\eta\}\right\}^{\frac{1}{\eta}} \tag{3.10}$$

where b is the boosting factor and $A(p_{Wt}, p_{W_r t})$ is the frame-level raw phone accuracy of a sentence W given the label transcription W_r, i.e.,

$$A(p_{Wt}, p_{W_r t}) = \begin{cases} 1, & p_{Wt} = p_{W_r t} \\ 0, & p_{Wt} \neq p_{W_r t} \end{cases} \tag{3.11}$$

where p_{Wt} is the raw phone which frame x_{rt} is aligned with and $P_W = \{p_{W1}, \ldots, p_{WT}\}$ is the phone sequence corresponding to word sequence W.

The objective function and backpropagation error of non-uniform BMCE can be derived correspondingly.

3.3 Implementation of Non-Uniform BMCE in the WFST Framework

The non-uniform MCE is implemented within the WFST framework. As is mentioned in [90], a decoded lattice of an utterance is generated by a beam pruning on the full search graph which is a composition of the WFST U and the HCLG graph. U, H, C, L and G denote the acoustic score of the utterance, the HMM structure, the phonetic context-dependency, the lexicon and the

grammar, respectively. The decoded lattice is a compact representation of the hypothesis space for the utterance. The lattice is converted to a compact version for higher efficiency.

For an utterance r, the competing hypothesis for non-uniform MCE training has to exclude the label transcription W_r as is shown in Eq. (3.4). This is accomplished by taking the *difference* operation of WFST. Assuming that $L_r(W)$ is the compact lattice for utterance r and $WFST(W_r)$ is the compiled WFST for W_r, the lattice representing the competing hypothesis in non-uniform MCE training is given by

$$L_r^{\text{NUMCE}} = L_r(W) - WFST(W_r) \tag{3.12}$$

The posterior $\gamma_{rt}^{W \neq W_r}(s)$ in Eq. (3.7) can be obtained by performing the forward-backward procedure on L_r^{NUMCE}.

In the WFST framework, non-uniform BMCE training of DNN can be easily implemented based on non-uniform MCE. The extra computation involved is to subtract b times the frame-level raw phone accuracy $A(p_{Wt}, p_{W_r})$ from the scaled acoustic log-likelihood on each arc at time t in the lattice while performing forward-backward on L_r^{NUMCE}. This can be viewed as a modification of the contribution from language model on each arc.

3.4 Experiments

3.4.1 Experiment on Switchboard

3.4.1.1 Dataset Description

We evaluate the performance of the proposed framework on a large-scale CTS task, i.e., the 300 hours Switchboard-1 Release 2 (LDC97S62). It consists of 2348 two-sided telephone conversations from 543 speakers (302 males and 241 females) in the United States. One topic is assigned to each of the conversation between two callers and about 70 topics in total are provided in the corpus.

For the keyword spotting task, the conversations on the topic of "Credit Card Use" (including 5649 utterances) are used as the test set and the rest of the Switchboard corpus form a training set with about 300 hours of speech. 18 keywords are selected for the spotting evaluation, which are

BANK, CARD, CASH, CHARGE, CHECK, MONTH, ACCOUNT, BALANCE, CREDIT, DOLLAR, HUNDRED, LIMIT, MONEY, PERCENT, TWENTY, VISA, DISCOVER, INTEREST. For both tasks, the Mississippi State transcripts and the 30K-word lexicon released with those transcripts are used. The lexicon contains pronunciations for all words and word fragments in the training data.

3.4.1.2 Baseline System

The baseline ASR system is built with Kaldi Speech Recognition Toolkit [91]. The GMM-HMMs are trained with the 300 hour training data using maximum-likelihood (ML) criterion. Each cross-word triphone is modeled by a 3-state left-to-right GMM-HMM (a 5-state HMM for silence). First, 9 frames (4 on each side of the current frame) of 13-dimensional Mel-frequency cepstral coefficient (MFCCs) are spliced together and projected down to 40 dimensions using linear discriminant analysis (LDA). Then a single semi-tied covariance (STC) transform is performed on the features obtained by LDA. Then speaker adaptive training is performed using a single feature-space maximum likelihood linear regression (FMLLR) transform estimated for each speaker. The resulting feature after FMLLR is called LDA+STC+FMLLR feature and is used for training the GMM-HMMs. The trigram language model (LM) is trained on 3M words of the training transcripts. We generate the forced alignment of the training data against the transcription using the GMM-HMM system. As shown in Table 3.1, the GMM-HMM system trained with ML criterion achieves 74.76% FOM for the keyword spotting task.

For training the FNN and BLSTM, the 36 dimensional log Mel filterbank features are extracted and then concatenated with 3 dimensional pitch features (consisting of probability of voicing, log pitch and delta log pitch) [92] to form a 39 dimensional "log Mel filterbank + pitch" feature.

For the FNN-HMM baseline, we first pre-train a deep belief network (DBN) containing stacked restricted Boltzmann machines that are trained generatively in a layerwise fashion. The DBN is then fine-tuned to train a FNN with cross-entropy objective using stochastic gradient descent (initial learning rate 0.008). The input to the FNN is an 11 frame (5 frames on each side of the current frame) context window of the 39 dimensional "log Mel filterbank + pitch" features globally normalized to have zero mean and unit variance. The resulting baseline FNNs has 7 layers (including 6 hidden layers), where each hidden layer has 2048 neurons, and the output layer has 8861 units.

The FNN is randomly initialized and then trained to minimize the cross-entropy (CE) criterion using senone-level forced alignment generated by the GMM-HMM system as the target. As in Table 3.1, the FNN-HMM baseline system trained with criterion achieves 78.06% FOM for the keyword spotting task.

To build the BLSTM-HMM baseline system, we stack 4 BLSTM hidden layers together and add a softmax output layer on the top to represent the 8861 senones posteriors. Each forward or backward hidden layer has 512 hidden units and is connected to a 256 dimensional recurrent projection layer. The forward and backward projection layers are concatenated together (to form a 512 dimensional vector) and fed as the input of the next BLSTM hidden layer. After appending delta and delta-delta coefficients to the 39 dimensional "log Mel filterbank + pitch" features, we use the 117 dimensional features with globally normalized zero mean and unit variance as the input to the BLSTM. The BLSTM is randomly initialized and then trained (initial learning rate 0.00002) to minimize the CE criterion using senone-level forced alignment generated by the GMM-HMM system as the target. The BLSTM-HMM baseline system trained with criterion achieves 80.93% FOM for the keyword spotting task as shown in Table 3.2,

3.4.1.3 Results of FNN Acoustic Models for Keyword Spotting

The FNN in the baseline system is then trained with the non-uniform BMCE criterion for keyword spotting. We generate the forced alignment and denominator lattice of the training data using the baseline FNN, compute posterior $\gamma_{rt}^{W \neq W_r}(s)$ from the difference lattice L_r^{NUBMCE}, impose error cost function $\epsilon_r(t)$ on the frames aligned with keywords and compute errors in Eq. (3.7) for back-propagation through time. For comparison, we also discriminatively train baseline FNN with MMI, sMBR and BMCE criteria.

In Table 3.1, we show the FOM results of deep FNN acoustic models with respect to different initial error costs K_1, K_2 and decay factors β. The system achieves the highest FOM 80.92% when $K_1 = K_2 = 12$ and $\beta = 0.3$, which is 2.86% and 1.55% absolute improvements over the baseline FNN and sMBR trained FNN. The best FOM is achieved when the learning rate is 0.0003, the slope of sigmoid α is 0.002 and the boosting factor is set at 0.07. We also observe that the FOM first increases as K_1 and K_2 grow and then gradually decreases when K_l and K_2 are larger than 14. The

Table 3.1: The FOM results of the FNN-HMM and GMM-HMM systems trained with different objectives for keyword spotting on Credit Card Use subset of Switchboard-1 Release 2.

System	K_1	K_2	β	FOM (%)
GMM ML	1	1	-	74.76
FNN CE	1	1	-	78.06
FNN MMI	1	1	-	79.05
FNN sMBR	1	1	-	79.37
FNN MCE	1	1	-	79.24
FNN BMCE	1	1	-	79.48
FNN Non-Uniform BMCE	6.0	6.0	0.1	80.29
	6.0	6.0	0.5	80.24
	8.0	8.0	0.1	80.37
	8.0	8.0	0.5	80.44
	10.0	10.0	0.1	80.57
	10.0	10.0	0.5	80.50
	12.0	12.0	0.1	80.38
	12.0	12.0	0.5	80.43
	14.0	14.0	0.1	80.65
	14.0	14.0	0.5	80.54
	16.0	16.0	0.1	**80.92**
	16.0	16.0	0.5	80.45
	18.0	18.0	0.1	80.86
	18.0	18.0	0.5	80.82
	20.0	20.0	0.1	80.91
	20.0	20.0	0.5	80.39
	22.0	22.0	0.1	80.31
	22.0	22.0	0.5	80.36

FOM increases or decreases more rapidly when the decay factor is smaller.

We plot the ROC curves for the FNNs trained with cross-entropy, BMCE and non-uniform MCE criteria in Fig. 3.2. The non-uniform MCE achieves consistent improvement over other objectives.

3.4.1.4 Results of BLSTM Acoustic Models for Keyword Spotting

In Table 3.2, we show the FOM results of deep BLSTM acoustic models with respect to different initial error costs K_1, K_2 and decay factors β. The system achieves the highest FOM 85.42% when $K_1 = K_2 = 10$ and $\beta = 0.3$, which is 4.49% and 1.23% absolute improvements over the baseline BLSTM and sMBR trained BLSTM. The best FOM is achieved when the learning rate is 0.00001, the slope of sigmoid α is 0.002 and the boosting factor is set at 0.07. We also observe that the FOM

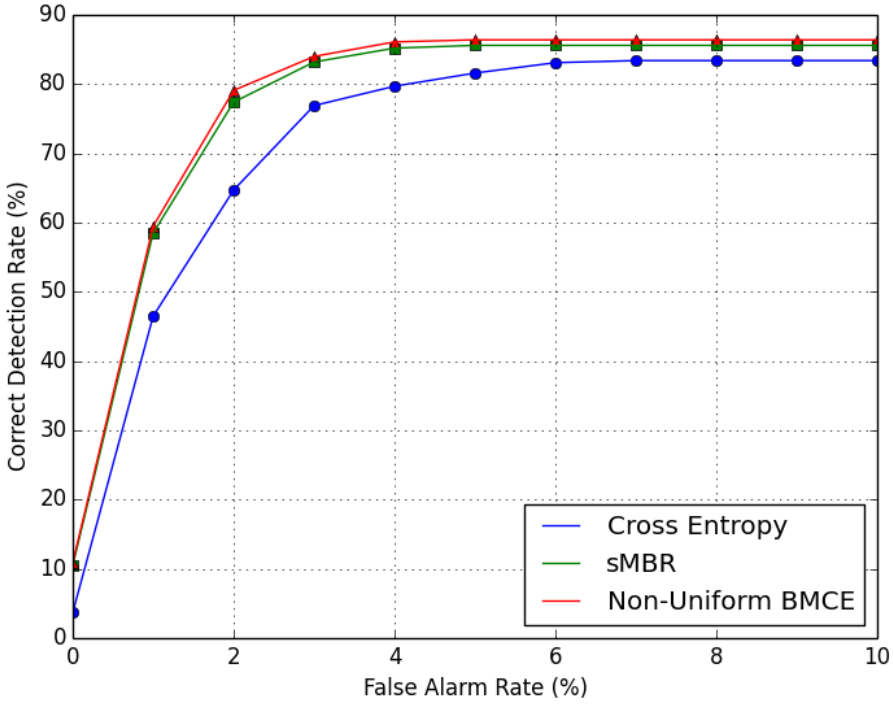


Figure 3.1: ROC curves of keyword spotting for baseline cross-entropy trained, BMCE trained and best performing non-uniform BMCE FNN-HMM system on the development set of HKUST dataset.

first increases as K_1 and K_2 grow and then gradually decreases when K_l and K_2 are larger than 10. The FOM increases or decreases more rapidly when the decay factor is smaller.

We plot the ROC curves for the BLSTMs trained with cross-entropy, BMCE and non-uniform MCE criteria in Fig. 3.2. The non-uniform MCE achieves consistent improvement over other objectives.

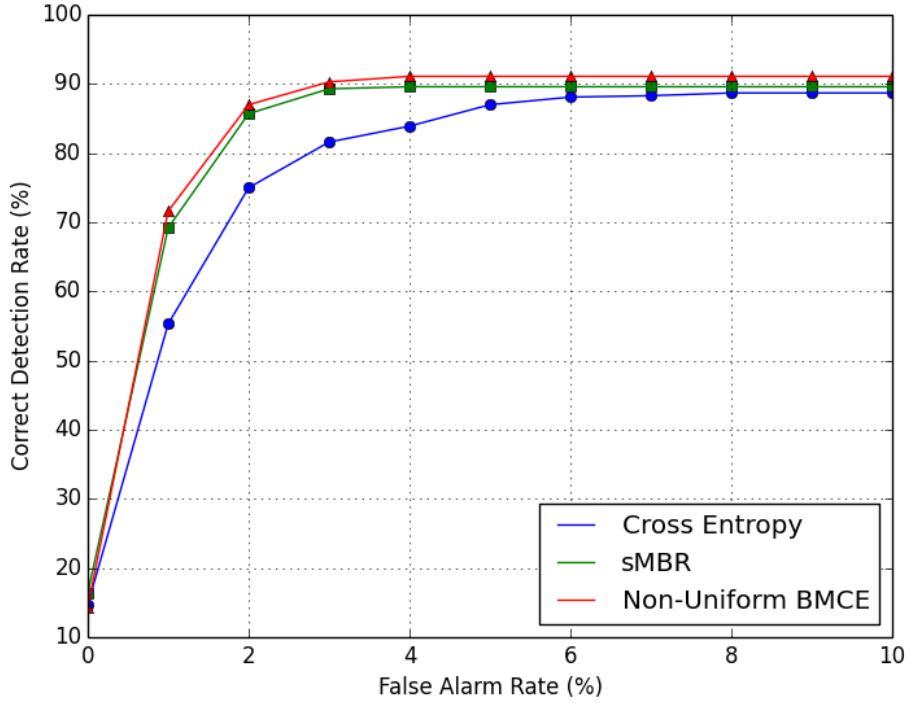


Figure 3.2: ROC curves of keyword spotting for baseline cross-entropy trained, BMCE trained and best performing non-uniform BMCE BLSTM-HMM system on the development set of HKUST dataset.

By comparing Table 3.1 with Table 3.2, we observe that FNN and BLSTM have the same trend of FOM variation with respect to the initial error cost function and decay factor. The non-uniform BMCE trained BLSTM achieves 7.37% and 4.88% absolute FOM gains over cross-entropy trained FNN and non-uniform BMCE trained FNN. Under other uniform error discriminative training criteria, the BLSTM in general leads to about 4.0%-4.5% absolute FOM improvements over the FNN, which are much larger than the 2.48% absolute FOM gain FNN achieves under the cross-entropy criterion. The large FOM improvement of BLSTM over FNN verifies its strong capability of mod-

Table 3.2: The FOM results of the BLSTM-HMM systems trained with different objectives for keyword spotting on Credit Card Use subset of Switchboard-1 Release 2.

System	K_1	K_2	β	FOM (%)
BLSTM CE (baseline)	1	1	-	80.93
BLSTM MMI	1	1	-	83.25
BLSTM sMBR	1	1	-	84.19
BLSTM BMCE	1	1	-	84.21
BLSTM Non-Uniform BMCE	6	6	0.3	84.89
	6	6	0.5	84.87
	6	6	0.7	84.38
	7	7	0.3	84.69
	7	7	0.5	85.01
	7	7	0.7	84.83
	8	8	0.3	85.02
	8	8	0.5	85.15
	8	8	0.7	85.11
	9	9	0.3	84.98
	9	9	0.5	84.63
	9	9	0.7	84.92
	10	10	0.3	**85.42**
	10	10	0.5	85.08
	10	10	0.7	84.91
	11	11	0.3	85.27
	11	11	0.5	84.96
	11	11	0.7	84.91
	12	12	0.3	85.01
	12	12	0.5	85.05
	12	12	0.7	85.01
	13	13	0.3	84.99
	13	13	0.5	85.01
	13	13	0.7	84.60
	14	14	0.3	85.02
	14	14	0.5	84.55
	14	14	0.7	85.01

Table 3.3: The WER (%) results of the GMM-HMM, FNN-HMM and BLSTM-HMM systems trained with different objectives evaluated on the development set of the HKUST dataset. The WERs for non-unifrom BMCE in the table correspond to the setups that achieve the best FOMs in the keyword spotting experiments (see Table 3.1 and Table 3.2).

System	WER (%)
GMM ML	27.65
FNN CE	19.97
FNN MMI	18.45
FNN sMBR	18.39
FNN BMCE	18.34
FNN Non-Uniform BMCE	18.51
BLSTM CE	18.59
BLSTM MMI	16.31
BLSTM sMBR	16.29
BLSTM BMCE	16.20
BLSTM Non-Uniform BMCE	16.47

eling long-term dependencies and high correlations between speech frames that spans over long dynamic time intervals.

3.4.1.5 Results of DNN Acoustic Models for LVCSR

We also show the WER performance for the LVCSR task on the test data in Table 3.3. The WERs for non-unifrom BMCE in the table correspond to the setups that achieve the best FOMs in the keyword spotting experiments (see Table 3.1 and Table 3.2). Although non-uniform BMCE achieves 1.43% and 2.21% absolute improvements over the CE objective, it is worse than the other discriminative objectives by 0.05%-0.15% and 0.1%-0.2% absolutely on FNN and BLSTM acoustic models respectively. The fact that the non-uniform MCE improves the FOM of keyword spotting but simultaneously degrades the WER of LVCSR justifies that the goal of non-uniform MCE training is to minimize the recognition errors on only the keywords instead of all possible words.

3.4.2 Experiments on HKUST Dataset

3.4.2.1 Dataset Description

HKUST Dataset (LDC2005S15) consists of 150 hours of Mandarin Chinese conversational telephone speech collected by the Hong Kong University of Science and Technology (HKUST) from

speakers in several cities across mainland China. It contains 873 and 24 call conversations for the training and development sets respectively.

Since there is no lexicon provided with the corpus and it contains both Chinese and English words (it is highly likely English words occurring in spontaneous mandarin speech), below we briefly describe how we prepare the bilingual lexicon. To deal with the occasional occurances of English words in Manderin speech, we construct the bilingual lexicon as follows. The in-vocabulary Chinese words are mapped to their pronunciations (Pinyin) using the dictionary CEDICT [93]. For out-of-vocabulary (OOV) words, we construct the pronunciations by concatenating the Pinyin of all the characters that form the word. All possible pronunciations for each word are enumerated. Then we map all Pinyin initials and finals (with tones) to Arpabet phonemes which are widely used in English via IPA rules similar to [94]. For the English words, the CMU dictionary [95] is used to map in-vocabulary words to their pronunciations. For the OOV words, Sequitur G2P [96] tool is used to map the graphemes to phonemes using pre-trained models. Further, each phoneme that corresponds to a Pinyin final is assigned with 6 different tones, eg., AO (mainly used for English words), AO1, AO2, AO3, AO4, AO5. (Note that the Pinyin initials are toneless.) The phoneme with the different tones share the same root in the decision tree while extra tonal questions are made for them.

In keywords spotting experiments, we use the development set as the test set and select 20 Chinese keywords: (like), (China), (university), (life), (friend), (country), (football), (Huangshan), (exercise), (basketball), (sing), (job), (major), (sports), (televisions), (sports), (study), (problem), (Taiwan), (student).

3.4.2.2 Baseline

The baseline system is built in the same way as is described in Section 3.4.1.2 except for the following changes. The Chinese words are segmented by an open-source tool mmseg [97] a tri-gram language model is then trained on all transcriptions from training set.

In the baseline FNN-HMM system, the FNN has 6 hidden layers, where each hidden layer has 2048 neurons, and the output layer has 2878 units representing senone posteriors. In the baseline BLSTN-HMM system, the BLSTM has 5 hidden layers with 256 hidden units in each layer. The

output layer consists of 2878 units predicting senone posteriors. Each forward and backward hidden layer is projected to 128 units through a projection layer. The forward and backward projection layers are concatenated to form a 256-dimensional vector before being fed into the next hidden layer. The BLSTM is trained with CE criterion at a learning rate of 0.00003. The senone-level forced alignment generated by the GMM-HMM system is used as the target to train both FNN-HMM and BLSTM-HMM systems. The baseline GMM, FNN and BLSTM systems achieve 60.13% 73.29% and 79.49% FOMs for keyword spotting on HKUST as shown in Table 3.4.

3.4.2.3 Results of FNN Acoustic Models for Keyword Spotting

The keyword spotting experiments are performed in the same way as described in Section 3.4.1.3 and the results are shown in Table 3.5 and Table 3.4.

In Table 3.4, we show the FOM results of FNN acoustic models with respect to different initial error costs K_1, K_2 and decay factors β. The system achieves the highest FOM 80.55% when $K_1 = K_2 = 2$ and $\beta = 0.5$, which is 7.26% and 3.16% absolute improvements over the baseline BLSTM and sMBR trained BLSTM. The best FOM is achieved when the learning rate is 0.0001, the slope of sigmoid α is 0.002 and the boosting factor is set at 0.07. We also observe that the FOM first increases as K_1 and K_2 grow and then gradually decreases when K_l and K_2 are larger than 11. The FOM increases or decreases more rapidly when the decay factor is smaller.

We plot the ROC curves for the FNNs trained with cross-entropy, BMCE and non-uniform MCE criteria in Fig. 3.3. The non-uniform MCE achieves consistent improvement over other objectives.

3.4.2.4 Results of BLSTM Acoustic Models for Keyword Spotting

In Table 3.5, we show the FOM results with respect to different initial error costs K_1, K_2 and decay factors β. The system achieves the highest FOM 86.39% when $K_1 = K_2 = 10$ and $\beta = 0.5$, which is 6.90% and 2.17% absolute improvements over the baseline BLSTM and sMBR trained BLSTM. The best FOM is achieved when the learning rate is 0.00001, the slope of sigmoid α is 0.002 and the boosting factor is set at 0.07. We also observe that the FOM first increases as K_1 and K_2 grow and then gradually decreases when K_l and K_2 are larger than 11. The FOM increases or decreases more rapidly when the decay factor is smaller. We plot the ROC curves for the BLSTMs trained with

Table 3.4: The FOM results of the FNN-HMM and GMM-HMM systems trained with different objectives for keyword spotting on the development set of HKUST dataset.

System	K_1	K_2	β	FOM (%)
GMM ML	1	1	-	60.13
FNN CE	1	1	-	73.29
FNN MMI	1	1	-	77.39
FNN sMBR	1	1	-	77.51
FNN BMCE	1	1	-	77.27
FNN Non-Uniform BMCE	2.0	2.0	0.3	79.61
	2.0	2.0	0.5	**80.55**
	2.0	2.0	0.7	80.31
	3.0	3.0	0.3	79.56
	3.0	3.0	0.5	79.28
	3.0	3.0	0.7	79.47
	4.0	4.0	0.3	78.51
	4.0	4.0	0.5	78.46
	4.0	4.0	0.7	78.62
	5.0	5.0	0.3	78.83
	5.0	5.0	0.5	78.45
	5.0	5.0	0.7	78.45
	6.0	6.0	0.3	78.49
	6.0	6.0	0.5	78.28
	6.0	6.0	0.7	78.74
	7.0	7.0	0.3	77.94
	7.0	7.0	0.5	78.26
	7.0	7.0	0.7	78.55

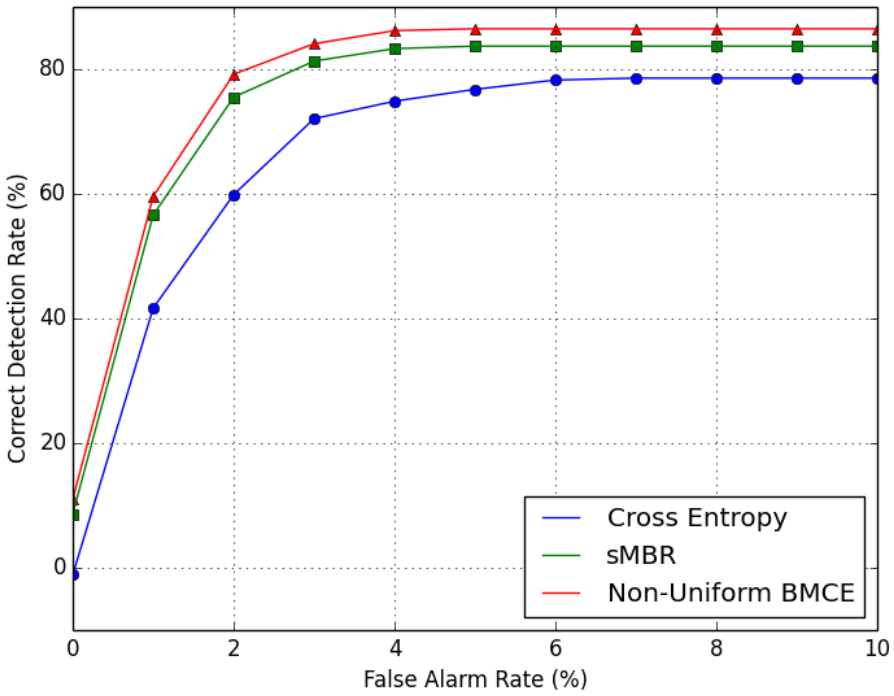


Figure 3.3: ROC curves of keyword spotting for baseline cross-entropy trained, BMCE trained and best performing non-uniform BMCE FNN-HMM system on the development set of HKUST dataset.

cross-entropy, BMCE and non-uniform MCE criteria in Fig. 3.4. The non-uniform MCE achieves consistent improvement over other objectives.

We plot the ROC curves for the BLSTMs trained with cross-entropy, BMCE and non-uniform MCE criteria in Fig. 3.4. The non-uniform MCE achieves consistent improvement over other objectives.

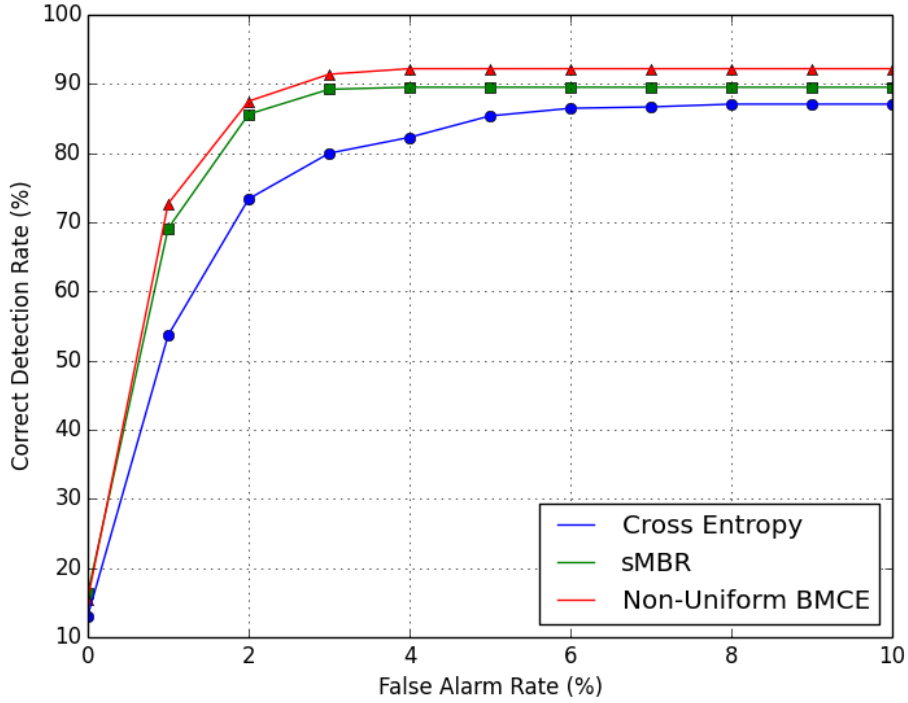


Figure 3.4: ROC curves of keyword spotting for baseline cross-entropy trained, BMCE trained and best performing non-uniform BMCE BLSTM-HMM system on the development set of HKUST dataset.

By comparing Table 3.4 and Table 3.4, we observe that FNN and BLSTM have the same trend of FOM variation with respect to the initial error cost function and decay factor. The non-uniform BMCE trained BLSTM achieves 7.37% and 4.88% absolute FOM gains over cross-entropy trained FNN and non-uniform BMCE trained DNN. Under other uniform error discriminative training criteria, the BLSTM in general leads to about 4.0%-4.5% absolute FOM improvements over the FNN, which are much larger than the 2.3% absolute FOM gain BLSTM achieves under cross-entropy criterion. The large FOM improvement of BLSTM over FNN verifies its strong capability of modeling

Table 3.5: The FOM results of the BLSTM-HMM systems trained with different objectives for keyword spotting on development set of the HKUST dataset.

System	K_1	K_2	β	FOM (%)
BLSTM CE (baseline)	1	1	-	79.49
BLSTM MMI	1	1	-	84.17
BLSTM sMBR	1	1	-	84.22
BLSTM BMCE	1	1	-	84.56
BLSTM Non-Uniform BMCE	7	7	0.3	85.42
	7	7	0.5	85.23
	7	7	0.7	86.16
	8	8	0.3	85.02
	8	8	0.5	85.34
	8	8	0.7	85.87
	9	9	0.3	85.54
	9	9	0.5	86.13
	9	9	0.7	85.28
	10	10	0.3	85.44
	10	10	0.5	**86.39**
	10	10	0.7	86.27
	11	11	0.3	85.77
	11	11	0.5	85.73
	11	11	0.7	85.59
	12	12	0.3	85.28
	12	12	0.5	85.31
	12	12	0.7	85.80
	13	13	0.3	85.44
	13	13	0.5	85.91
	13	13	0.7	85.57
	14	14	0.3	85.21
	14	14	0.5	85.93
	14	14	0.7	85.91
	15	15	0.3	85.58
	15	15	0.5	85.58
	15	15	0.7	85.58

Table 3.6: The CER (%) results of the GMM-HMM, FNN-HMM and BLSTM-HMM systems trained with different objectives evaluated on the development set of the HKUST dataset. The CERs for non-unifrom BMCE in the table correspond to the setups that achieve the best FOMs in the keyword spotting experiments (see Table 3.4 and Table 3.5)

System	CER (%)
GMM SAT	49.60
FNN CE	39.60
FNN MMI	36.97
FNN sMBR	36.90
FNN BMCE	36.96
FNN Non-Uniform BMCE	37.55
BLSTM CE	35.44
BLSTM MMI	32.68
BLSTM sMBR	32.65
BLSTM BMCE	32.59
BLSTM Non-Uniform BMCE	32.80

long-term dependencies and high correlations between speech frames that spans over long dynamic time intervals.

3.4.2.5 Results of DNN Acoustic Models for LVCSR

We also show the CER performance for the LVCSR task on the test data in Table 3.6. The CERs for non-unifrom BMCE in the table correspond to the setups that achieve the best FOMs in the keyword spotting experiments (see Table 3.4 and Table 3.5). The CERs Although non-uniform BMCE achieves 2.05% and 2.64% absolute CER improvements over the CE objective, it is worse than other discriminative objectives by 0.5%-0.6% and 0.1%-0.3% absolutely on FNN and BLSTM acoustic models respectively. The fact that the non-uniform MCE improves the FOM of keyword spotting but simultaneously degrades the CER of LVCSR justifies that the goal of non-uniform MCE training is to minimize the recognition errors on only the keywords instead of all possible words.

3.5 Conclusions

In this chapter, we formulate the keyword spotting problem as a non-uniform error ASR problem and show that DNNs can be dicriminatively trained using non-uniform BMCE criterion which weighs the errors on keywords much more significantly than those on non-keywords in an ASR task. By

using FNN-HMM acoustic model, we are able to model the multi-frame distributions, which conventional systems find difficult to accomplish. The further integration with BLSTM-HMM system enables the capturing of long-term dependencies within the variable-duration dynamic speech signal instead of a fixed-size window using a FNN-HMM. The proposed system is implemented within a WFST framework.

Experiments are conducted on Switchboard-1 Release 2 and HKUST datasets. The non-uniform MCE training of FNN achieves 2.48% and 7.26% FOM improvements over the cross entropy baseline system on Switchboard and HKUST datasets respectively. The non-uniform MCE training of BLSTM achieves 4.49% and 7.37% FOM improvements over the cross entropy baseline system on Switchboard and HKUST datasets respectively.

CHAPTER 4

MINIMUM SEMANTIC ERROR COST TRAINING FOR TOPIC SPOTTING ON CONVERSATIONAL SPEECH

4.1 Introduction

Topic spotting on spontaneous conversational speech is an essential technique for spoken-dialog systems. The response of a spoken-dialog system is often guided by the topic category of the speaker's utterance. The topic spotting is aimed at classifying an utterance into one of a pre-defined set of topics.

Many methods have been proposed for topic spotting on conversational speech. In [98, 99, 100, 101], a set of keywords are first selected according to their contributions for the topic discrimination and the topic spotting is then conducted by scoring the *one-best* transcription generated by an LVCSR system based on the selected keywords. A similar idea is applied to the famous AT&T HMIHY call-routing task [99, 100], in which salient words or phrases are acquired, recognized and searched in fluent speech by an ASR. The call-type of an utterance is classified based on these salient words. In [101], a BOOSTEXTER algorithm is used to learn the ASR language model and a mapping from the ASR transcriptions into weightings over topics. In Bell Lab's natural language call routing system [102, 103, 104], the n-gram terms, queries and documents from the LVCSR output are first embedded in semantic vectors using latent semantic analysis (LSA), the calls are then routed to the desired destination according to the similarity scores computed from the query and document vectors.

However, these methods are based on the one-best transcriptions of the utterances generated by the LVCSR, which may not be accurate for a spontaneous conversational speech. Fortunately, the correct transcription is highly likely to be one of the word sequences represented by the LVCSR decoded lattice, i.e, the WFST.

To take advantage of the multiple hypothesized word sequences on the decoded lattices, Cortes et al. proposed rational kernels [105], which are a series of kernels defined on the WFSTs. The

topic classification is conducted via support vector machine (SVM) with the n-gram rational kernels which maps the WFSTs (lattices) to a high dimensional n-gram feature space and then employs an inner product for the topic identification [106, 107]. However, the n-gram rational kernel assumes an exact match of the n-grams (words or phrases) and treats the contribution of each n-gram to the topic discrimination uniformly. To overcome this problem, Weng et al. [108, 109] proposed the latent semantic rational kernels (LSRK) for the topic spotting on spontaneous speech. In the LSRK framework, the WFSTs (lattices) are mapped onto a reduced dimensional latent semantic space rather than the n-gram feature space. LSRK is generalized to incorporate external knowledge from several text analysis techniques such as WordNet [110, 16].

However, the word lattices of the utterances in [108] are generated by a Gaussian mixture model (GMM)-hidden Markov model (HMM) based LVCSR system trained with the maximum likelihood estimation (MLE). With MLE, a GMM is trained to model the distribution of the speech frames given a senone (tri-phone state), which does not necessarily lead to a minimized recognition error or a maximized topic spotting accuracy. Many discriminatvie training methods such as minimum classification error (MCE) [55, 11, 12], maximum mutual information (MMI) [56, 57], minimum word error (MWE) [59], minimum phone error (MPE) [58], state-level minimum Bayes risk (sMBR) [60, 61] and boosted MMI [65] have been proposed to further refine the acoustic model.

For the topic spotting on conversational speech, the lattices are classified based on their semantic meaning rather than their spellings or pronunciations and a high phoneme or state accuracy of a sentence does not necessarily lead to a high semantic accuracy. For instance, the sentence *"This machine is productive."* is much more semantically correct than *"This machine is inefficient."* given the reference *"This machine is efficient."*, but its phoneme or state accuracy is much lower than the latter one. For the topic spotting task, the LVCSR is expected to generate word lattices that are accurate in terms of the semantic meanings instead of the pronunciations. Therefore, we propose a minimum semantic error cost (MSEC) training of an acoustic model, in which the expected semantic error cost of all possible word sequences on the lattices is minimized given the reference. The semantic error cost between a pair of words can be estimated via LSA or recurrent neural networks (RNN) learned vector space word representations. The expected semantic error cost of the hypothesized sentences can be obtained by accumulating the word-word semantic error costs on the lattices

via the forward-backward algorithm.

In addition, the GMM-HMM acoustic model with diagonal covariance matrices in [108] are not good at handling highly correlated frames and the concatenation of neighboring frames will inevitably bring about the curse of dimensionality issue during the model training procedure. Therefore, we introduce the deep bi-directional long short-term memory (BLSTM)-HMM for acoustic modeling. The cycles in a BLSTM allows it to store and update the contextual information about the past and future inputs in its internal state for an amount of time that is not fixed a priori, but rather depends on its weights and on the input data [87]. The deep BLSTMs are able to exploit the long-term temporal contextual information within a dynamically changing window over the input speech sequence. We define the MSEC objective function and derive the backpropagation error for the BLSTM. The BLSTM is then optimized using backpropagation through time and stochastic gradient descent. With the MSEC training of the BLSTM acoustic model, the LVCSR decoder is able to generate word lattices with significantly reduced semantic error cost for the subsequent topic spotting within the LSRK framework. The BLSTM acoustic model is trained with the Switchboard-1 Release 2 dataset, which is a large-scale spontaneous conversational telephone speech (CTS) dataset. The proposed method achieves 3.5% - 4.5% absolute improvement over the BLSTM baseline trained with the cross-entropy criterion for the topic classification task on a subset of Switchboard-1 Release 2.

4.2 Minimum Semantic Error Cost Training of BLSTM Acoustic Model for Topic Spotting

With the cross-entropy criterion, the BLSTM acoustic models are trained to model the senone distribution given an input speech frame, which do not necessarily lead to a minimized recognition error rate in LVCSR tasks. An improved performance can be achieved by discriminatively training the DNN [111] and LSTM [112] acoustic models with MCE, MPE, MWE, sMBR and similar criteria.

For topic spotting on conversational speech, the speech signal is first decoded to a word lattice by an LVCSR system and the SVM with LSRK then operates on the decoded lattices to predict the topic category. The lattices are classified based on their semantic meaning rather than their spellings or pronunciations and the high phone or state accuracy of a sentence does not necessarily lead to the high semantic accuracy.

To improve the topic spotting accuracy, the LVCSR system is expected to generate word lattices that are accurate in terms of the semantic meaning rather than the pronunciation. This motivates us to devise a new objective function for discriminatively training the BLSTM acoustic model so that the LVCSR can generate word lattices with a reduced *semantic error cost*. Therefore, we propose the MSEC training of BLSTM acoustic model for topic spotting with LSRK.

We first define the word-word semantic error cost $C(i,j)$ of mistakenly recognizing one word with index i to another with index j as the *negative of the semantic similarity* between the words i and j denoted by $S_{i,j}$, i.e., $C(i,j) = -S_{i,j}$. The expected semantic error cost of all hypothesized word sequences on the lattice with respect to the reference can be accumulated from the semantic error cost of the words.

Assume that the training data is given by training utterances $r = \{1, \ldots, R\}$. $X_r = \{x_{r1}, \ldots, x_{rT_r}\}$ is the sequence of observations for utterance r, W_r is the word sequence in the reference (label transcription) for utterance r. W is a word sequence in the hypothesis set encapsulated in the decoded speech unit lattice for utterance r. $S_W = \{s_{W1}, \ldots, s_{WT}\}$ is the senone sequence corresponding to W, where s_{Wt} is the senone which frame x_{rt} is aligned with.

The MSEC is aimed at minimizing the expected semantic error cost of all possible word sequences given the reference. The objective function is formulated as

$$\begin{aligned}\mathcal{L}_{\text{MSEC}} &= \sum_{r=1}^{R} \sum_{W} P(W|X_r) C(W, W_r) \\ &= \sum_{r=1}^{R} \frac{\sum_W P(X_r|W) P(W) C(W, W_r)}{\sum_{W'} P(X_r|W') P(W')}\end{aligned} \tag{4.1}$$

where $C(W, W_r)$ is the semantic error cost of mis-recognizing the reference W_r as the hypothesis sentence W.

Take the derivative of Eq. (4.1) with respect to the activation $a_{rt}(s)$ for senone s at the output layer is

$$\begin{aligned}\frac{\partial \mathcal{L}_{MSEC}}{\partial a_{rt}(s)} &= \sum_u \frac{\partial \mathcal{L}_{MSEC}}{\partial \log p(x_{rt}|u)} \frac{\partial \log p(x_{rt}|u)}{\partial a_{rt}(s)} \\ &= \gamma_{rt}^{W}(s) \left\{ E_{P(W|s_{Wt}=s, X_r)}[C(W, W_r)] - E_{P(W|X_r)}[C(W, W_r)] \right\}\end{aligned} \tag{4.2}$$

where $\gamma_{rt}^{W}(s)$ is the posterior of being in senone s at time t, computed over the denominator lattice of the utterance r, $\log p(x_{rt}|u)$ is the log-likelihood of x_{rt} given senone u obtained by subtracting the log senone prior $\log p(u)$ from the log senone posterior $\log(y_t)$ in Eq. (2.40).

Eq. (4.2) is the error to be backpropagated through time to derive the gradients for all the parameters of BLSTM, in which $E_{P(W|s_{Wt}=s,X_r)}[C(W,W_r)]$ and $E_{P(W|X_r)}[C(W,W_r)]$ are obtained by accumulating the word-word semantic error costs $C(i,j)$ (i.e., $-S_{i,j}$) through performing the *forward-backward algorithm* on the decoded word lattice. The decoded lattices of the utterance X_r can be viewed as a directed graph WFST(X_r). For a state q of the WFST(X_r), $\mathcal{DP}(q)$ denotes the set of direct predecessors of q and $\mathcal{DS}(q)$ denotes the set of direct successors of q. The arrow (arc) directed from state p to state q is denoted by $l_{p,q}$, the weight on $l_{p,q}$ is denoted by $g_{p,q}$, the word (input label) of the $l_{p,q}$ is denoted by $m_{p,q}$ and the time for $l_{p,q}$ in X_r is $t_{p,q}$. $\mathcal{F}(X_r)$ is the set of final states in the WFST(X_r) and g_f is the final weight of the final state f. W_{rt} is the word at time t of the reference word sequence W_r. The first round of forward-backward is performed with forward and backward likelihood $\alpha_q^{(1)}$ and $\beta_q^{(1)}$.

$$\alpha_q^{(1)} = \sum_{p\in\mathcal{DP}(q)} \alpha_p^{(1)} g_{p,q} \tag{4.3}$$

$$\beta_q^{(1)} = \sum_{p\in\mathcal{DS}(q)} \beta_p^{(1)} g_{q,p} \tag{4.4}$$

$$P(W|X_r) = \sum_{f\in\mathcal{F}(X_r)} \alpha_f^{(1)} g_f \tag{4.5}$$

The second round of forward-backward is formulated as

$$\alpha_q^{(2)} = \frac{1}{\alpha_q^{(1)}} \sum_{p\in\mathcal{DP}(q)} \alpha_p^{(1)} g_{p,q} [\alpha_p^{(2)} + C(m_{p,q}, W_{rt_{p,q}})] \tag{4.6}$$

$$\beta_q^{(2)} = \frac{1}{\beta_q^{(1)}} \sum_{p\in\mathcal{DS}(q)} \beta_p^{(1)} g_{q,p} [\beta_p^{(2)} + C(m_{q,p}, W_{rt_{q,p}})] \tag{4.7}$$

$$E_{P(W|X_r)}[C(W,W_r)] = \frac{\sum_{f\in\mathcal{F}(X_r)} \alpha_f^{(1)} g_f \alpha_f^{(2)}}{P(W|X_r)} \tag{4.8}$$

where $\alpha_q^{(2)}$ and $\beta_q^{(2)}$ are the average cost of the partial state sequences preceding and following q

respectively. Assume that senone s is on the arrow directed from state p_s to state q_s of the WFST, we have

$$E_{P(W|s_{Wt}=s,X_r)}[C(W,W_r)] = \alpha_{p_s}^{(2)} + C(m_{p_s,q_s}, W_{rt_{p_s,q_s}}) + \beta_{q_s}^{(2)} \tag{4.9}$$

$$\gamma_{rt}^{W}(s) = \frac{\alpha_{p_s}^{(1)} g_{p_s,q_s} \beta_{q_s}^{(1)}}{P(W|X_r)} \tag{4.10}$$

The semantic error cost between sentences is computed from the semantic similarity matrix S and S can be designed in multiple ways to incorporate any form of external knowledge. In this chapter, we explore the following ways to incorporate the external knowledge. The word-word semantic similarity $S_{i,j}$ (i.e, $-C(i,j)$) can be computed from LSA. In LSA, a document is first represented by a column vector d indexed by the word in the vocabulary and the corpus of documents is represented by a word-document matrix $D = [d_1, \ldots, d_m]$. The columns of D are indexed by the documents. $D_{i,j}$ describes the number of occurrence of word i in document j. With the singular value decomposition (SVD) and the low-rank matrix approximation, we have $D \approx U_K \Sigma_K V_K^\top$, where Σ_K contains only the largest K singular values in Σ and U_K, and V_K contains the K left and right singular vectors corresponding to Σ_K, respectively. In the LSA framework, the *semantic similarity matrix* S is formulated as

$$S = U_K \Sigma_K^{-1} \Sigma_K^{-1} U_K^\top \tag{4.11}$$

where S is a square matrix with a dimension equal to the number of words in the vocabulary and the element $S_{i,j}$ of S is the *semantic similarity* between word i and word j. We set the diagonal elements of S to be the term frequency-inverse document frequency (tf-idf) [113] weights of the corresponding words and scale the non-diagonal elements proportionally.

The word-word semantic similarity $S_{i,j}$ can also be obtained from the distributed vector representations of words learned by an RNN language model (LM) (e.g., the continuous bag-of-words model and the continuous skip-gram model [114]) from a large amount of text data. By training an RNN LM, we obtain not only the model itself but also the vector-space word representations that are implicitly learned by the input layer weights. These word representations encode precise syntactic

and semantic word relationships as well as linguistic regularities and patterns [17]. Suppose w_i is a column vector representation for the word i, the word vocabulary can be represented by a matrix $W = [w_1, \ldots, w_v]^\top$. With the SVD and the low-rank approximation, the similarity matrix S can be formulated as

$$S = WW^\top = W_K I_K W_K^\top. \tag{4.12}$$

4.3 Distributed Word Representations Learned by Recurrent Neural Networks

The RNN LM has achieved extraordinary performance on many automatic speech recognition tasks [17]. By training an RNN LM, we obtains not only the model itself but also the learned vector-space word representations that are implicitly learned by the input layer weights. These representations are capable of capturing the syntactic and semantic regularities in language and the relationships between words.

As shown in Fig. 4.1, the input of the RNN LM is e_t represents a 1-of-N coding of the word at time t and the output is o_t representing the probability distribution over all the words at time t. One-hot vector e_{t+1} is the target (label) for the prediction o_t.

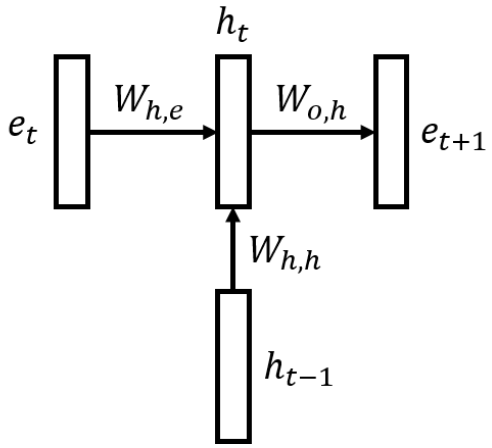


Figure 4.1: The architecture of RNN LM where the input layer matrix $W_{h,e}$ encodes the word representations.

Let h_t denote the hidden layer vector at time t. The RNN LM is formulated as

$$h_t = \text{sigmoid}(W_{h,e}e_t + W_{h,h}h_{t-1} + b_h) \tag{4.13}$$

$$o_t = \text{softmax}(W_{o,h}h_t) \tag{4.14}$$

The word representations exist in the input layer matrix $W_{h,e}$. Specifically, the i th column of the input layer matrix $W_{h,e}$ is the vector representation of the word that corresponds to the i th dimension of e_t.

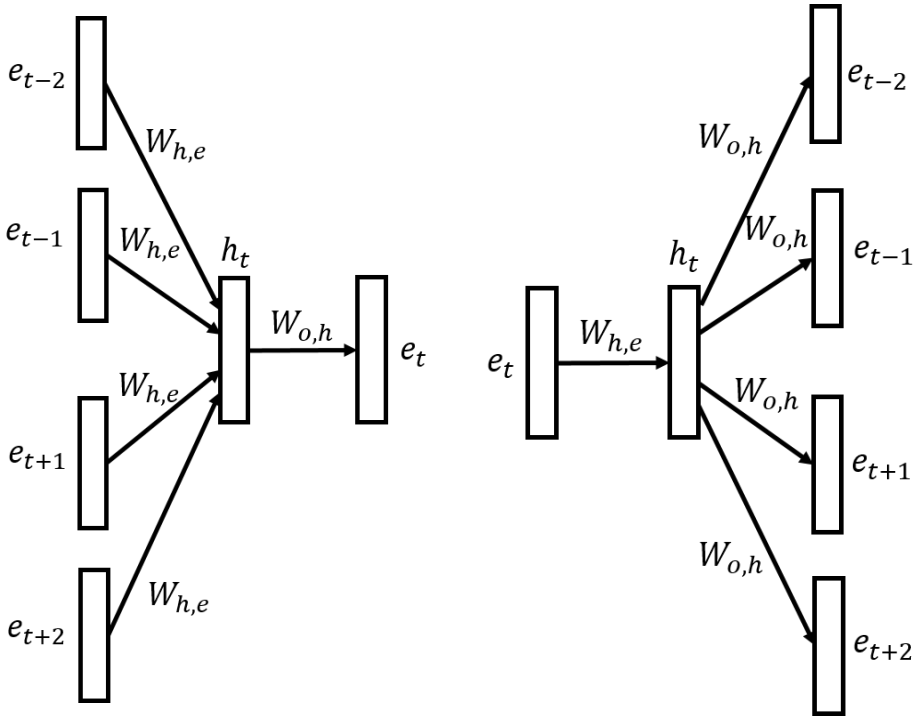
With advanced RNN architectures, the continuous bag-of-words (CBOW) model and the continuous skip-gram model are proposed in [114, 115]. The CBOW predicts the current word with context in both the past and future while the skip-gram predicts the surrounding words given the current word as shown in Fig. 4.2a and Fig. 4.2b. The word representations are encoded in the input layer matrix $W_{h,e}$. These two models achieve the state-of-the-art performance for measuring syntactic and semantic word similarities.

4.4 Latent Semantic Rational Kernel for Topic Spotting

Rational kernels are a series of kernels operated on WFSTs. If we compactly represent the ASR output of a speech signal as lattices, the topic classification task can be performed using SVM with rational kernels based on WFSTs (lattices). Let A be a WFSA defined over the semiring $\mathbb{K}$ and the alphabet Σ. Let B be a WFSA defined over the semiring $\mathbb{K}$ and alphabet Δ. Let T be a WFST over semiring $\mathbb{K}$ and ψ be a function mapped from $\mathbb{K}$ to the set of real number $\mathbb{R}$. The rational kernel $K(A, B)$ over A and B is given by

$$K(A, B) = \psi\left(\bigoplus_{(x,y)\in\Sigma\times\Delta} [\![A]\!](x) \otimes [\![T]\!](x, y) \otimes [\![B]\!](y)\right) \tag{4.15}$$

The n-gram rational kernel [106, 105] is widely used in speech and text classification tasks for its positive definite and symmetric property. Let L denote a WFST output (word lattice) from an ASR system, which defines a distribution $P_L(s)$ over all word sequences s represented by L. The



(a) CBOW

(b) Skip Gram

Figure 4.2: The architecture of advanced RNN-LM. a) CBOW predicts the current word e_t given the context e_{t-2}, e_{t-1}, e_{t+1}, e_{t+2}; b) skip gram predicts the context e_{t-2}, e_{t-1}, e_{t+1}, e_{t+2} given the current word e_t

expected number of occurences of an n-gram sequence x with n words in WFST L is given by

$$c(L, x) = \sum_{s} P_L(s) c(s, x) \tag{4.16}$$

where $c(s, x)$ denotes the number of occurences of x in the word sequence s.

Then n-gram rational kernel k_n for two WFSTs L_1 and L_2 is defined as

$$k_n(L_1, L_2) = \sum_{|x|=n} c(L_1, x) c(L_2, x) = \phi(L_1)^T \phi(L_2) \tag{4.17}$$

where $|\cdot|$ denote the number of words in a sequence. Therefore, the n-gram kernel is the sum of the products of the expected counts that L_1 and L_2 assign to their common n-gram sequences. With the n-gram kernel, we first map the WFSTs to vectors $\phi(L)$ in the n-gram space with each dimension being the expected count of an n-gram sequence and then take the inner product between vectors.

However, the n-gram rational kernel assumes that WFSTs from the same topic share many exact-matched n-grams and assumes that each n-gram term contributes equally to the discrimination of a topic. To circumvent these problems, we map the WFSTs into a latent semantic rational space with a reduced dimension where more sophisticated term-term relations can be accurately calculated. Let F denote a linear transform to map $\phi(L)$ in n-gram space to the latent semantic rational space, the LSRK is defined as

$$\begin{aligned} k_n(L_1, L_2) &= \langle F\phi(L_1), F\phi(L_2) \rangle = \phi(L_1)^T F^T F \phi(L_2) \\ &= \phi(L_1) S \phi(L_2) \end{aligned} \tag{4.18}$$

where S matrix is a *term-term semantic similarity matrix* which specifies the *semantic similarity* between n-gram terms, i.e., the value of matrix element $S_{i,j}$ measures the semantic similarity between terms i and j. The n-gram rational kernel can be viewed as a special case of LSRK by assuming the semantic similarity of the same term to be 1 and the semantic similarity between different terms to be 0, under which case the S matrix is degenerated to an identity matrix I.

In this chapter, the topic classification is performed by a multi-class SVM with LSRK that takes the decoded lattices of the utterances as the input. In the WFST framework, the LSRK is formulated

as follow

$$\begin{aligned} k_n(L_1, L_2) &= w[(L_1 \circ T) \circ \text{WFST}(F) \circ \text{WFST}(F)^{-1} \circ (T^{-1} \circ L_2)] \\ &= w[(L_1 \circ T) \circ \text{WFST}(S) \circ (T^{-1} \circ L_2)] \end{aligned} \tag{4.19}$$

where $w[B]$ denotes the shortest distance from the start state to the set of final states of the transducer B. The transducer T is used to extract all words defined as

$$T = (\Sigma \times \{\epsilon\})^* (\sum_{y \in \Sigma} \{y\} \times \{y\})^n (\Sigma \times \{\epsilon\})^* \tag{4.20}$$

where Σ is the word vocabulary and ϵ denotes empty label. WFST(F) is a WFST encoding the transform from n-gram space to latent semantic space. The composition of WFST(F) and WFST$(F)^{-1}$ is equivalent to WFST(S) which encodes the *semantic similarity matrix* S.

$$\text{WFST}(S) = (\epsilon \times \{\epsilon\})^* (\sum_{y \in \Sigma} \{y\} \times \{y\})^n (\epsilon \times \{\epsilon\})^* \tag{4.21}$$

Two neighboring states of WFST(S) are connected by $M \times N$ arcs. Each of these arcs represents one element in the S matrix. Specifically, $S_{i,j}$ is represented by an arc with input label i, output label j and weight $S_{i,j}$. The WFST(S) is constructed by a repetition of n sets of $M \times N$ arcs representing the S matrix that connects $(n+1)$ WFST states. Fig. 4.3 shows an example of WFST(S) in bi-gram case.

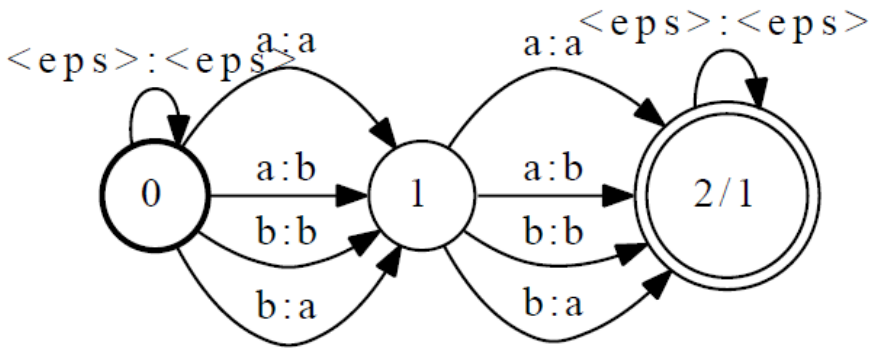


Figure 4.3: WFST(S) with vocabulary $\Sigma = \{a, b\}$ in bi-gram case.

As incorporating the full high-order n-gram features in LSA will lead to a prohibitively large matrix, certain pruning schemes on those high-order n-gram features have to be conducted in this case. Thus, we leave the exploration of the high-order features for future study. In this chapter, we only use unigram (word) features and S degenerates to a word-word semantic similarity matrix as discussed in Section 4.3. The LSRK can be generalized with respect to the semantic similarity matrix S to incorporate external knowledge from LSA and RNN LM as in Eq. (4.11) and Eq. (4.12) to achieve a better performance for topic spotting.

We compose the semantic similarity matrix WFST(S) with T transducer to construct the LSRK as in Eq. (4.19) that takes the word lattices of the speech signal as the input. The topic classification is performed by using multi-class SVM with the composed LSRK.

4.5 Experiments

4.5.1 Dataset Description

We evaluate the performance of the proposed framework on a large-scale CTS task, i.e., the 300 hours Switchboard-1 Release 2 (LDC97S62). It consists of 2348 two-sided telephone conversations from 543 speakers (302 males and 241 females) in the United States. One topic is assigned to each of the conversation between two callers and about 70 topics in total are provided in the corpus.

However, a large number of utterances within the 300 hours Switchboard data do not fit into a clear topic and are not suitable for the topic spotting task (e.g., "Oh yeah", "Um-hum", "You are right."). The selection of utterances are based on the length of the transcriptions after filtering out the filler words, functional words and stop words. We keep the utterances transcriptions of which have more than 20 words after filtering. The threshold is set based on the trade-off between the utterance duration and the number of remaining utterances. After the first round of filtering, we have 9192 utterances left. We further sift out the topics that have less than 200 utterances and finally have 4405 utterances on 19 different topics for topic spotting task. From each topic, we randomly select 90% utterances for training and 10% for testing. (The 3955 training and 450 test utterances used for topic spotting) as shown in Table 4.1. The rest of the Switchboard-1 corpus is used for training the acoustic model. The 3M words of the training transcripts are used to train a trigram language

Table 4.1: The number of training and test utterances for each topic selected for topic classification task.

Topic	Train	Test	Total
Weather Climate	250	28	278
Recycling	247	28	275
Restaurants	245	28	273
Recipes/Food/Cooking	242	28	270
Exercise and Fitness	230	26	256
Gun Control	212	24	236
Buying a Car	207	24	231
Pets	204	23	227
Gardening	200	23	223
Capital Punishment	197	23	220
TV Program	197	22	219
Auto Repairs	197	22	219
Public Education	196	22	218
Movies	193	22	215
Drug Testing	193	22	215
Womens Role	191	22	213
Hobbies and Crafts	188	21	209
Camping	186	21	207
Air Pollution	180	21	201
Total	3955	450	4405

model for decoding.

4.5.2 Discriminative training of BLSTM acoustic model for lattice generation (WFST)

We first train a BLSTM acoustic model for the LVCSR system to generate word lattices that are suited for topic spotting. A 36- dimensional log Mel filterbank feature vector is extracted and then concatenated with a 3-dimensional pitch feature vector (consisting of probability of voicing, log pitch and delta log pitch) [92] to form a 39 dimensional "log Mel filterbank + pitch" feature. To build a BLSTM, we stack 4 hidden layers and add a softmax output layer on the top to represent the 8784 senones posteriors. Each forward or backward hidden layer has 1024 hidden units and is connected to a 512 dimensional recurrent projection layer. The forward and backward projection layers are concatenated together (to form a 1024 dimensional vector) and fed as the input of the next BLSTM hidden layer. After appending delta and delta-delta coefficients to the 39 dimensional "log Mel filterbank + pitch" features, we use the 117 dimensional features with globally normalized zero

mean and unit variance as the input to the BLSTM. The BLSTM is randomly initialized and then trained (initial learning rate 0.00002) to minimize the cross-entropy (CE) criterion using senone-level forced alignment generated by a GMM-HMM system as the target.

The BLSTM in the baseline system is then trained with the MSEC criterion as described in Section 4.2. The semantic similarity matrix S_{MSEC} is constructed with LSA using Eq. (4.11) with a rank of 750. The word-document matrix D is formed with the transcriptions of a subset of training utterances (2438 in total) in the topic spotting task. The diagonal elements of S_{MSEC} is set to the tf-idf weights of the corresponding word and the non-diagonal elements are scaled proportionally. Since the size of S_{MSEC} is very large (over $30,000 \times 30,000$), we prune the non-diagonal elements by keeping only the largest $80,000$ elements and setting the rest to zeros. For comparison, we also discriminatively train the baseline BLSTM with MWE and sMBR.

4.5.3 LSRK for topic spotting

We generate the decoded lattices for the 3955 training utterances using BLSTM based LVCSR systems trained in Section 4.5.2 and train a multi-class SVM with LSRK using these lattices for topic spotting. The LSRK is constructed with a semantic similarity matrix S_{LSRK} derived from the RNN LM learned vector-space word representations in Eq. (4.12). Each word in vocabulary is represented by a 4000-dimensional vector learned from a skip-gram model [114]. The skip-gram model is trained on a billion characters of the English Wikipedia[1] using the word2vec toolkit [114]. The out-of-vocabulary words are represented by a zero vector. We approximate the S_{LSRK} with a matrix of rank K and pruned it to M non-zero non-diagonal elements. The S_{LSRK} is scaled and pruned in the same way as we did for S_{MSEC}.

The topic classification accuracies for lattices generated by LVCSR trained with different objective function are shown in Table 4.2. The lattices generated by MSEC trained BLSTM achieve the best topic classification accuracy 60.00% when $M = 160,000; K = 400$ or $M = 240,000; K = 800$ or $M = 320,000; K = 800$, which is 3.5% - 4.5% absolutely higher than the baseline BLSTM trained with cross-entropy criterion. The BLSTM trained with MSEC also achieves 2%- 3% absolute accuracy gains over other discriminative training objectives. The classification accuracies vary

[1] English Wikipedia is available on http://mattmahoney.net/dc/textdata.html

slightly when M and K go beyond $160,000$ and 400 respectively.

Table 4.2: The topic classification accuracies (%) of BLSTM-HMM systems trained with different objectives on the subset of Switchboard-1 Release 2. M is the number of non-zero non-diagonal elements left in S_{LSRK} after pruning and K is the rank of S_{LSRK} after low rank approximation.

M	K	Objective Functions			
		CE	sMBR	MWE	MSEC
160,000	400	56.89	58.00	57.56	**60.00**
	800	56.67	58.00	57.56	59.33
	1200	56.67	57.78	57.33	58.56
240,000	400	55.78	58.00	57.78	59.56
	800	56.22	57.56	57.33	**60.00**
	1200	56.00	57.33	57.56	59.33
320,000	400	56.22	58.44	57.78	59.56
	800	55.56	58.00	57.78	**60.00**
	1200	56.00	58.00	57.33	59.56

4.5.4 Large-vocabulary continuous speech recognition

We evaluate the LVCSR performance of the MSEC-trained BLSTM on the Switchboard portion of the 2000 HUB 5 English (LDC2002S09) and compare it with the other objectives. The ASR is conducted with the same acoustic and language models as the ones used in Sections 4.5.2 and 4.5.3 for topic spotting. The BLSTM trained with MSEC achieves 13.9% word error rate (WER), which is 0.7% and 0.8% absolutely lower than the BLSTMs trained with sMBR and MWE respectively as in Table 4.3. The degradation of LVCSR performanece is expected since MSEC is designed to minimize the expected semantic error cost instead of expected state or word errors as in sMBR or MWE.

Table 4.3: The LVCSR WER performance of BLSTM-HMM systems trained with different objectives on the Switchboard portion of the 2000 HUB 5 English dataset.

System	WER (%)
BLSTM CE	14.6
BLSTM sMBR	13.1
BLSTM MWE	13.2
BLSTM MSEC	13.9

4.6 Conclusion

In this chapter, we compensate for the mismatch between the objectives of ASR and spoken language understanding (SLU) by proposing an MSEC criterion to train the BLSTM acoustic model. MSEC aims at minimizing the expected semantic error cost of all possible word sequences on the lattices given the reference such that the ASR can generate lattices that are more semantically accurate and better suited for topic spotting with LSRK. The word-word semantic error cost is first computed from either the latent semantic analysis or distributed vector-space word representations learned from the RNNs and is then accumulated to form the expected semantic error cost of the hypothesized word sequences.

The MSEC achieves a 3.5% - 4.5% absolute topic classification accuracy improvement over the baseline BLSTM trained with the cross-entropy criterion on Switchboard dataset. We show that the MSEC training of BLSTM can help an LVCSR to generate lattices that are more semantically accurate and thus leads to a higher topic classification accuracy than other training objectives.

CHAPTER 5

SPEAKER-INVARIANT TRAINING FOR ROBUST SPEECH RECOGNITION

5.1 Introduction

The DNN based acoustic models have been widely used in ASR and have achieved extraordinary performance improvement [116, 117]. However, the performance of a speaker-independent (SI) acoustic model trained with speech data from a large number of speakers is still affected by the spectral variations in each speech unit caused by the inter-speaker variability. Many factors contribute to this inter-speaker variability, such as the differences in speakers' vocal tract configurations, ages, genders, accents and speaking rates. Such speaker variations lead to high variance in the spectral distribution of the speech signal that corresponds to each speech unit and thus large overlaps among distributions. Directly trained with the diffused data, the SI acoustic model has limited discriminative power and leads to high word error rate (WER) in ASR. Therefore, speaker-adaptive training [19, 20, 118, 119] and speaker adaptation methods are widely used to boost the recognition system performance, such as regularization-based [68, 70, 120, 72], transformation-based [76, 121, 122], singular value decomposition-based [123, 77, 78] and subspace-based [79, 82, 124] approaches.

Recently, adversarial learning has captured great attention of deep learning community given its remarkable success in estimating generative models [22]. In speech, it has been applied to noise-robust [26, 83, 27, 30, 28] and conversational ASR [125] using gradient reversal layer [23] or domain separation network [29]. Inspired by this, we propose *speaker-invariant training (SIT)* via adversarial learning to reduce the effect of speaker variability in acoustic modeling. In SIT, a DNN acoustic model and a DNN speaker classifier are jointly trained to simultaneously optimize the primary task of minimizing the senone classification loss and the secondary task of mini-maximizing the speaker classification loss. Through this adversarial multi-task learning procedure, a feature extractor is learned as the bottom layers of the DNN acoustic model that maps the input speech frames from different speakers into *speaker-invariant* and senone-discriminative deep hidden features, so that further senone classification is based on representations with the speaker factor already normal-

ized out. The DNN acoustic model with SIT can be directly used to generate word transcription for unseen test speakers through *one-pass online* decoding. On top of the SIT DNN, further adaptation can be performed to adjust the model towards the test speakers, achieving even higher ASR accuracy.

We evaluate SIT with ASR experiments on CHiME-3 dataset, the SIT FNN acoustic model achieves 4.99% relative WER improvement over the baseline SI FNN. Further, SIT achieves 4.86% relative WER gain over the SI FNN when the same unsupervised speaker adaptation process is performed on both models. With t-distributed stochastic neighbor embedding (t-SNE) [126] visualization, we show that, after SIT, the deep feature distributions of different speakers are well aligned with each other, which demonstrates the strong capability of SIT in reducing speaker-variability.

5.2 Related Work

Speaker-adaptive training (SAT) is proposed to generate canonical acoustic models coupled with speaker adaptation. For Gaussian mixture model (GMM)-hidden Markov model (HMM) acoustic model, SAT applies unconstrained [18] or constrained [127] model-space linear transformations that separately model the speaker-specific characteristics and are jointly estimated with the GMM-HMM parameters to maximize the likelihood of the training data. Cluster-adaptive training (CAT) [128] is then proposed to use a linear interpolation of all the cluster means as the mean of the particular speaker instead of a single cluster as representative of a particular speaker. However, SAT of GMM-HMM needs to have two sets of models, the SI model and canonical model. During testing, the SI model is used to generate the first pass decoding transcription, and the canonical model is combined with speaker-specific transformation to adapt to the new speaker.

For DNN-HMM acoustic model, CAT [20] and multi-basis adaptive neural networks [19] are proposed to represent the weight and/or the bias of the speaker-dependent (SD) affine transformation in each hidden layer of a DNN acoustic model as a linear combination of SI bases, where the combination weights are low-dimensional SD speaker representations. The canonical SI bases with reduced variances are jointly optimized with the SD speaker representations during the SAT to minimize the cross-entropy loss. During unsupervised adaptation, the test speaker representations are re-estimated using alignments from the first-pass decoding of the test data with SI DNN as the su-

pervisions and are used in the second-pass decoding to generate the transcription. Factorized hidden layer [129] is similar to [20, 19], but includes SI DNN weights as part of the linear combination. In [130], SD speaker codes are transformed by a set of SI matrices and then directly added to the biases of the hidden-layer affine transformations. The speaker codes and SI transformations are jointly estimated during SAT. For these methods, two passes of decoding are required to generate the final transcription in unsupervised adaption setup, which increases the computational complexity of the system.

In [119, 131], an SI adaptation network is learned to derive speaker-normalized features from i-vectors to train the canonical DNN acoustic model. The i-vectors for the test speakers are then estimated and used for decoding after going through the SI adaptation network. In [125], a reconstruction network is trained to predict the input i-vector given the speech feature and its corresponding i-vector are at the input of the acoustic model. The mean-squared error loss of the i-vector reconstruction and the cross-entropy loss of the DNN acoustic model are jointly optimized through adversarial multi-task learning. Although these methods generate the final transcription with one-pass of decoding, they need to go through the entire test utterances in order to estimate the i-vectors, making it impossible to perform online decoding. Moreover, the accuracy of i-vectors estimation are limited by the duration of the test utterances. The estimation of i-vector for each utterance also increases the computational complexity of the system.

SIT directly minimizes the speaker variations by optimizing an adversarial multi-task objective other than the most basic cross entropy object as in SAT. It forgoes the need of estimating any additional SI bases or speaker representations during training or testing. The direct use of SIT DNN acoustic model in testing enables the generation of word transcription for unseen test speakers through *one-pass online* decoding. Moreover, it effectively suppresses the inter-speaker variability via a lightweight system with much reduced training parameters and computational complexity. To achieve additional gain, unsupervised speaker adaptation can also be further conducted on the SIT model with one extra pass of decoding.

5.3 Speaker-Invariant Training

To perform SIT, we need a sequence of speech frames $X = \{x_1, \ldots, x_N\}$, a sequence of senone labels $Y = \{y_1, \ldots, y_N\}$ aligned with X and a sequence of speaker labels $S = \{s_1, \ldots, s_N\}$ aligned with X. The goal of SIT is to reduce the variances of hidden and output units distributions of the DNN acoustic model that are caused by the inherent inter-speaker variability in the speech signal. To achieve speaker-robustness, we learn a *speaker-invariant* and *senone-discriminative* deep hidden feature in the DNN acoustic model through adversarial multi-task learning and make senone posterior predictions based on the learned deep feature. In order to do so, we view the first few layers of the acoustic model as a feature extractor network M_f with parameters θ_f that maps input speech frames X from different speakers to deep hidden features $F = \{f_1, \ldots, f_N\}$ (see Fig. 5.1) and the upper layers of the acoustic model as a senone classifier M_y with parameters θ_y that maps the intermediate features F to the senone posteriors $p(q|f;\theta_y), q \in \mathcal{Q}$ as follows:

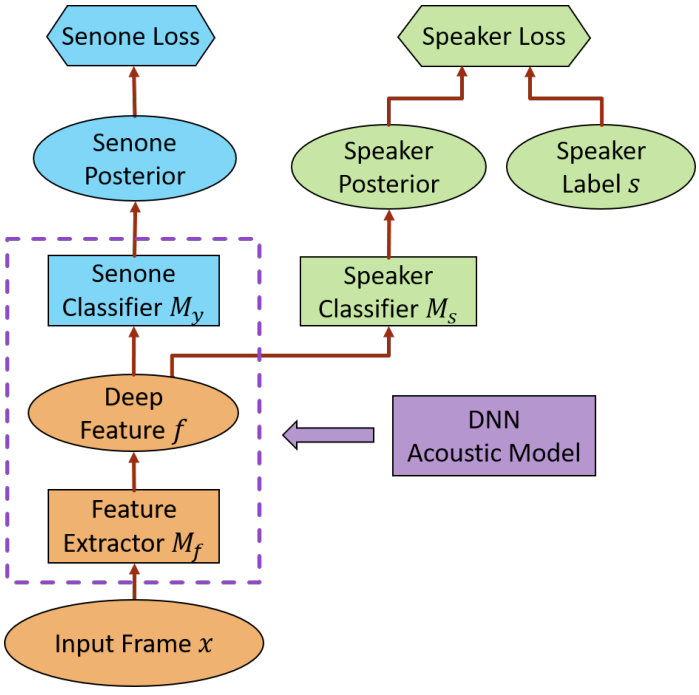


Figure 5.1: The framework of speaker-invariant training via adversarial learning for unsupervised adaptation of the acoustic models

$$M_y(f_i) = M_y(M_f(x_i)) = p_y(q|x_i;\theta_f,\theta_y) \tag{5.1}$$

We further introduce a speaker classifier network M_s which maps the deep features F to the speaker posteriors $p_s(a|x_i;\theta_s,\theta_f), a \in \mathcal{A}$ as follows:

$$M_s(M_f(x_i)) = p_s(a|x_i;\theta_s,\theta_f) \tag{5.2}$$

where a is one speaker in the set of all speakers $\mathcal{A}$.

To make the deep features F speaker-invariant, the distributions of the features from different speakers should be as close to each other as possible. Therefore, the M_f and M_s are jointly trained with an adversarial objective, in which θ_f is adjusted to *maximize* the speaker classification loss $\mathcal{L}^f_{\text{speaker}}(\theta_f)$ while θ_s is adjusted to *minimize* the frame-level speaker classification loss $\mathcal{L}^s_{\text{speaker}}(\theta_s)$ below:

$$\begin{aligned}\mathcal{L}_{\text{speaker}}(\theta_f,\theta_s) &= -\sum_i^N \log p_s(s_i|x_i;\theta_f) \\ &= -\sum_i^N \sum_{a\in\mathcal{A}} \mathbb{1}_{[a=s_i]} \log M_s(M_f(x_i))\end{aligned} \tag{5.3}$$

where s_i denote the speaker label for the input frame x_i of the acoustic model.

This minimax competition will first increase the discriminativity of M_s and the speaker-invariance of the features generated by M_f, and will eventually converge to the point where M_f generates extremely confusing features that M_s is unable to distinguish.

At the same time, we want to make the deep features senone-discriminative by minimizing the cross-entropy loss between the predicted senone posteriors and the senone labels as follows:

$$\mathcal{L}_{\text{senone}}(\theta_f,\theta_y) = -\sum_i p_y(y_i|x_i;\theta_f,\theta_y) M_y(M_f(x_i)) \tag{5.4}$$

In SIT, the acoustic model network and the condition classifier network are trained to jointly optimize the primary task of senone classification and the secondary task of speaker classification

with an adversarial objective function. Therefore, the total loss is constructed as

$$\mathcal{L}_{\text{total}}(\theta_f, \theta_y, \theta_s) = \mathcal{L}_{\text{senone}}(\theta_f, \theta_y) - \lambda \mathcal{L}_{\text{speaker}}(\theta_s, \theta_f) \tag{5.5}$$

where λ controls the trade-off between the senone loss and the speaker classification loss in Eq.(5.4) and Eq.(5.3) respectively.

We need to find the optimal parameters $\hat{\theta}_y, \hat{\theta}_f$ and $\hat{\theta}_s$ such that

$$(\hat{\theta}_f, \hat{\theta}_y) = \min_{\theta_y, \theta_f} \mathcal{L}_{\text{total}}(\theta_f, \theta_y, \hat{\theta}_s) \tag{5.6}$$

$$\hat{\theta}_s = \max_{\theta_s} \mathcal{L}_{\text{total}}(\hat{\theta}_f, \hat{\theta}_y, \theta_s) \tag{5.7}$$

All the parameters of acoustic model network and the speaker classifier network are updated jointly as follows via back propagation with stochastic gradient descent (SGD):

$$\theta_f \leftarrow \theta_f - \mu \left[\frac{\partial \mathcal{L}_{\text{senone}}}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_{\text{speaker}}}{\partial \theta_f} \right] \tag{5.8}$$

$$\theta_s \leftarrow \theta_s - \mu \frac{\partial \mathcal{L}_{\text{speaker}}}{\partial \theta_s} \tag{5.9}$$

$$\theta_y \leftarrow \theta_y - \mu \frac{\partial \mathcal{L}_{\text{senone}}}{\partial \theta_y} \tag{5.10}$$

where μ is the learning rate.

Note that the negative coefficient $-\lambda$ in Eq. (5.8) induces reversed gradient that maximizes $\mathcal{L}_{\text{speaker}}(\theta_f, \theta_s)$ in Eq. (5.3) and makes the deep feature speaker-invariant. Without the reversal gradient, SGD would make representations different across domains in order to minimize $\mathcal{L}_{\text{speaker}}(\theta_f, \theta_s)$. For easy implementation, gradient reversal layer is introduced in [23], which acts as an identity transform in the forward propagation and multiplies the gradient by $-\lambda$ during the backward propagation.

The optimized network consisting of M_f and M_s is used as the SIT acoustic model for ASR on test data.

5.4 Experiments

In this chapter, we perform SIT on a FNN-hidden Markov model (HMM) acoustic model for ASR on CHiME-3 dataset.

5.4.1 Dataset Description

The CHiME-3 dataset is released with the 3rd CHiME speech Separation and Recognition Challenge [132], which incorporates the Wall Street Journal corpus sentences spoken in challenging noisy environments, recorded using a 6-channel tablet based microphone array. CHiME-3 dataset consists of both real and simulated data. The real speech data was recorded in five real noisy environments (on buses (BUS), in cafés (CAF), in pedestrian areas (PED), at street junctions (STR) and in booth (BTH)). To generate the simulated data, the clean speech is first convolved with the estimated impulse response of the environment and then mixed with the background noise separately recorded in that environment [133]. The noisy training data consists of 1999 real noisy utterances from 4 speakers, and 7138 simulated noisy utterances from 83 speakers in the WSJ0 SI-84 training set recorded in 4 noisy environments. There are 3280 utterances in the development set including 410 real and 410 simulated utterances for each of the 4 environments. There are 2640 utterances in the test set including 330 real and 330 simulated utterances for each of the 4 environments. The speakers in training set, development set and the test set are mutually different (i.e., 12 different speakers in the CHiME-3 dataset). The training, development and test data sets are all recorded in 6 different channels.

In the experiments, we use 9137 noisy training utterances in the CHiME-3 dataset as the training data. The real and simulated development data in CHiME-3 are used as the test data. Both the training and test data are far-field speech from the 5th microphone channel. The WSJ0 text corpus with 5K-word lexicon is used to train a 3-gram language model utilized in out experiments.

5.4.2 Baseline System

In the baseline system, we first train an SI deep FNN-HMM acoustic model using 9137 noisy training utterances with cross-entropy criterion.

The 29-dimensional log Mel filterbank features together with 1st and 2nd order delta features (totally 87-dimensional) for both the clean and noisy utterances are extracted by following the process in [134]. Each frame is spliced together with 5 left and 5 right context frames to form a 957-dimensional feature. The spliced features are fed as the input of the deep FNN after global mean and variance normalization. The FNN has 7 hidden layers with 2048 hidden units for each layer. The output layer of the FNN has 3012 output units corresponding to 3012 senone labels. Senone-level forced alignment of the clean data is generated using a Gaussian mixture model-HMM system. As shown in Table 5.1, the WERs for the SI FNN are 17.84% and 17.72% respectively on real and simulated test data respectively. Note that our experimental setup does not achieve the state-of-the-art performance on CHiME-3 dataset (e.g., we did not perform beamforming, sequence training or use recurrent neural network language model for decoding.) since our goal is to simply verify the effectiveness of SIT in reducing inter-speaker variability.

5.4.3 Speaker-Invariant Training for Robust Speech Recognition

We further perform SIT on the baseline noisy FNN acoustic model with 9137 noisy training utterances in CHiME-3. The feature extractor M_f is initialized with the first N_h layers of the FNN and the senone classifier is initialized with the rest $(7 - N_h)$ hidden layers plus the output layer. N_h indicates the position of the deep hidden feature in the acoustic model. The speaker classifier M_s is a feedforward FNN with 2 hidden layers and 512 hidden units for each layer. The output layer of M_s has 87 units predicting the posteriors of 87 speakers in the training set. M_f, M_y and M_s are jointly trained with an adversarial multi-task objective as described in Section 5.3. N_h and λ are fixed at 2 and 3.0 in our experiments. The SIT FNN acoustic model achieves 16.95% and 16.54% WER on the real and simulated test data respectively, which are 4.99% and 6.66% relative improvements over the SI FNN baseline.

5.4.4 Visualization of Deep Features

We randomly select two male speakers and two female speakers from the noisy training set and extract speech frames aligned with the phoneme "ah" for each of the four speakers. In Figs. 5.2 and 5.3, we visualize the deep features F generated by the SI and SIT FNN acoustic models when the

Table 5.1: The ASR WER (%) performance of SI and SIT FNN acoustic models on real and simulated development set of CHiME-3.

System	Data	BUS	CAF	PED	STR	Avg.
SI	Real	24.77	16.12	13.39	17.27	17.84
	Simu	18.07	21.44	14.68	16.70	17.72
SIT	Real	22.91	15.63	12.77	16.66	**16.95**
	Simu	16.64	20.23	13.53	15.96	**16.54**

"ah" frames of the four speakers are given as the input using t-SNE. In Fig. 5.2, the deep feature distributions in the SI model for the male (in red and green) and female speakers (in back and blue) are far away from each other and even the distributions for the speakers of the same gender are separated from each other. While after SIT, the deep feature distributions for all the male and female speakers are well aligned with each other as shown in Fig. 5.3. The significant increase in the overlap among distributions of different speakers justifies that the SIT remarkably enhances the speaker-invariance of the deep features F. The adversarial optimization of the speaker classification loss does not just serve as a regularization term to achieve better generalization on the test data.

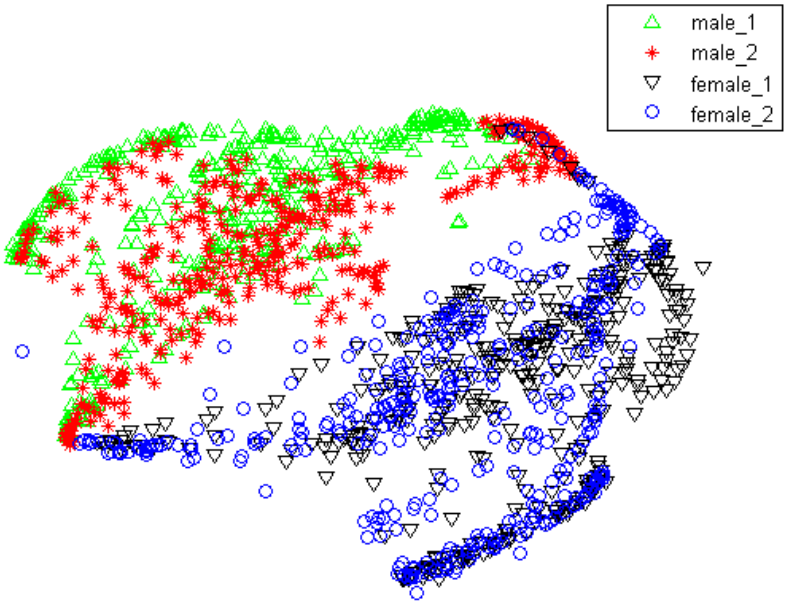


Figure 5.2: t-SNE visualization of the deep features F generated by the SI FNN acoustic model when speech frames aligned with phoneme "ah" from two male and two female speakers in CHiME-3 training set are fed as the input. 1095, 729, 1057, 423 deep features are generated for "female 1", "female 2", "male 1" and "male 2" respectively.

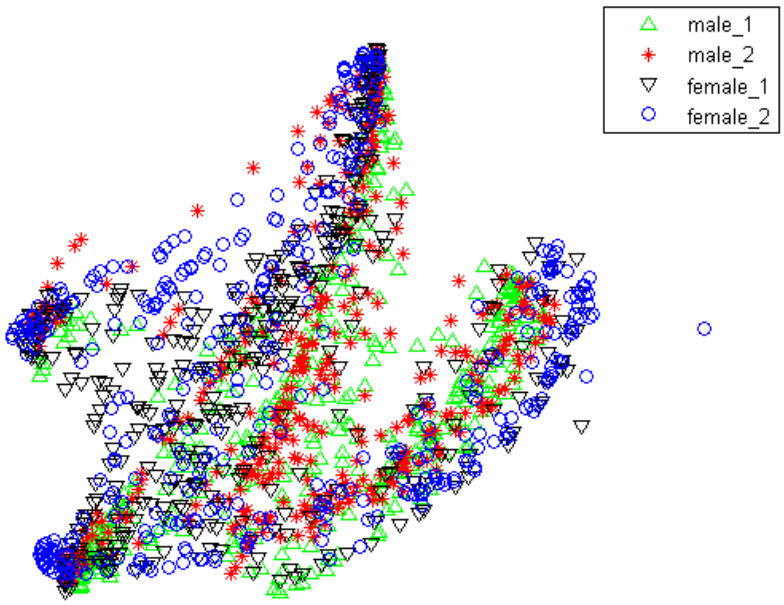


Figure 5.3: t-SNE visualization of the deep features F generated by the SIT FNN acoustic model when the same speech frames as in Fig. 5.2 are fed as the input. 1095, 729, 1057, 423 deep features are generated for "female 1", "female 2", "male 1" and "male 2" respectively.

5.4.5 Unsupervised Speaker Adaptation

SIT aims at suppressing the effect of inter-speaker variability on FNN acoustic model so that the acoustic model is more compact and has stronger discriminative power. When adapted to the same test speakers, the SIT FNN is expected to achieve higher ASR performance than the baseline SI FNN due to the smaller overlaps among the distributions of different speech units.

In our experiment, we adapt the SI and SIT FNNs to each of the 4 speakers in the test set in an unsupervised fashion. The constrained re-training (CRT) [135] method is used for adaptation, where we re-estimate the FNN parameters of only a subset of layers while holding the remaining parameters fixed during cross-entropy training. The adaptation target (1-best alignment) is obtained through the first-pass decoding of the test data, and the second-pass decoding is performed using the SA SI and SI FNNs.

The WER results for unsupervised speaker adaptation is shown in Table 5.2, in which only the bottom 2 layers of the SI and SIT FNNs are adapted during CRT. The speaker-adapted (SA) SIT FNN achieves 15.46% WER which is 4.86% relatively higher than the SA SI FNN. The CRT

adaptation provides 8.91% and 8.79% relative WER gains over the unadapted SI and SIT models respectively. The lower WER after speaker adaptation indicates that SIT has effectively reduced the high variance and overlap in an SI acoustic model caused by the inter-speaker variability.

Table 5.2: The ASR WER (%) performance of SA SI and SA SIT FNN acoustic models after CRT unsupervised speaker adaptation on real development set of CHiME-3.

System	BUS	CAF	PED	STR	Avg.
SA SI	22.76	15.56	11.52	15.37	16.25
SA SIT	21.42	14.79	11.11	14.70	**15.46**

5.5 Conclusions

In this chapter, SIT is proposed to suppress the effect of inter-speaker variability on the SI DNN acoustic model. In SIT, a DNN acoustic model and a speaker classifier network are jointly optimized to minimize the senone classification loss, and simultaneously mini-maximize the speaker classification loss. Through this adversarial multi-task learning procedure, a feature extractor network is learned to map the input frames from different speakers to deep hidden features that are both *speaker-invariant* and senone-discriminative.

Evaluated on CHiME-3 dataset, a deep SIT FNN acoustic model achieves 4.99% relative WER improvement over the baseline SI FNN. With the unsupervised adaptation towards the test speakers using CRT, the SA SIT FNN achieves additional 8.79% relative WER gain, which is 4.86% relatively improved over the SA SI FNN. With t-SNE visualization, we show that, after SIT, the deep feature distributions of different speakers are well aligned with each other, which verifies the strong capability of SIT in reducing speaker-variability.

SIT forgoes the need of estimating any additional SI bases or speaker representations which are necessary in other conventional approaches such as SAT. The SIT trained DNN acoustic model can be directly used to generate the transcription for unseen test speakers through *one-pass online* decoding. It enables a lightweight speaker-invariant ASR system with reduced number of parameters for both training and testing. Additional gains are achievable by performing further unsupervised speaker adaptation on top of the SIT model.

In the future, we will evaluate the performance of the i-vector based speaker-adversarial multi-

task learning [125] on CHiME-3 dataset and compare it with the proposed SIT. Moreover, we will perform SIT on thousands of hours of data to verify the its scalability to large dataset.

CHAPTER 6

ADVERSARIAL TEACHER-STUDENT LEARNING FOR UNSUPERVISED ADAPTATION

6.1 Introduction

With the advance of deep learning, the performance of ASR has been greatly improved [136, 137, 116, 138, 117]. However, the ASR still suffers from large performance degradation when a well-trained acoustic model is presented in a new domain [67, 139]. Many domain adaptation techniques were proposed to address this issue, such as regularization-based [68, 70, 120, 72], transformation-based [76, 121, 122], singular value decomposition-based [123, 77, 78] and subspace-based [79, 82, 118, 124] approaches. Although these methods effectively mitigate the mismatch between source and target domains, they reply on the transcription or the first-pass decoding hypotheses of the adaptation data.

To address these limitations, teacher-student (T/S) learning [24] is used to achieve unsupervised adaptation [25] with no exposure to any transcription or decoded hypotheses of the adaptation data. In T/S learning, the posteriors generated by the teacher model are used in lieu of the hard labels derived from the transcriptions to train the target-domain student model. Although T/S learning achieves large word error rate (WER) reduction in domain adaptation, it is similar to the traditional training criterion such as cross entropy (CE) which implicitly handles the variations in each speech unit (e.g. senone) caused by the speaker and environment variability in addition to phonetic variations. The structure of vocal tract, regional dialect and speaker idiosyncracies contribute to the variations in the characteristics of the speakers' voice. Differences in signal-to-noise ratios, types of noise sources, room impulse responses lead to the environment variations in the adaptation speech.

Recently, adversarial training has become a hot topic in deep learning with its great success in estimating generative models [22]. It has also been applied to noise-robust [26, 83, 27, 30] and speaker-invariant [21] ASR using gradient reversal layer [23] or domain separation network [29]. A deep intermediate feature is learned to be both discriminative for the main task of senone

classification and invariant with respect to the shifts among different conditions. Here, one condition refers to one particular speaker or one acoustic environment. For unsupervised adaptation, both the T/S learning and adversarial training forgo the need for any labels or decoded results of the adaptation data. T/S learning is more suitable for the situation where parallel data is available since the paired data allows the student model to be better-guided by the knowledge from the source model, while adversarial training is more powerful when such data is not available.

To benefit from both methods, in this chapter, we advance T/S learning with *adversarial T/S training* for condition-robust unsupervised domain adaptation, where a student acoustic model and a domain classifier are jointly trained to minimize the Kullback-Leibler (KL) divergence between the output distributions of the teacher and student models as well as to min-maximize the condition classification loss through adversarial multi-task learning. A senone-discriminative and *condition-invariant* deep feature is learned in the adapted student model through this procedure. Based on this, we further propose the *multi-factorial adversarial (MFA)* T/S learning where the condition variabilities caused by multiple factors are minimized simultaneously. Evaluated with the noisy CHiME-3 test set, the proposed method achieves 44.60% and 5.38% relative WER improvements over the clean model and a strong T/S adapted baseline acoustic model, respectively.

6.2 Teacher-Student Learning

By using T/S learning for unsupervised adaption, we want to learn a student acoustic model that can accurately predict the senone posteriors of the target-domain data from a well-trained source-domain teacher acoustic model. To achieve this, we only need two sequences of *unlabeled* parallel data, i.e., an input sequence of source-domain speech frames to the teacher model $X^T = \{x_1^T, \ldots, x_N^T\}$ and an input sequence of target-domain speech frames to the student model $X^S = \{x_1^S, \ldots, x_N^S\}$. X^T and X^S are parallel to each other, i.e, each pair of x_i^S and $x_i^T, \forall i \in \{1, \ldots, N\}$ are frame-by-frame synchronized.

T/S learning aims at minimizing the Kullback-Leibler (KL) divergence between the output distributions of the teacher model and the student model by taking the unlabeled parrallel data X^T and X^S as the input to the models. The KL divergence between the teacher and student output

distributions $p_T(q|x_i^T;\theta_T)$ and $p_S(q|x_i^S;\theta_S)$ is

$$\mathcal{KL}(p_T||p_S) = \sum_i \sum_{q\in\mathcal{Q}} p_T(q|x_i^T;\theta_T) \log\left(\frac{p_T(q|x_i^T;\theta_T)}{p_S(q|x_i^S;\theta_S)}\right) \tag{6.1}$$

where q is one of the senones in the senone set $\mathcal{Q}$, i is the frame index, θ_T and θ_S are the parameters of the teacher and student models respectively. To learn a student network that approximates the given teacher network, we minimize the KL divergence with respect to only the parameters of the student network while keeping the parameters of the teacher model fixed, which is equivalent to minimizing the loss function below:

$$\mathcal{L}(\theta_S) = -\sum_i \sum_{q\in\mathcal{Q}} p_T(q|x_i^T;\theta_T) \log p_S(q|x_i^S;\theta_S) \tag{6.2}$$

The target domain data used to adapt the student model is usually recorded under multiple conditions, i.e., the adaptation data often comes from a large number of different talkers speaking under various types of environments (e.g., home, bus, restaurant and etc). T/S learning can only implicitly handle the inherent speaker and environment variability in the speech signal and its robustness can be improved if it can explicitly handle the condition invariance.

6.3 Adversarial Teacher-Student Learning

In this section, we propose the *adversarial T/S learning* (see Fig. 6.1) to effectively suppress the condition (i.e., speaker and environment) variations in the speech signal and achieve robust unsupervised adaptation with multi-conditional adaptation data.

Similar to the T/S learning, we first clone the student acoustic model from the teacher and use unlabeled parallel data as the input to adapt the student model. To achieve condition-robustness, we learn a *condition-invariant* and *senone-discriminative* deep feature in the adapted student model through the senone posteriors generated by the teacher model and the condition label for each frame. In order to do so, we view the first few layers of the acoustic model as a feature extractor with parameters θ_f that maps input speech frames X^S of different conditions to deep intermediate features $F^S = \{f_1^S, \ldots, f_N^S\}$ and the upper layers of the student network as a senone classifier M_y with

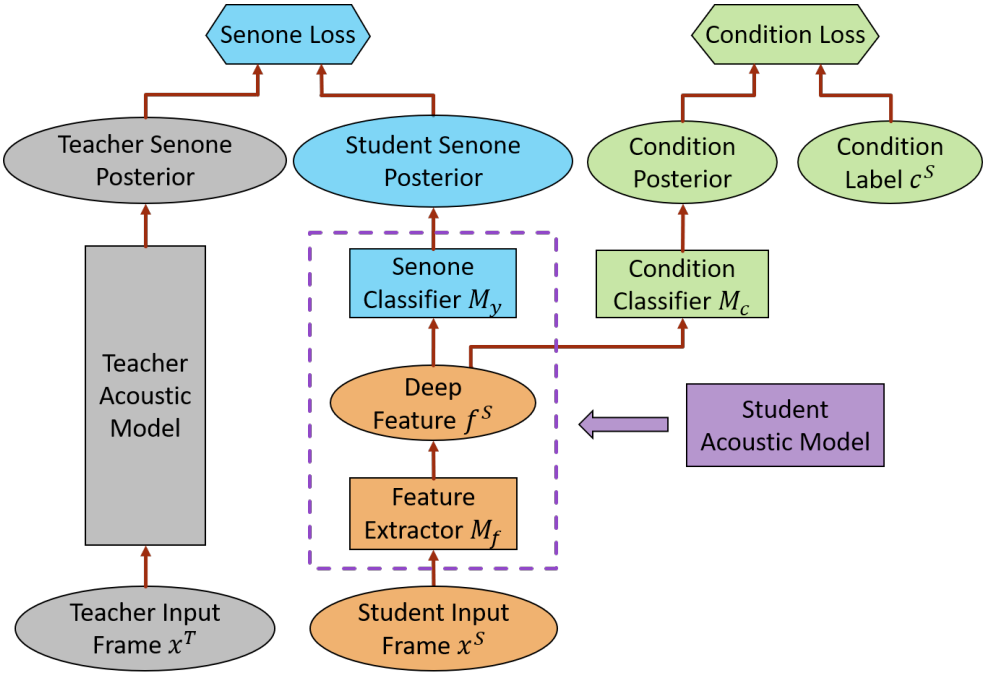


Figure 6.1: The framework of adversarial T/S learning for unsupervised adaptation of the acoustic models

parameters θ_y that maps the intermediate features F^S to the senone posteriors $p_S(q|f_i^S;\theta_y), q \in \mathcal{Q}$ as follows:

$$M_y(f_i^S) = M_y(M_f(x_i^S)) = p_S(q|x_i^S;\theta_f,\theta_y) \tag{6.3}$$

where we have $\theta_S = \{\theta_f, \theta_y\}$ as the student model.

We further introduce a condition classifier network M_c with θ_c which maps the deep features F^S to the condition posteriors $p_c(a|x_i^S;\theta_c,\theta_f), a \in \mathcal{A}$ as follows:

$$M_c(M_f(x_i^S)) = p_c(a|x_i^S;\theta_c,\theta_f) \tag{6.4}$$

where a is one condition in the set of all conditions $\mathcal{A}$.

To make the deep features F^S condition-invariant, the distributions of the features from different conditions should be as close to each other as possile. Therefore, the M_f and M_c are jointly trained with an adversarial objective, in which θ_f is adjusted to *maximize* the condition classification loss

$\mathcal{L}_{\text{condition}}(\theta_f, \theta_c)$ while θ_c is adjusted to *minimize* the $\mathcal{L}_{\text{condition}}(\theta_f, \theta_c)$ below:

$$\begin{aligned}\mathcal{L}_{\text{condition}}(\theta_f, \theta_c) &= -\sum_{i}^{N} \log p_c(c_i^S | x_i^S; \theta_f, \theta_c) \\ &= -\sum_{i}^{N} \sum_{a \in \mathcal{A}} \mathbb{1}_{[a=c_i^S]} \log M_c(M_f(x_i^S))\end{aligned} \tag{6.5}$$

where c_i^S denote the condition label for the input frame x_i^S of the student model.

This minimax competition will first increase the discriminativity of M_c and the condition-invariance of the features generated by M_f and will eventually converge to the point where M_f generates extremely confusing features that M_c is unable to distinguish.

At the same time, we use T/S learning to let the behavior of the student model in the target domain approach the behavior of the teacher model in the source domain by minimizing the KL divergence of the output distributions between the student and teacher acoustic models. Equivalently, we minimize the loss function in Eq. (6.2) as re-formulated below:

$$\mathcal{L}_{\text{TS}}(\theta_f, \theta_y) = -\sum_{i} \sum_{q \in \mathcal{Q}} p_T(q | x_i^T; \theta_f, \theta_y) M_y(M_f(x_i^S)) \tag{6.6}$$

In adversarial T/S learning, the student network and the condition classifier network are trained to jointly optimize the primary task of T/S learning using soft targets from the teacher model and the secondary task of condition classification with an adversarial objective function. Therefore, the total loss is constructed as

$$\mathcal{L}_{\text{total}}(\theta_f, \theta_y, \theta_c) = \mathcal{L}_{\text{TS}}(\theta_f, \theta_y) - \lambda \mathcal{L}_{\text{condition}}(\theta_f, \theta_c) \tag{6.7}$$

where λ controls the trade-off between the T/S loss and the condition classification loss in Eq.(6.6) and Eq.(6.5) respectively.

We need to find the optimal parameters $\hat{\theta}_y, \hat{\theta}_f$ and $\hat{\theta}_c$ such that

$$(\hat{\theta}_f, \hat{\theta}_y) = \min_{\theta_y, \theta_f} \mathcal{L}_{\text{total}}(\theta_f, \theta_y, \hat{\theta}_c) \tag{6.8}$$

$$\hat{\theta}_c = \max_{\theta_c} \mathcal{L}_{\text{total}}(\hat{\theta}_f, \hat{\theta}_y, \theta_c) \tag{6.9}$$

The parameters are updated as follows via back propagation through time with stochastic gradient descent (SGD):

$$\theta_f \leftarrow \theta_f - \mu \left[\frac{\partial \mathcal{L}_{\text{TS}}}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_{\text{condition}}}{\partial \theta_f} \right] \tag{6.10}$$

$$\theta_c \leftarrow \theta_c - \mu \frac{\partial \mathcal{L}_{\text{condition}}}{\partial \theta_c} \tag{6.11}$$

$$\theta_y \leftarrow \theta_y - \mu \frac{\partial \mathcal{L}_{\text{TS}}}{\partial \theta_y} \tag{6.12}$$

where μ is the learning rate.

Note that the negative coefficient $-\lambda$ in Eq. (6.10) induces reversed gradient that maximizes $\mathcal{L}_{\text{condition}}(\theta_f, \theta_c)$ in Eq. (6.5) and makes the deep feature condition-invariant. For easy implementation, gradient reversal layer is introduced in [23], which acts as an identity transform in the forward propagation and multiplies the gradient by $-\lambda$ during the backward propagation.

The optimized student network consisting of M_f and M_y is used as the adapted acoustic model for ASR in the target-domain.

6.4 Multi-factorial Adversarial Teacher-Student Learning

Speaker and environment are two different factors that contribute to the inherent variability of the speech signal. In Section 6.3, adversarial T/S learning is proposed to reduce the variations induced by the single condition. For a more comprehensive and thorough solution to the condition variability problem, we further propose the multi-factorial adversarial (MFA) T/S learning, in which multiple factors causing the condition variability are suppressed simultaneously through adversarial multi-task learning.

In MFA T/S framework, we keep the senone classifier M_y and feature extractor M_f the same

as in adversarial T/S, but introduce R condition classifiers $M_c^r, r = 1, \ldots, R$. M_c^r maps the deep feature to the posteriors of the p-th condition. To make the deep features F^S condition-invariant to each factor, we jointly train M_f and M_c with an adversarial objective, in which θ_f is adjusted to *maximize* the total condition classification loss of all factors while θ_c^r is adjusted to *minimize* the total condition classification loss of all factors. At the same time, we minimize the KL divergence between the output distributions of the teacher and student models. The total loss function for MFA T/S learning is formulated as

$$\mathcal{L}_{\text{total}}(\theta_f, \theta_y, \theta_c^1, \ldots, \theta_c^R) = \mathcal{L}_{\text{TS}}(\theta_f, \theta_y) - \lambda \sum_{r=1}^{R} \mathcal{L}_{\text{condition}}^r(\theta_c^r, \theta_f) \tag{6.13}$$

where $\mathcal{L}_{\text{TS}}$ is defined in Eq. (6.6) and $\mathcal{L}_{\text{condition}}^r$ for each r are formulated in the same way as in Eq. (6.5). All the parameters are optimized in the same way as in Eq. (6.8) to Eq. (6.12). Note that better performance may be obtained when the condition losses have different combination weights. However, we just equally add them together in Eq. (6.13) to avoid tuning.

6.5 Experiments

To compare directly with the results in [25], we use exactly the same experiment setup as in [25]. We perform unsupervised adaptation of a clean long short-term memory (LSTM)- recurrent neural networks (RNN) [140] acoustic model trained with 375 hours of Microsoft Cortana voice assistant data to the noisy CHiME-3 dataset [132] using T/S and adversarial T/S learning. The CHiME-3 dataset incorporates Wall Street Journal (WSJ) corpus sentences spoken in challenging noisy environments, recorded using a 6-channel tablet. The real far-field noisy speech from the 5th microphone channel in CHiME-3 development data set is used for testing. A standard WSJ 5K word 3-gram language model (LM) is used for decoding.

The clean acoustic model is an LSTM-RNN trained with cross-entropy criterion. We extract 80-dimensional input log Mel filterbank feature as the input to the acoustic model. The LSTM has 4 hidden layers with 1024 units in each layer. A 512-dimensional projection layer is inserted on

top each hidden layer to reduce the number of parameters. The output layer has 5976 output units predicting senone posteriors. A WER of 23.16% is achieved when evaluating the clean model on the test data. The clean acoustic model is used as the teacher model in the following experiments.

6.5.1 T/S Learning for Unsupervised Adaptation

We first use parallel data consisting of 9137 pairs of clean and noisy utterances in the CHiME-3 training set (named as "clean-noisy") as the adaptation data for T/S learning. In order to let the student model be invariant to environments, the training data for student model should include both clean and noisy data. Therefore, We extend the original T/S learning work in [25] by also including 9137 pairs of the clean and clean utterances in CHiME-3 (named as "clean-clean") for adaptation. By perform T/S learning with both the "clean-noisy" and "clean-clean" parallel data, the learned student model should perform well on both the clean and noisy data because it will approach the behavior of teacher model on clean data no matter it is presented with clean or noisy data.

The unadapted Cortana model has 6.96% WER on the clean test set. After T/S learning with both the "clean-noisy" and "clean-clean" parallel data, the student model has 6.99% WER on the clean test. As the focus of this study is to improve T/S adaptation on noisy test data, we will only report results with the CHiME-3 real noisy channel 5 test set. The WER results on the noisy channel 5 test set of T/S learning are shown in Table 6.1. The T/S learning achieves 13.88% and 13.56% average WERs when adapted to "clean-noisy" and "clean-noisy & clean-clean" respectively, which are 40.05% and 41.45% relative improvements over the unadapted clean model. Note that our experimental setup does not achieve the state-of-the-art performance on CHiME-3 dataset (e.g., we did not perform beamforming, sequence training or use RNN LM for decoding.) since our goal is to simply verify the effectiveness of adversarial T/S learning in achieving condition-robust unsupervised adaptation.

6.5.2 Adversarial T/S Learning for Environment-Robust Unsupervised Adaptation

We adapt the clean acoustic model with the "clean-noisy & clean-clean" parallel data using adversarial T/S learning so that the resulting student model is environment invariant. The feature extractor M_f is initialized with the first N_h hidden layers of the clean student LSTM and the senone classifier

Table 6.1: The WER (%) performance of unadapted, T/S learning adapted LSTM acoustic models for robust ASR on the real noisy channel 5 test set of CHiME-3.

System	Adaptation Data	BUS	CAF	PED	STR	Avg.
Unadapted	-	27.93	24.93	18.53	21.38	23.16
T/S	clean-noisy	16.00	15.24	11.27	13.07	13.88
	clean-noisy, clean-clean	15.96	14.32	11.00	13.04	13.56

Table 6.2: The WER (%) performance of adversarial T/S learning adapted LSTM acoustic models for robust ASR on the real noisy channel 5 test set of CHiME-3. The adaptation data consists of "clean-noisy" and "clean-clean".

System	Conditions	BUS	CAF	PED	STR	Avg.
Adversarial T/S	2 environments	15.24	13.95	10.71	12.76	13.15
	6 environments	15.58	13.23	10.65	13.10	13.12
	87 speakers	14.97	13.63	10.84	12.24	12.90
	87 speakers, 6 environments	15.38	13.08	10.47	12.45	**12.83**

M_y is initialized with the last $(4 - N_h)$ hidden layers plus the output layer of the clean LSTM. N_h indicates the position of the deep feature in the student network. The condition classifier network M_c has 2 hidden layers with 512 units in each hidden layer.

To achieve environment-robust unsupervised adaptation, the condition classifier network M_c is designed to predict the posteriors of different environments at the output layer. As the adaptation data comes from both the clean and noisy environments, we first use an M_c with 2 output units to predict these two environments. As shown in Table 6.2, the adversarial T/S learning with 2-environment condition classifier achieves 13.15% WER, which are 43.22% and 3.02% relatively improved over the unadapted and T/S learning adapted models respectively. The N_h and λ are fixed at 4 and 5.0 respectively in all our experiments.

However, the noisy data in CHiME-3 is recorded under 5 different noisy environments, i.e, on buses (BUS), in cafes (CAF), in pedestrian areas (PED), at street junctions (STR) and in booth (BTH). To mitigate the speech variations among these environments, we further use an M_c with 6 output units to predict the posteriors of the 5 noisy and 1 clean environments. The WER with 6-environment condition classier is 13.12% which achieves 43.35% and 3.24% relative improvement over the unadapted and T/S learning adapted baseline models respectively. The increasing amount of noisy environments to be normalized through adversarial T/S learning lead to very limited WER

improvement which indicates that the differences among various kinds of noises are not significant enough in CHiME-3 as compared to the distinctions between clean and noisy data.

6.5.3 Adversarial T/S Learning for Speaker-Robust Unsupervised Adaptation

To achieve speaker-robust unsupervised adaptation, M_c is designed to predict the posteriors of different speaker identities at the output layer. The 7138 simulated and 1999 real noisy utterances in CHiME-3 training set are dictated by 83 and 4 different speakers respectively and the 9137 clean utterances are read by the same speakers. In speaker-robust adversarial T/S adaptation, an M_c with 87 output units are used to predict the posteriors of the 87 speakers. From Table 6.2, the adversarial T/S learning with 87-speaker condition classifier achieves 12.90% WER, which is 44.30% and 4.87% relative improvement over the unadapted and T/S adapted baseline models respectively. Larger WER improvement is achieved by speaker-robust unsupervised adaptation than the environment-robust methods. This is because T/S learning itself is able to reduce the environment variability through directly teaching the noisy student model with the senone posteriors from the clean data, which limits the space of improvement that environment-robust adversarial T/S learning can obtain.

6.5.4 Multi-factorial Adversarial T/S Learning for Unsupervised Adaptation

Speaker and environment robustness can be achieved simultaneously in unsupervised adaptation through MFA T/S learning, in which we need two condition classifiers: M_c^1 predicts the posteriors of 87 speakers and M_c^2 predicts the posteriors of 1 clean and 5 noisy environments in the adaptation data. From Table 6.2, the MFA T/S learning achieves 12.83% WER, which is 44.60% and 5.38% relative improvement over unadapted and T/S baseline models. The MFA T/S achieves lower WER than all the unifactorial adversarial T/S systems because it addresses the variations caused by all kinds of factors.

6.6 Conclusions

In this chapter, adversarial T/S learning is proposed to adapt a clean acoustic model to highly mismatched multi-conditional noisy data in a purely unsupervised fashion. To suppress the condition variability in speech signal and achieve robust adaptation, a student acoustic model and a condition

classifier are jointly optimized to minimize the KL divergence between the output distributions of the teacher and student models while simultaneously mini-maximize condition classification loss. We further propose the MFA T/S learning where multiple condition classifiers are introduced to reduce the condition variabilities caused by different factors. The proposed methods requires only the unlabeled parallel data for domain adaptation.

For environment adaptation on CHiME-3 real noisy channel 5 dataset, T/S learning gets 41.45% relative WER reduction from the clean-trained acoustic model. Adversarial T/S learning with environment and speaker classifiers achieves 3.24% and 4.87% relative WER improvements over the strong T/S learning model, respectively. MFA T/S achieves 5.38% relative WER improvement over the same baseline. On top of T/S learning, reducing speaker variability proves to be more effective than reducing environment variability T/S learning on CHiME-3 dataset because T/S learning already addresses most environment mismatch issues. Simultaneously decreasing the condition variability in multiple factors can further slightly improve the ASR performance.

The adversarial T/S learning was verified its effectiveness with a relatively small CHiME-3 task. We recently developed a far-field speaker system using thousands of hours data with T/S learning [141]. We are now currently applying the proposed adversarial T/S learning to further improve our far-field speaker system.

CHAPTER 7

DOMAIN SEPARATION NETWORKS FOR UNSUPERVISED ADAPTATION

7.1 Introduction

In recent years, advances in deep learning have led to remarkable performance boost in ASR [142, 136, 137, 116, 138, 117]. However, ASR systems still suffer from large performance degradation when acoustic mismatch exists between the training and test conditions [67, 139]. Many factors contribute to the mismatch, such as variation in environment noises, channels and speaker characteristics. Domain adaptation is an effective way to address this limitation, in which the acoustic model parameters or input features are adjusted to compensate for the mismatch.

One difficulty with domain adaptation is that available data from the target domain is usually limited, in which case the acoustic model can be easily overfitted. To address this issue, regularization-based approaches are proposed in [68, 70, 120, 143] to regularize the neuron output distributions or the model parameters. In [75, 76], transformation-based approaches are introduced to reduce the number of learnable parameters. In [123, 77, 78], the trainable parameters are further reduced by singular value decomposition of weight matrices of a neural network. Although these methods utilize the limited data from the target domain, they still require labelling for the adaptation data and can only be used in supervised adaptation.

Unsupervised domain adaptation is necessary when human labelling of the target domain data is unavailable. It has become an important topic with the rapid increase of the amount of untranscribed speech data for which the human annotation is expensive. Pawel et al. proposed to learn the contribution of hidden units by additional amplitude parameters [121] and differential pooling [144]. Recently, Wang et al. proposed to adjust the linear transformation learned by batch normalized acoustic model in [145]. Although these methods lead to increased performance in the ASR task when no labels are available for the adaptation data, they still rely on the senone (tri-phone state) alignments against the unlabeled adaptation data through first pass decoding. The first pass decoding result is unreliable when the mismatch between the training and test conditions is signifi-

cant. It is also time-consuming and can be hardly applied to huge amount of adaptation data. There are even situations when decoding adaptation data is not allowed because of the privacy agreement signed with the speakers. These methods depending on the first pass decoding of the unlabeled adaptation data is sometimes called "semi-supervised" adaptation in literature.

The goal of our study is to achieve *purely* unsupervised domain adaptation *without* any exposure to the labels or the decoding results of the adaptation data in the target domain. In [73] we show that the source-domain model can be effectively adapted without any transcription by using teacher-student (T/S) learning [74], in which the posterior probabilities generated by the source-domain model can be used in lieu of labels to train the target-domain model. However, T/S learning relies on the availability of parallel unlabeled data which can be usually simulated. However, if parallel data is not available, we cannot use T/S learning for model adaptation. In this study, we are exploring the solution to domain adaptation without parallel data and without transcription. Recently, adversarial training has become a very hot topic in deep learning because of its great success in estimating generative models [22]. It was first applied to the area of unsupervised domain adaptation by Ganin et al. in [23] in a form of multi-task learning. In their work, the unsupervised adaptation is achieved by learning deep intermediate representations that are both discriminative for the main task (image classification) on the source domain and invariant with respect to mismatch between source and target domains. The domain invariance is achieved by the adversarial training of the domain classification objective functions. This can be easily implemented by augmenting any feed-forward models with a few standard layers and a *gradient reversal layer (GRL)*. This GRL approach has been applied to acoustic models for unsupervised adaptation in [27] and for increasing noise robustness in [26, 83]. Improved ASR performance is achieved in both scenarios.

However, the GRL method focuses only on learning a domain-invariant representation, ignoring the unique characteristics of each domain, which could also be informative. Inspired by this, Bousmailis et al. [29] proposed the *domain separation networks (DSNs)* to separate the deep representation of each training sample into two parts: one private component that is unique to its domain and one shared component that is invariant to the domain shift. In this chapter, we propose to apply DSN for unsupervised domain adaptation on a DNN-hidden Markov model (HMM) acoustic model, aiming to increase the noise robustness in speech recognition. In the proposed framework,

the shared component is learned to be both senone-discriminative and domain-invariant through adversarial multi-task training of a shared component extractor and a domain classifier. The private component is trained to be orthogonal with the shared component to implicitly increase the degree of domain-invariance of the shared component. A reconstructor DNN is used to reconstruct the original speech feature from the private and shared components, serving for regularization. The proposed method achieves 11.08% relative WER improvement over the GRL training approach for robust ASR on the CHiME-3 dataset.

7.2 Domain Separation Networks

In the *purely* unsupervised domain adaptation task, we only have access to a sequence of speech frames $X^s = \{x_1^s, \ldots, x_{N_s}^s\}$ from the source domain distribution, a sequence of senone labels $Y^s = \{y_1^s, \ldots, y_{N_s}^s\}$ aligned with source data X^s and a sequence of speech frames $X^t = \{x_1^t, \ldots, x_{N_t}^t\}$ from a target domain distribution. Senone labels or other types of transcription are *not* available for the target speech sequence X^t.

When applying domain separation networks (DSNs) to the unsupervised adaptation task, our goal is to learn the shared (or common) component extractor DNN M_c that maps an input speech frame x^s from source domain or x^t from target domain to a *domain-invariant* shared component f_c^s or f_c^t respectively. At the same time, learn a senone classifier DNN M_y that maps the shared component f_c^s from the source domain to the correct senone label y^s.

To achieve this, we first perform adversarial training of the domain classifier DNN M_d that maps the shared component f_c^s or f_c^t to its domain label d^s or d^t, while simultaneously minimizing the senone classificaton loss of M_y given shared component f_c^s from the source domain to ensure the *senone-dicriminativeness* of f_c^s.

For the source or the target domain, we extract the source or the target private component f_p^s or f_p^t that is unique to the source or the target domain through a source or a target private component extractor M_p^s or M_p^t. The shared and private components of the same domain are trained to be orthogonal to each other to further enhance the degree of domain-invariance of the shared components. The extracted shared and private components of each speech frame are concatenated and fed as the input of a reconstructor M_r to reconstruct the input speech frame x^s or x^t.

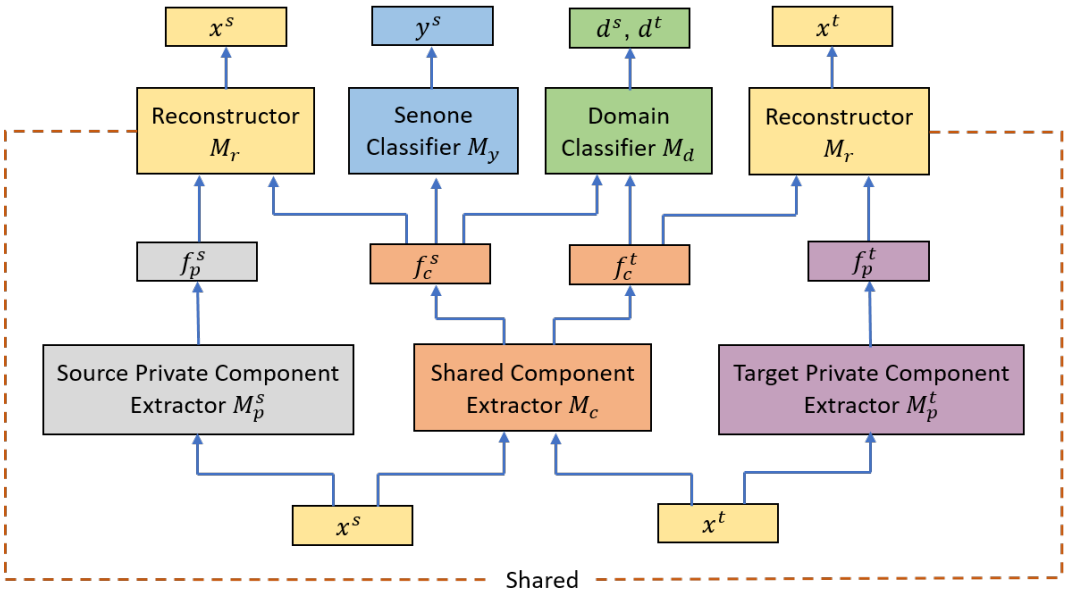


Figure 7.1: The architecture of domain separation networks.

The architecture of DSN is shown in Fig. 7.1, in which all the sub-networks are jointly optimized using SGD. The optimized shared component extractor M^c and senone classifier M_y form the adapted acoustic model for subsequent robust speech recognition.

7.2.1 Deep Neural Networks Acoustic Model

The shared component extractor M_c and senone predictor of the DSN are initialized from an DNN-HMM acoustic model. The DNN-HMM acoustic model is trained with labeled speech data (X^s, Y^s) from the source domain. The senone-level alignment Y_s is generated by a well-trained GMM-HMM system.

Each output unit of the DNN acoustic model corresponds to one of the senones in the set $\mathcal{Q}$. The output unit for senone $q \in \mathcal{Q}$ is the posterior probability $p(q|x_n^s)$ obtained by a softmax function.

7.2.2 Shared Component Extraction with Adversarial Training

The well-trained acoustic model DNN in Section 7.2.1 can be decomposed into two parts: a share component extractor M_c with parameters θ_c and a senone classifier M_y with parameters θ_y. An input speech frame from source domain x^s is first mapped by the M_c to a K-dimensional shared

component $f_c^s \in \mathcal{R}^K$. f_c^s is then mapped to the senone label posteriors by a senone classifier M_y with parameters θ_y as follows.

$$M_y(f_c^s) = M_y(M_c(x_i^s)) = p(\hat{y}_n^s = q|x_i^s; \theta_c, \theta_y) \tag{7.1}$$

where $\hat{y}_i^s$ denotes the predicted senone label for source frame x_i^s and $q \in \mathcal{Q}$.

The domain classifier DNN M_d with parameters θ_d takes the shared component from source domain f_c^s or target domain f_c^t as the input to predict the two-dimensional domain label posteriors as follows (the 1st and 2nd output units stand for the source and target domains respectively).

$$M_d(M_c(x_i^s)) = p(\hat{d}_i^s = a|x_i^s; \theta_c, \theta_d),\ a \in \{1, 2\} \tag{7.2}$$

$$M_d(M_c(x_j^t)) = p(\hat{d}_j^t = a|x_j^t; \theta_c, \theta_d),\ a \in \{1, 2\} \tag{7.3}$$

where $\hat{d}_i^s$ and $\hat{d}_j^t$ denote the predicted domain labels for the source frame x_i^s and the target frame x_j^t respectively.

In order to adapt the source domain acoustic model (i.e., M_c and M_y) to the *unlabeled* data from target domain, we want to make the distribution of the source domain shared component $P(f_c^s) = P(M_c(x^s))$ as close to that of the target domain $P(f_c^t) = P(M_c(x^t))$ as possible. In other words, we want to make the shared component domain-invariant. This can be realized by adversarial training, in which we adjust the parameters θ_c of shared component extractor to *maximize* the loss of the domain classifier $\mathcal{L}_{\text{domain}}^c(\theta_c)$ below while adjusting the parameters θ_d to *minimize* the loss of the domain classifier $\mathcal{L}_{\text{domain}}^d(\theta_d)$ below.

$$\mathcal{L}_{\text{domain}}^d(\theta_d) = -\sum_i^{N_s} \log p(\hat{d}_i^s = 1|x_i^s; \theta_d) - \sum_j^{N_t} \log p(\hat{d}_j^t = 2|x_j^t; \theta_d) \tag{7.4}$$

$$\mathcal{L}_{\text{domain}}^c(\theta_c) = -\sum_i^{N_s} \log p(\hat{d}_i^s = 1|x_i^s; \theta_c) - \sum_j^{N_t} \log p(\hat{d}_j^t = 2|x_j^t; \theta_c) \tag{7.5}$$

This minimax competition will first increase the capability of both the shared component extractor

and the domain classifier and will eventually converge to the point where the shared component extractor generates extremely confusing representations that domain classifier is unable to distinguish (i.e., domain-invariant).

Simultaneously, we minimize the loss of the senone classifier below to ensure the domain-invariant shared component f_c^s is also discriminative to senones.

$$\mathcal{L}_{\text{senone}}(\theta_c, \theta_y) = -\sum_{i}^{N_s} \log p(y_i^s | x_i^s; \theta_y, \theta_c) \tag{7.6}$$

Since the adversarial training of the domain classifier M_d and shared component extractor M_c has made the distribution of the target domain shared-component f_c^t as close to that of f_c^s as possible, the f_c^t is also senone-discriminative and will lead to minimized senone classification error given optimized M_y. Because of the domain-invariant property, good adaptation performance can be achieved when the target domain data goes through the network.

7.2.3 Private Components Extraction

To further increase the degree of domain-invariance of the shared components, we explicitly model the private component that is unique to each domain by a private component extractor DNN M_p parameterized by θ_p. M_p^s and M_p^t map the source frame x^s and the target frame x^t to hidden representations $f_p^s = M_p^s(x^s)$ and $f_p^t = M_p^t(x^t)$ which are the private components of the source and target domains respectively. The private component for each domain is trained to be orthogonal to the shared component by minimizing the difference loss below.

$$\mathcal{L}_{\text{diff}}(\theta_c, \theta_p^s, \theta_p^t) = \left|\left|\sum_{i}^{N_s} M_c(x_i^s) M_p^s(x_i^s)^\top\right|\right|_F^2 + \left|\left|\sum_{j}^{N_t} M_c(x_j^t) M_p^t(x_j^t)^\top\right|\right|_F^2 \tag{7.7}$$

where $||\cdot||_F^2$ is the squared Frobenius norm. All the vectors are assumed to be column-wise.

As a regularization term, the predicted shared and private components are then concatenated and fed into a reconstructor DNN M_r with parameters θ_r to recover the input speech frames x^s and x^t from both source and target domains respectively. The reconstructor is trained to minimize the

mean square error based reconstruction loss as follows:

$$\mathcal{L}_{\text{recon}}(\theta_c, \theta_p^s, \theta_p^t, \theta_r) = \sum_i^{N_s} ||\hat{x}_i^s - x_i^s||_2^2 + \sum_j^{N_t} ||\hat{x}_j^t - x_j^t||_2^2 \tag{7.8}$$

$$\hat{x}_i^s = M_r([M_c(x_i^s), M_p^s(x_i^s)]) \tag{7.9}$$

$$\hat{x}_j^t = M_r([M_c(x_j^t), M_p^t(x_j^t)]) \tag{7.10}$$

where $[\cdot, \cdot]$ denotes concatenation of two vectors.

The total loss of DSN is formulated as follows and is jointly optimized with respect to the parameters.

$$\begin{aligned}\mathcal{L}_{\text{total}}(\theta_y, \theta_c, \theta_d, \theta_p^s, \theta_p^t, \theta_r) &= \mathcal{L}_{\text{senone}}(\theta_c, \theta_y) + \mathcal{L}_{\text{domain}}^d(\theta_d) \\ &- \alpha\mathcal{L}_{\text{domain}}^c(\theta_c) + \beta\mathcal{L}_{\text{diff}}(\theta_c, \theta_p^s, \theta_p^t) + \gamma\mathcal{L}_{\text{recon}}(\theta_c, \theta_p^s, \theta_p^t, \theta_r)\end{aligned} \tag{7.11}$$

$$\min_{\theta_y, \theta_c, \theta_d, \theta_p^s, \theta_p^t, \theta_r} \mathcal{L}_{\text{total}}(\theta_y, \theta_c, \theta_d, \theta_p^s, \theta_p^t, \theta_r) \tag{7.12}$$

All the parameters of DSN are jointly optimized through backprogation with stochastic gradient descent (SGD) as follows:

$$\theta_c \leftarrow \theta_c - \mu\left[\frac{\partial\mathcal{L}_{\text{senone}}}{\partial\theta_c} - \alpha\frac{\partial\mathcal{L}_{\text{domain}}^c}{\partial\theta_c} + \beta\frac{\partial\mathcal{L}_{\text{diff}}}{\partial\theta_c} + \gamma\frac{\partial\mathcal{L}_{\text{recon}}}{\partial\theta_c}\right] \tag{7.13}$$

$$\theta_d \leftarrow \theta_d - \mu\frac{\partial\mathcal{L}_{\text{domain}}^d}{\partial\theta_d} \tag{7.14}$$

$$\theta_y \leftarrow \theta_y - \mu\frac{\partial\mathcal{L}_{\text{senone}}}{\partial\theta_y} \tag{7.15}$$

$$\theta_p^s \leftarrow \theta_p^s - \mu\left[\beta\frac{\partial\mathcal{L}_{\text{diff}}}{\partial\theta_p^s} + \gamma\frac{\partial\mathcal{L}_{\text{recon}}}{\partial\theta_p^s}\right] \tag{7.16}$$

$$\theta_p^t \leftarrow \theta_p^t - \mu\left[\beta\frac{\partial\mathcal{L}_{\text{diff}}}{\partial\theta_p^t} + \gamma\frac{\partial\mathcal{L}_{\text{recon}}}{\partial\theta_p^t}\right] \tag{7.17}$$

$$\theta_r \leftarrow \theta_r - \mu\frac{\partial\mathcal{L}_{\text{recon}}}{\partial\theta_r} \tag{7.18}$$

Note that the negative coefficient $-\alpha$ in Eq. (7.13) induces reversed gradient that maximizes the domain classification loss in Eq. (7.5) and makes the shared components domain-invariant. Without the reversal gradient, SGD would make representations different across domains in order

to minimize Eq. (7.4). For easy implementation, GRL is introduced in [23], which acts as an identity transform in the forward pass and multiplies the gradient by $-\alpha$ during the backward pass.

The optimized shared component extractor M_c and senone classifier M_y form the adapted acoustic model for robust speech recognition.

7.3 Experiments

In this chapter, we perform the *pure* unsupervised environment adaptation of a deep FNN-HMM acoustic model with domain separation networks for robust speech recognition on CHiME-3 dataset.

7.3.1 Dataset Description

The CHiME-3 dataset is released with the 3rd CHiME speech Separation and Recognition Challenge [132], which incorporates the Wall Street Journal corpus sentences spoken in challenging noisy environments, recorded using a 6-channel tablet based microphone array. CHiME-3 dataset consists of both real and simulated data. The real speech data was recorded in four real noisy environments (on buses (BUS), in cafés (CAF), in pedestrian areas (PED), and at street junctions (STR)). To generate the simulated data, the clean speech is first convoluted with the estimated impulse response of the environment and then mixed with the background noise separately recorded in that environment [133]. The noisy training data consists of 1600 real noisy utterances from 4 speakers, and 7138 simulated noisy utterances from 83 speakers in the WSJ0 SI-84 training set recorded in 4 noisy environments. There are 3280 utterances in the development set including 410 real and 410 simulated utterances for each of the 4 environments. There are 2640 utterances in the test set including 330 real and 330 simulated utterances for each of the 4 environments. The speakers in training set, development set and the test set are mutually different (i.e., 12 different speakers in the CHiME-3 dataset). The training, development and test data sets are all recorded in 6 different channels.

8738 clean utterances corresponding to the 8738 noisy training utterances in the CHiME-3 dataset are selected from the WSJ0 SI-85 training set to form the clean training data in our experiments. WSJ 5K word 3-gram language model is used for decoding.

7.3.2 Baseline System

In the baseline system, we first train a deep FNN-HMM acoustic model with clean speech and then adapt the clean acoustic model to noisy data using GRL unsupervised adaptation in [23]. Hence, the source domain is with clean speech while the target domain is with noisy speech.

The 29-dimensional log Mel filterbank features together with 1st and 2nd order delta features (totally 87-dimensional) for both the clean and noisy utterances are extracted by following the process in [134]. Each frame is spliced together with 5 left and 5 right context frames to form a 957-dimensional feature. The spliced features are fed as the input of the FNN after global mean and variance normalization. The FNN has 7 hidden layers with 2048 hidden units for each layer. The output layer of the FNN has 3012 output units corresponding to 3012 senone labels. Senone-level forced alignment of the clean data is generated using a GMM-HMM system. The FNN is first trained with 8738 clean training utterances in CHiME-3 and the alignment to minimize the cross entropy loss and then tested with simulation and real development data of CHiME-3.

The FNN well-trained with clean data is then adapted to the 8738 noisy utterances from Channel 5 using GRL method. No senone alignment of the noisy adaptation data is used for the unsupervised adaptation. The feature extractor is initialized with the first 4 hidden layers of the clean FNN and the senone classifier is initialized with the last 3 hidden layers plus the output layers of the clean FNN. The domain classifier is a deep FNN with two hidden layers and each hidden layer has 512 hidden units. The output layer of the domain classifier has 2 output units representing source and target domains. The 2048 hidden units of the 4^{th} hidden layer of the FNN acoustic model is fed as the input to the domain classifier. A GRL is inserted in between the deep representation and the domain classifier for easy implementation. The GRL adapted system is tested on real and simulation noisy development data in CHiME-3 dataset.

7.3.3 Domain Separation Networks for Unsupervised Adaptation

We adapt the clean FNN acoustic model trained in Section 7.3.2 to the 8738 noisy utterances using DSN. No senone alignment of the noisy adaptation data is used for the unsupervised adaptation.

The DSN is implemented with CNTK 2.0 Toolkit [146]. The shared component extractor M_c is

initialized with the first N_h hidden layers of the clean FNN and the senone classifier M_y is initialized with the last $(7-N_h)$ hidden layers plus the output layer of the clean FNN. N_h indicates the position of shared component in the FNN acoustic model and ranges from 3 to 7 in our experiments. The domain classifier M_d of the DSN has exactly the same architecture as that of the GRL.

The private component extractors M_p^s and M_p^t for the clean and noisy domains are both feed-forward FNNs with 3 hidden layers and each hidden layer has 512 hidden units. The output layers of both M_p^s and M_p^t have 2048 output units. The reconstructor M_r is a FNN with 3 hidden layers and each hidden layer has 512 hidden units. The output layer of the M_r has 957 output units with no non-linear activation functions to reconstruct the spliced input features.

The activation functions for the hidden units of M_c is sigmoid. The activation functions for hidden units of M_p^s, M_p^t, M_d and M_r are rectified linear units (ReLU). The activation functions for the output units of M_c and M_d are softmax. The activation functions for the output units of M_p^s, M_p^t are sigmoid. All the sub-networks except for M_y and M_c are randomly initialized. The learning rate is fixed at 5×10^{-5} throughout the experiments. The adapted DSN is tested on real and simulation development data in CHiME-3 Dataset.

Table 7.1: The WER (%) performance of unadapted acoustic model, GRL and DSN adapted acoustic models for robust ASR on real and simulated development set of CHiME-3.

System	Data	BUS	CAF	PED	STR	Avg.
Clean	Real	36.25	31.78	22.76	27.18	29.44
	Simu	26.89	37.74	24.38	26.76	28.94
GRL	Real	35.93	28.24	19.58	25.16	27.16
	Simu	26.14	34.68	22.01	25.83	27.16
DSN	Real	32.62	23.48	17.29	23.46	**24.15**
	Simu	23.38	30.39	19.51	22.01	**23.82**

7.3.4 Result Analysis

Table 7.1 shows the WER performance of clean, GRL adapted and DSN adapted FNN acoustic models for ASR. The clean FNN achieves 29.44% and 28.25% WERs on the real and simulated development data respectively. The GRL adapted acoustic model achieves 27.16% and 27.16% WERs on the real and simulated development data. The best WER performance for DSN adapted acoustic model are 24.15% and 23.82% on real and simulated development data, which achieve

Table 7.2: The ASR WERs (%) for the DSN adapted acoustic models with respect to N_h reversal gradient coefficient α on the real development set of CHiME-3.

N_h	Reversal Gradient Coefficient α									
	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	Avg.
3	27.2	26.24	25.76	26.51	26.12	26.92	26.65	26.91	27.41	26.64
4	26.56	26.08	25.75	25.99	25.88	26.76	27.0	27.13	27.74	26.54
5	26.53	25.9	26.07	25.88	25.72	26.17	27.36	26.67	27.37	26.41
6	25.77	25.17	25.06	24.94	24.6	25.19	25.53	25.42	25.93	25.29
7	25.99	25.5	24.73	24.43	25.08	24.53	25.07	24.15	24.29	**24.86**

11.08% and 12.30% relative improvement over the GRL baseline system and achieve 17.97 % and 17.69% relative improvement over the unadapted acoustic model. The best WERs are achieved when $N_h = 7$ and $\alpha = 8.0$. By comparing the GRL and DSN performance at $N_h = 4$, we observe that the introduction of private components and reconstructor lead to 5.1% relative improvements in WER.

We investigate the impact of shared component position N_h and the reversal gradient coefficient α on the WER performance as in Table 7.2. We observe that the WER decreases with the growth of N_h, which is reasonable as the higher hidden representation of a well-trained DNN acoustic model is inherently more senone-discriminative and domain-invariant than the lower layers and can serve as a better initialization for the DSN unsupervised adaptation.

7.4 Conclusions

In this chapter, we investigate the domain adaptation of the DNN acoustic model by using domain separation networks. Different from the conventional supervised, semi-supervised and T/S adaptation approaches, DSN is capable of adapting the acoustic model to the adaptation data without any exposure to its transcription, decoded lattices or unlabeled parallel data from the source domain. The shared component between source and target domains extracted by DSN through adversarial multi-task training is both domain-invariant and senone-discriminative. The extraction of private component that is unique to each domain significantly improves the degree of domain-invariance and the ASR performance.

When evaluated on the CHiME-3 dataset for environment adaption task, the DSN achieves

11.08% and 17.97% relative WER improvement over the GRL baseline system and the unadapted FNN acoustic model. The WER decreases when higher hidden representations of the DNN acoustic model are used as the initial shared component. The WER first decreases and then increases with the growth of the reversal gradient coefficient.

CHAPTER 8

ADAPTIVE BEAMFORMING NETWORKS FOR MULTICHANNEL ROBUST SPEECH RECOGNITION

8.1 Introduction

Although extraordinary performance has been achieved in ASR with the advent of DNNs [1, 2], the performance still degrades dramatically in noisy and far-field situations [147, 133]. To achieve robust speech recognition, multiple microphones can be used to enhance the speech signal, reduce the effects of noise and reverberation, and improve the ASR performance. In this scenario, an essential step of the ASR front-end processing is multichannel filtering, or *beamforming*, which steers a spatial sensitivity region, or "beam," in the direction of the target source, and inserts spatial suppression regions, or "nulls," in the directions corresponding to noise and other interference.

Delay-and-sum (DAS) beamforming is widely used for multichannel signal processing [148], in which the multichannel inputs of an microphone array are delayed to be aligned in time and then summed up to be a single channel signal. The signal from the target direction is enhanced and the noises and interferences coming from other directions are attenuated. Filter-and-sum beamforming applies filters to the input channels before summing them up [149]. As a filter-and-sum beamformer, minimum variance distortionless response (MVDR) minimizes the estimated noise level under the condition of no distortion in the desired signal [150]. In addition, the generalized eigenvalue (GEV) beamforming with blind analytic normalization controls the desired signal by a single channel post-filter which requires no knowledge about the array geometry, the impulse response from source to microphone array or the direction-of-arrival [151]. In [152] and [153], speech or noise masks are predicted through LSTM for MVDR and GEV respectively.

Although these methods have achieved good performance in beamforming, their goal is to optimize only the signal-level objective (e.g., SNR). In order to achieve robust speech recognition, it is more important to jointly optimize beamforming and acoustic model with the objective of maximizing the ASR performance. In [154], the parameters of a frequency-domain beamformer are first

estimated by a DNN based on the generalized cross correlation between microphones. Conventional features are extracted from the beamformed signal before passing through a second DNN for acoustic modeling. Instead of filtering in the frequency domain, [155] performs spatial and spectral filtering through time-domain convolution over raw waveform. The output feature is then passed to a convolutional LSTM DNN (CLDNN) acoustic model to predict the context-dependent state output targets. In [156], the beamforming and frequency decomposition are factorized into separate layers in the network. These approaches assume that the speaker position and the environment are fixed and estimate constant filter coefficients for either beamforming or spatial and spectral filtering.

However, in real noisy and far-field scenarios, as the position of the source (speaker), noise and room impulse response keep changing, the time-invariant filter coefficients estimated by these neural networks may fail to robustly enhance the target signal. Therefore, we propose to adaptively estimate the beamforming filter coefficients at each time frame using an LSTM to deal with any possible changes of the source, noise or channel conditions. The enhanced signal is generated by applying these time-variant filter coefficients to the short-time Fourier transform (STFT) of the array signals. Log filter-bank like features are obtained from the enhanced signal and then passed to a deep LSTM acoustic model to predict the senone posterior. The LSTM beamforming network and the LSTM acoustic model are jointly trained using truncated back-propagation through time (BPTT) with a cross-entropy objective. STFT coefficients of the array signals are used as the input of the beamforming network. In ASR systems of [157, 158], the speech signal is enhanced by NMF and LSTM before fed into the acoustic model. But speech enhancement module and the acoustic model are not jointly optimized to minimize the WER and the input is only single channel signal.

Previous work [159] has shown that the speech separation performance can be improved by incorporating the speech recognition alignment information within the speech enhancement framework. Inspired by this, we feed the units of the top hidden layer of the LSTM acoustic model at the previous time step back as an auxiliary input to the beamforming network to predict the current filter coefficients. Note that our work is different from [160] in that: (1) we perform adaptive beamforming over 5 different input channels, but their system works only on 2 input channels; (2) our adaptive LSTM beamformer predicts only the frequency domain filter coefficients and performs frequency domain filter-and-sum over STFT coefficients, while their work majorly focuses on the

time-domain filtering with raw waveforms as the input; (3) the log Mel filter bank like features are generated with fixed log Mel transform over the beamformed STFT coefficients for acoustic modeling in our work, while time/frequency domain convolution is performed with trainable parameters on the beamformed features in their work; (4) no additional gate modulation is applied to the feedback to reduce the system complexity for our much smaller dataset. In the experiments, we show that this feedback captures high-level knowledge about the acoustic states and increases the performance. The experiments are conducted with the CHiME 3 dataset. The joint training of LSTM adaptive beamforming network and deep LSTM acoustic model achieves 7.75% absolute gain over the single channel signal on the real test data. The acoustic model feedback provides an extra gain of 0.22%.

8.2 LSTM Adaptive Beamforming

8.2.1 Adaptive Filter-and-Sum Beamforming

As a generalization of the delay-and-sum beamforming, filter-and-sum beamformer processes the signal from each microphone using a finite impulse response (FIR) filter before summing them up. In frequency domain, this operation can be written as:

$$\hat{x}_{t,f} = \sum_{m=1}^{M} g_{f,m} x_{t,f,m}, \tag{8.1}$$

where $x_{t,f,m} \in \mathcal{C}$ is the complex STFT coefficient for the time-frequency index (t, f) of the signal from channel m, $g_{f,m} \in \mathcal{C}$ is the beamforming filter coefficient and $\hat{x}_{t,f} \in \mathcal{C}$ is the complex STFT coefficient of the enhanced signal. In Eq. (8.1), $t = 1, \ldots, T, f = 1, \ldots, F$ and M, T, F are the numbers of microphones, time frames and frequencies. To cope with the time-variant source position and room impulse response, we make the filter coefficients time-dependent and propose the adaptive filter-and-sum beamforming:

$$\hat{x}_{t,f} = \sum_{m=1}^{M} g_{t,f,m} x_{t,f,m}, \tag{8.2}$$

where $g_{t,f,m} \in \mathcal{C}$ is time-variant complex filter coefficient.

8.2.2 Adaptive LSTM Beamforming Network

The LSTM network is a special kind of recurrent neural network (RNN) with purpose-built memory cells to store information [161]. The LSTM has been successfully applied to many different tasks [162, 163] due to its strong capability of learning long-term dependencies. The LSTM takes in an input sequence $x = \{x_1, \ldots, x_T\}$ and computes the hidden vector sequence $h = \{h_1, \ldots, h_T\}$ by iterating the equation below

$$h_t = \text{LSTM}(x_t, h_{t-1}) \tag{8.3}$$

We implement the LSTM in Eq. (8.3) with no peep hole connections.

In this chapter, we apply *real-value* LSTM to the adaptive filter-and-sum beamformer to predict the real and imaginary parts of the complex filter coefficients at time t and channel m. That is, we introduce the following real-value vectors for complex values $g_{t,f,m}$ and $x_{t,f,m}$ in Eq. (8.2):

$$\begin{aligned} g_{t,m} &\triangleq [\Re(g_{t,f,m}), \Im(g_{t,f,m})]_{f=1}^{F} \in \mathcal{R}^{2F} \\ x_t &\triangleq [\Re(x_{t,f,m}), \Im(x_{t,f,m})]_{f=1,m=1}^{F,M} \in \mathcal{R}^{2FM}. \end{aligned}$$

With this representation, the real-value LSTM predicts $g_{t,m}$ as follows:

$$p_t = W_{x,p} x_t \tag{8.4}$$

$$h_t = \text{LSTM}^{BF}(p_t, h_{t-1}) \tag{8.5}$$

$$g_{t,m} = \tanh(W_{h,m} h_t), \quad m = 1, \ldots, M, \tag{8.6}$$

where $W_{x,p}$ and $W_{h,m}$ are projection matrices. We use $\tanh(\cdot)$ function to limit the range of the filter coefficients within $[-1, 1]$.

The real and imaginary parts of the STFT coefficient $\hat{x}_{t,f}$ of the beamformed signal are gener-

ated by Eq. (8.2) as follows

$$\begin{cases} \Re(\hat{x}_{t,f}) & = \sum_{m=1}^{M} \Re(x_{t,f,m})\Re(g_{t,f,m}) - \Im(x_{t,f,m})\Im(g_{t,f,m}) \\ \Im(\hat{x}_{t,f}) & = \sum_{m=1}^{M} \Re(x_{t,f,m})\Im(g_{t,f,m}) + \Im(x_{t,f,m})\Re(g_{t,f,m}). \end{cases} \tag{8.7}$$

More sophisticated features can be extracted from the beamformed STFT coefficients and are passed to the LSTM acoustic model to predict the senone posterior. In our experiments, the log Mel filterbank like feature is generated from Eq. (8.7) by

$$z_t = \log\left(\text{Mel}(P_t)\right) \tag{8.8}$$

$$P_t = \left[\Re(\hat{x}_{t,f})^2 + \Im(\hat{x}_{t,f})^2\right]_{f=1}^{F} \in \mathcal{R}^F \tag{8.9}$$

where $\text{Mel}(\cdot)$ is the operation of Mel matrix multiplication, and P_t is F dimensional real-value vector of the power spectrum of the beamformed signal at time t. Global mean and variance normalization is applied to this log Mel filterbank like feature. Note that all operations in this section are performed with the *real-value* computation, and can be easily represented by a differentiable computational graph.

8.2.3 Deep LSTM Acoustic Model

Recently, LSTMs are shown to be more effective than DNNs [1, 2] and conventional RNNs [44, 45] for acoustic modeling as they are able to model temporal sequences and long-range dependencies more accurately than the others especially when the amount of training data is large. LSTM has been successfully applied in both the RNN-HMM hybrid systems [46, 47] and the end-to-end system [50, 51] with connectionist temporal classification objective [50, 164, 165, 52] or attention mechanism [51, 166, 167].

In this chapter, the deep LSTM-HMM hybrid system is utilized for acoustic modeling. A forced alignment is first generated by a GMM-HMM system and is then used as the frame-level acoustic targets which the LSTM attempts to classify. The LSTM is trained with cross-entropy objective function using truncated BPTT. In this chapter, to connect the deep LSTM with the adaptive LSTM

beamformer, we compute log Mel filterbank z_t from the beamformed STFT coefficients.

$$q_t = W_{z,p} z_t \tag{8.10}$$

$$s_t = \text{LSTM}^{AM}(q_t, s_{t-1}) \tag{8.11}$$

$$y_t = \text{softmax}(W_{s,y} s_t) \tag{8.12}$$

q_t is the projection of z_t into a high-dimensional space and y_t is the senone posterior.

8.2.4 Integrated Network of LSTM Adaptive Beamformer and Deep LSTM Acoustic Model

In order to achieve robust speech recognition by making use of multichannel speech signals, LSTM beamformer in Section 8.2.2 and the deep LSTM acoustic model in Section 8.2.3 need to be jointly optimized with the objective of maximizing the ASR performance. In other words, the beamforming LSTM needs to be concatenated with the LSTM acoustic model to form an integrated network that takes multichannel STFT coefficients as the input and produces senone posteriors as illustrated in Fig. 8.1. The deep LSTM has three hidden layers in our experiments but only one is shown here for simplicity.

To train the integrated LSTM network, we connect the beamforming network (8.2) – (8.6), log Mel filtering (8.8), and the acoustic model (8.10) – (8.12) as a single feed forward network, and back-propagate the gradient of the cross-entropy objective function through the network so that both the adaptive beamformer and the acoustic model are optimized for the ASR task by using multi-channel training data.

On top of that, we feed the hidden units of the top hidden layer of the deep LSTM acoustic model back to the input of the LSTM beamformer as the auxiliary feature to predict the filter coefficients at next time. By introducing the acoustic model feedback, the Eq. (8.5) is re-written as

$$h_t = \text{LSTM}^{BF}((p_t, s_{t-1}), h_{t-1}) \tag{8.13}$$

where (p_t, s_{t-1}) is the concatenation of the acoustic feedback from previous time s_{t-1} and the current projection p_t.

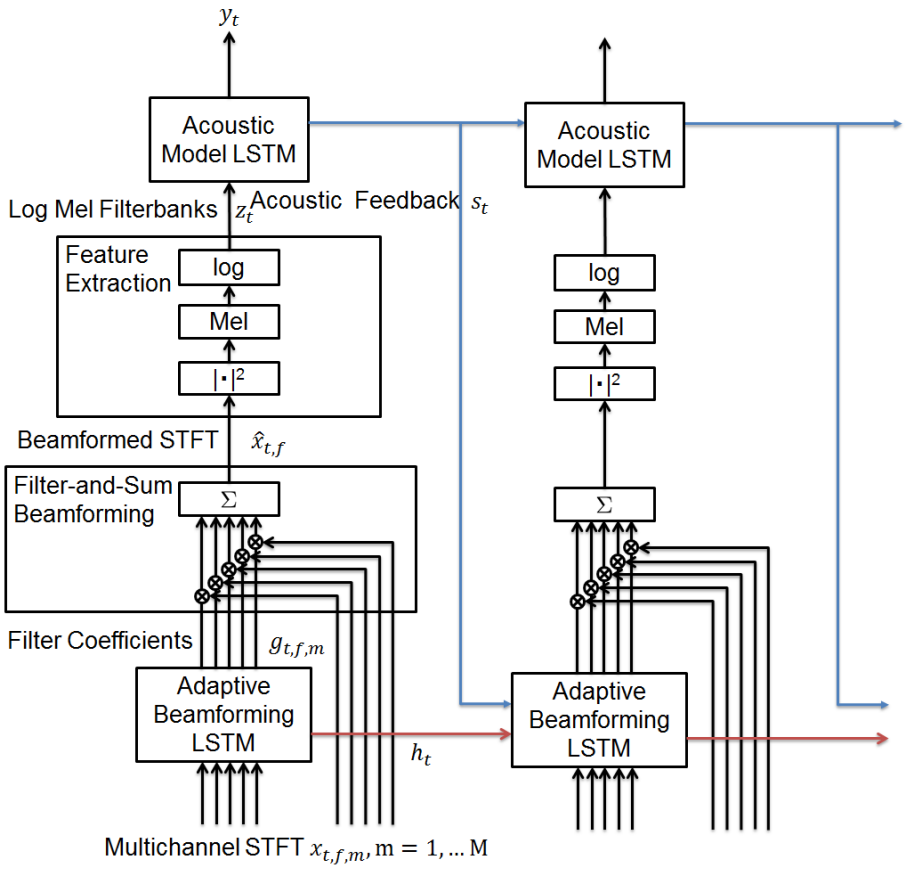


Figure 8.1: The unfolded integrated network of an LSTM adaptive beamformer and an LSTM acoustic model. The acoustic feedback (in blue) is introduced to allow the hidden units in LSTM acoustic model to assist in predicting the filter coefficient at current time.

Direct training of the integrated network easily falls into a local optimum as the gradients for the LSTM beamformer and the deep LSTM acoustic model have different dynamic ranges. For a robust estimation of the model parameters, the training should be performed in sequence as shown in Algorithm 1.

Algorithm 1 Train LSTM adaptive beamformer and deep LSTM acoustic model

1: Train a deep LSTM acoustic model with log Mel filterbank feature extracted from the speech of all channels to minimize the cross-entropy objective.
2: Initialize the integrated network with the deep LSTM acoustic model in Step 1.
3: Train the integrated network with the ASR cross-entropy objective, update only the parameters in the LSTM beamformer.
4: Jointly train the integrated network in Step 3 with the ASR cross-entropy objective, updating all parameters in the LSTM beamformer and deep LSTM acoustic model.
5: Introduce the acoustic feedback and re-train the integrated network with the ASR objective, updating all the parameters.

During recognition, the acoustic probabilities yielded by the integrated network and test utterances are combined with the state transition probabilities from the HMM and the word transition probabilities from the language model which can be performed through weighted finite state transducer.

8.3 Experiments

8.3.1 Dataset Description

The CHiME-3 dataset is released with the 3rd CHiME speech Separation and Recognition Challenge [132], which incorporates the Wall Street Journal corpus sentences spoken by talkers situated in challenging noisy environment recorded using a 6-channel tablet based microphone array. CHiME-3 dataset consists of both real and simulated data. The real data is recorded speech spoken by actual talkers in four real noisy environments (on buses, in cafés, in pedestrian areas, and at street junctions). To generate the simulated data, the clean speech is first convoluted with the estimated impulse response of the environment and then mixed with the background noise separately recorded in that environment [133]. The training set consists of 1600 real noisy utterances from 4 speakers, and 7138 simulated noisy utterances from the 83 speakers in the WSJ0 SI-84 training set recorded in 4 noisy environments. There are 3280 utterances in the development set including 410 real and

Table 8.1: The WER performance (%) of the baseline LSTM acoustic model (AM), BeamformIt-enhanced signal as the input of the AM, joint training of LSTM beamformer and LSTM acoustic model (BF+AM) with or without acoustic feedback.

System	Input Feature	Simu Dev	Real Dev	Simu Test	Real Test
AM (baseline)	Fbank	16.15	19.24	23.02	32.88
BeamformIt+AM	STFT	14.32	12.99	24.36	21.21
BF+AM (fixed)	STFT	15.23	15.01	23.14	25.64
BF+AM	STFT	14.43	15.19	22.40	25.13
BF+AM+Feedback	STFT	14.28	15.10	22.23	24.91

410 simulated utterances for each of the 4 environments. There are 2640 utterances in the test set including 330 real and 330 simulated utterances for each of the 4 environments. The speakers in training set, development set and the test set are mutually different (i.e., 12 different speakers in the CHiME-3 dataset). The training, development and test data are all recorded in 6 different channels. The WSJ0 text corpus is also used to train the language model.

8.3.2 Baseline System

The baseline system is built with Chainer [168] and Kaldi [91] toolkits. 40-dimensional log Mel filterbank features extracted by Kaldi from all 6 channels are used to train a deep LSTM acoustic model using Chainer. The LSTM has 3 layers and each hidden layer has 1024 units. The output layer has 1985 units, each of which corresponds to a senone target. The input feature is first projected to a 1024 dimensional space before being fed into the LSTM. The forced alignment generated by a GMM-HMM system trained with data from all 6 channels is used as the target for LSTM training. During evaluation, only the development and test data from the 5^{th} channel is used for testing (only for the baseline system). The LSTM is trained using BPTT with a truncation size of 100 and a learning rate of 0.01. The batch size for stochastic gradient descent (SGD) is 100. The WER performance of the baseline system is shown in Table 8.1.

8.3.3 LSTM Adaptive Beamformer

The 257-dimensional complex STFT coefficients are extracted for the speech in channels $1, 3, 4, 5, 6$. The real and imaginary parts of STFT coefficients from all the 5 channels are concatenated together

to form $257 \times 2 \times 5 = 2570$ dimensional input of the beamforming LSTM. The input is projected to 1024 dimensional space before being fed into the LSTM. The beamforming LSTM has one hidden layer with 1024 units. The hidden units vector is projected to 5 sets of $257 \times 2 = 514$ dimensional filter coefficients for adaptively beamforming signals from 5 channels using Eq. (8.2). The MSE objective is computed between the beamformed signal and BeamformIt [169]. The beamforming LSTM is trained using BPTT with a truncation size of 100, a batch size of 100 and a learning rate of 1.0.

8.3.4 Joint Training of the Integrated Network

The baseline LSTM acoustic model trained in Section 8.3.2 and the LSTM adaptive beamformer trained in Section 8.3.3 are concatenated together as the initialization of the integrated network. A feature extraction layer is inserted in between the two LSTMs to extract 40-dimensional log Mel filterbank features with Eq. (8.8). The integrated network is trained in a way described in Steps 3, 4 and 5 of Section 8.2.4. BPTT with a truncation size of 100 and a batch size of 100 and a learning rate of 0.01 is used for training. The data from all 5 channels in the development and test set is used for evaluating the integrated network. The WER performance for different cases are shown in Table 8.1.

8.3.5 Result Analysis

From Table 8.1, the best system is the integrated network of an LSTM adaptive beamformer and a deep LSTM acoustic model with the acoustic feedback, which achieves 14.28%, 15.10%, 22.23%, 24.91% WERs on the simulated development set, real development set, simulated test set and real test set of the CHiME-3 dataset respectively. The joint training of the integrated network without updating the deep LSTM acoustic model achieves absolute gains of 0.92%, 4.23% and 7.24% over the baseline system on the simulated development set, real development set and real test set respectively. The joint training of the integrated network with all the parameters updated achieves absolute gains of 1.72%, 4.05%, 0.62% and 7.75% respectively over the baseline systems on the simulated development set, real development set, simulated test set and real test set respectively. The large performance improvement justifies that the LSTM adaptive beamformer is able to estimate the real-

time filter coefficients adaptively in response to the changing source position, environmental noise and room impulse response with the LSTM acoustic model jointly trained to optimize the ASR objective. Further absolute gains of 0.15%, 0.09%, 0.17% and 0.22% are achieved with the introduction of acoustic feedback, which indicates that the high-level acoustic information is also helpful in predicting the filter coefficients at the next time step.

Note that although the proposed system with acoustic feedback achieves 0.04% and 2.13% absolute gains over the beamformed signal generated by BeamformIt on the simulated development and test sets, it does not work as well as the BeamformIt on the real development and test sets. One possible reason is that the training data of the integrated network is mostly simulated data (7138 out of 8738 training utterances are simulated). The integrated network is over-fitted to the simulated data such that it does not perform as good as on the real data in the test set. However, the BeamformIt does not need any training data and is not over-fitted to either simulated or real data, therefore, it works better than the proposed method on real data but worse than proposed method on simulated data. Another factor is that the BeamformIt implementation, the two-step time delay of arrival Viterbi postprocessing makes use of both the past and future information in predicting the best alignment of multiple channels at the current time, while in our system, only the history in the past is utilized to estimate the current filter coefficients. This also explains the differences in WER performance and can be alleviated by using bidirectional LSTM as part of the future work.

8.3.6 Beamformed Feature

The LSTM beamformer adaptively predicts the time-variant beamforming coefficients and performs filter-and-sum beamforming over the 5 input channels. The log Mel filter bank feature is obtained from the STFT coefficients. From Fig. 8.2, we see that the log Mel filter bank feature obtained from the LSTM adaptive beamformer is quite similar to the log Mel filter bank feature extracted from the STFT coefficients beamformed by BeamformIt for the same utterance. The SNR is not high but matches the LSTM acoustic model well for maximizing the ASR performance.

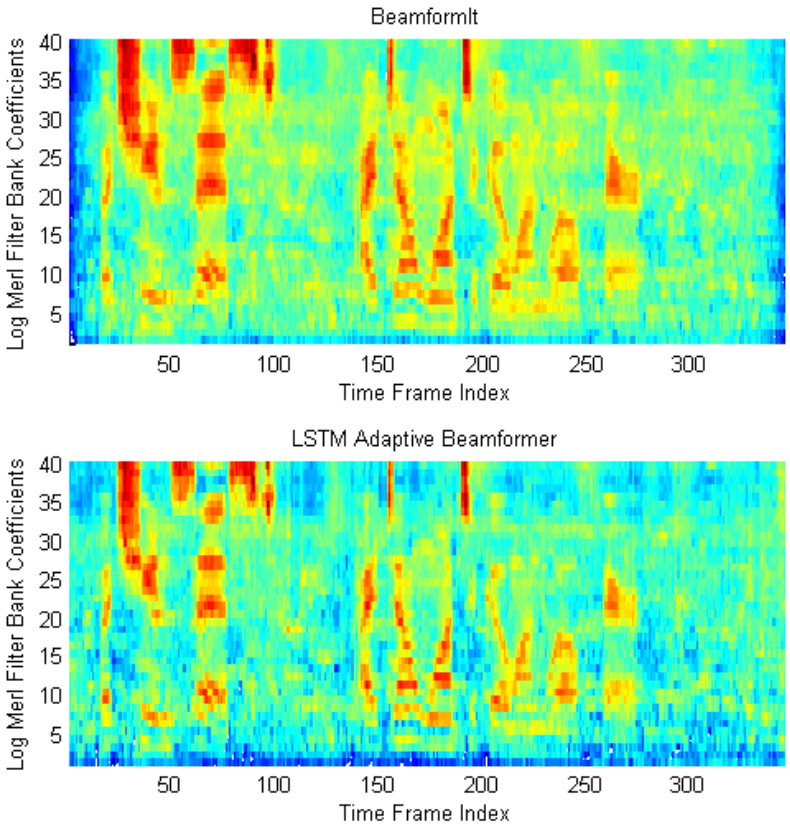


Figure 8.2: The comparison of the log Mel filter bank coefficients of the same utterance extracted from STFT coefficients beamformed by BeamformIt (upper) and LSTM adaptive beamformer (lower) .

8.4 Conclusions

In this chapter, LSTM adaptive beamforming is proposed to adaptively predict the real-time beamforming filter coefficients to deal with the time-variant source location, environmental noise and room impulse response inherent in the multichannel speech signal. To achieve robust ASR, the LSTM adaptive beamformer is jointly trained with a deep LSTM acoustic model to optimize the ASR objective. This framework achieves absolute gains of 1.72%, 4.05%, 0.62% and 7.75% over the baseline system on the CHiME-3 dataset. Further improvement is achieved by introducing the acoustic feedback to assist in predicting the filter coefficients.

However, our approach does not work as well as the BeamformIt on real data because the majority of the training data is simulated such that our integrated network is overfitted to the simulated data while BeamformIt is not a data-driven approach which does not have model to be overfitted to any kind of data.

CHAPTER 9

CONCLUSIONS

In this thesis, we have achieved robust automatic speech understanding (ASU) through discriminative training of the DNN acoustic models in two different ways. To expand the application of ASU to various environments and conditions, we have proposed four adaptive training approaches to address three crucial problems regarding acoustic modeling.

The ASU is first achieved by recognizing the keywords that are semantically important. We formulate the keyword spotting as a non-uniform error ASR problem and demonstrate that DNNs acoustic models can be successfully trained on the non-uniform minimum classification error (MCE) criterion which weighs the errors on keywords much more significantly than those on non-keywords in an ASR task. The integration with a FNN-HMM system enables modeling of multi-frame distributions, which conventional systems find difficult to accomplish. The non-uniform MCE training of FNN achieves 2.48% and 7.26% FOM improvements over the cross entropy baseline system on Switchboard and HKUST datasets respectively. The further composition with the BLSTM-HMM system enables the capturing of long-term dependencies within the variable-duration dynamic speech signal instead of a fixed-size window using a FNN-HMM. The non-uniform MCE training of BLSTM achieves 4.49% and 7.37% FOM improvements over the cross entropy baseline system on Switchboard and HKUST datasets respectively. The keyword spotting system is implemented within a weighted finite state transducer (WFST) framework and the DNN is optimized using standard backpropagation and stochastic gradient decent.

ASU is commonly the process of ASR followed by spoken language understanding (SLU). To compensate for the mismatch between the ASR and objectives, we propose the minimum semantic error cost (MSEC) training of a deep BLSTM-HMM acoustic model for generating lattices that are semantically accurate and are better suited for topic spotting with LSRK. With the MSEC training, the expected semantic error cost of all possible word sequences on the lattices is minimized given the reference. The word-word semantic error cost is first computed from either the latent semantic

analysis or distributed vector-space word representations learned from the RNNs and is then accumulated to form the expected semantic error cost of the hypothesized word sequences. The MSEC achieves 3.5% - 4.5% absolute topic classification accuracy improvement over the baseline BLSTM trained with cross-entropy on Switchboard dataset.

To expand the application of ASU to various conditions and environments, we first suppress the effect of inter-speaker variability on speaker-independent (SI) DNN acoustic model by proposing speaker-invariant training (SIT). In SIT, a DNN acoustic model and a speaker classifier network are jointly optimized to minimize the senone (tied triphone state) classification loss, and simultaneously mini-maximize the speaker classification loss. A speaker-invariant and senone-discriminative deep feature is learned through this adversarial multi-task learning. With SIT, a canonical DNN acoustic model with significantly reduced variance in its output probabilities is learned with no explicit SI transformations or speaker-specific representations used in training or testing. Evaluated on the CHiME-3 dataset, the SIT achieves 4.99% relative word error rate (WER) improvement over the conventional SI acoustic model. With the unsupervised adaptation towards the test speakers, the speaker-adapted (SA) SIT acoustic model achieves additional 8.79% relative WER gain, which is 4.86% relatively improved over the SA SI acoustic model. With t-SNE visualization, we show that, after SIT, the deep feature distributions of different speakers are well aligned with each other, which verifies the strong capability of SIT in reducing speaker-variability.

Secondly, to compensate for the acoustic mismatch between training and test conditions, we propose adversarial teacher-student (T/S) learning for unsupervised adaptation of the DNN acoustic model. In this method, a student acoustic model and a condition classifier are jointly optimized to minimize the Kullback-Leibler divergence between the output distributions of the teacher and student models, and simultaneously, to min-maximize the condition classification loss. A condition-invariant deep feature is learned in the adapted student model through this procedure. We further propose multi-factorial adversarial (MFA) T/S learning which suppresses condition variabilities caused by multiple factors simultaneously. Evaluated with the noisy CHiME-3 test set, adversarial T/S learning achieves relative WER improvements of 44.30% and 4.87%, respectively, over a clean source model and a strong T/S learning baseline model on by suppressing speaker variability. MFA T/S learning achieves 44.60% and 5.38% relative WER over the unadapted and T/S adapted models.

To further improve the capability of adversarial learning in unsupervised adaptation, we propose to use domain separation network to characterize the difference between the source and target domain distributions by explicitly modeling the private component of each domain in addition to learning a domain-invariant feature (i.e. the shared component between domains) that is also senone-discriminative via adversarial learning. The private component is trained to be orthogonal with the shared component and thus implicitly increases the degree of domain-invariance of the shared component. When applied to the unsupervised environment adaptation task, DSN achieved 17.97% and 11.08% relative WER reductions from the unadapted acoustic model and the gradient reversal layer, a representative adversarial training method, for ASR on CHiME-3 dataset. The WER decreases when higher hidden representations of the DNN acoustic model are used as the initial shared component.

Thirdly, we address the far-field speech recognition in noisy and reverberant conditions problem by proposing adaptive LSTM beamforming network for multichannel ASR. An LSTM-RNN adaptively estimates the real-time beamforming filter coefficients to cope with non-stationary environmental noise and dynamic nature of source and microphones positions which results in a set of time-varying room impulse responses. The LSTM adaptive beamformer is jointly trained with a deep LSTM acoustic model to predict senone labels. Further, we use hidden units in the deep LSTM acoustic model to assist in predicting the beamforming filter coefficients. The LSTM beamforming network achieves 7.97% absolute gain over baseline systems with no beamforming on CHiME-3 real evaluation set.

REFERENCES

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

[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[3] E. Shriberg, "Spontaneous speech: How people really talk and why engineers should care," in *in Proc. European Conf. on Speech Communication and Technology (Eurospeech*, 2005, pp. 1781–1784.

[4] R. C. Rose and D. B. Paul, "A hidden markov model based keyword recognition system," in *Proc. ICASSP*, 1990, 129–132 vol.1.

[5] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *Acoustics, speech and signal processing (icassp), 2014 ieee international conference on*, IEEE, 2014, pp. 4087–4091.

[6] D. R. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Interspeech*, 2007, pp. 314–317.

[7] C. Weng and B. H. Juang, "Discriminative training using non-uniform criteria for keyword spotting on spontaneous speech," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 300–312, 2015.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[11] Z. Meng and B.-H. Juang, "Non-uniform boosted mce training of deep neural networks for keyword spotting," in *Proceedings of INTERSPEECH*, 2016, pp. 770–774.

[12] ——, "Non-uniform mce training of deep long short-term memory recurrent neural networks for keyword spotting," in *Proceedings of INTERSPEECH*, Aug. 2017, pp. 3547–3551.

[13] J. Pan, C. Liu, Z. Wang, Y. Hu, and H. Jiang, "Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why dnn surpasses gmms in acoustic modeling," in *Proc. ISCSLP*, 2012, pp. 301–305.

[14] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[15] Z. Meng and B.-H. Juang, "Minimum semantic error cost training of deep long short-term memory networks for topic spotting on conversational speech," in *Proceedings of INTERSPEECH*, Aug. 2017, pp. 2496–2500.

[16] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *JASIST*, vol. 41, no. 6, p. 391, 1990.

[17] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations.," in *Proc. of NAACL HLT*, vol. 13, 2013, pp. 746–751.

[18] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, IEEE, vol. 2, 1996, pp. 1137–1140.

[19] C. Wu and M. J. F. Gales, "Multi-basis adaptive neural network for rapid adaptation in speech recognition," in *Proc. ICASSP*, 2015, pp. 4315–4319.

[20] T. Tan, Y. Qian, and K. Yu, "Cluster adaptive training for deep neural network based acoustic model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 459–468, 2016.

[21] Z. Meng, J. Li, Z. Chen, *et al.*, "Speaker-invariant training via adversarial learning," in *Proc. ICASSP*, 2018.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, Curran Associates, Inc., 2014, pp. 2672–2680.

[23] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. ICML*, ser. Proceedings of Machine Learning Research, vol. 37, Lille, France: PMLR, 2015, pp. 1180–1189.

[24] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size DNN with output-distribution-based criteria.," in *INTERSPEECH*, 2014, pp. 1910–1914.

[25] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large-scale domain adaptation via teacher-student learning," in *INTERSPEECH*, 2017.

[26] Y. Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition.," in *Interspeech*, 2016, pp. 2369–2372.

[27] S. Sun, B. Zhang, L. Xie, and Y. Zhang, "An unsupervised deep domain adaptation approach for robust speech recognition," *Neurocomputing*, vol. 257, pp. 79 –87, 2017.

[28] Z. Meng, J. Li, Y. Gong, and B.-H. F. Juang, "Adversarial teacher-student learning for unsupervised domain adaptation," in *Proc.ICASSP*, IEEE, 2018.

[29] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. NIPS*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., Curran Associates, Inc., 2016, pp. 343–351.

[30] Z. Meng, Z. Chen, V. Mazalov, J. Li, and Y. Gong, "Unsupervised adaptation with domain separation networks for robust speech recognition," in *Proceeding of ASRU*, 2017.

[31] Z. Meng, S. Watanabe, J. R. Hershey, and H. Erdogan, "Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, IEEE, 2017, pp. 271–275.

[32] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in *Readings in speech recognition*, Elsevier, 1990, pp. 65–74.

[33] B. H. Juang, "Deep neural networks–a developmental perspective," *APSIPA Transactions on Signal and Information Processing*, vol. 5, 2016.

[34] F. Jelinek and R. Mercer, "Interpolated estimation of markov source parameters from sparse data," in *Proc. Workshop on Pattern Recognition in Practice, 1980*, 1980.

[35] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE transactions on acoustics, speech, and signal processing*, vol. 35, no. 3, pp. 400–401, 1987.

[36] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *Ieee transactions on information theory*, vol. 37, no. 4, pp. 1085–1094, 1991.

[37] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, IEEE, vol. 1, 1995, pp. 181–184.

[38] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999.

[39] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[40] T. Mikolov, "Statistical language models based on neural networks," PhD thesis, BRNO University of Technology, 2012.

[41] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*, Springer, 2008, pp. 559–584.

[42] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiát, S. Kombrink, P. Motlíček, Y. Qian, *et al.*, "Generating exact lattices in the wfst framework," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, IEEE, 2012, pp. 4213–4216.

[43] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, Springer, 2010, pp. 177–186.

[44] C. Weng, D. Yu, S. Watanabe, and B. H. F. Juang, "Recurrent deep neural networks for robust speech recognition," in *Proc. ICASSP*, 2014, pp. 5532–5536.

[45] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust asr," in *Proc. ICASSP*, 2012, pp. 4085–4088.

[46] A. Graves, N. Jaitly, and A. r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Proc. ASRU*, 2013, pp. 273–278.

[47] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling.," in *Interspeech*, 2014, pp. 338–342.

[48] Z. Meng, S. Watanabe, J. Hershey, and H. Erdogan, "Deep long short-term memory adaptive beamforming networks for multichannel robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2017.

[49] H. Erdogan, T. Hayashi, J. R. Hershey, T. Hori, C. Hori, W.-N. Hsu, S. Kim, J. Le Roux, Z. Meng, and S. Watanabe, "Multi-channel speech recognition: Lstms all the way through," in *The 4th International Workshop on Speech Processing in Everyday Environments*, 2016.

[50] A. Graves, A. r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.

[51] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.

[52] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," *arXiv preprint arXiv:1512.02595*, 2015.

[53] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *JMLR*, vol. 3, no. Aug, pp. 115–143, 2002.

[54] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification (pattern recognition)," *IEEE Transactions on signal processing*, vol. 40, no. 12, pp. 3043–3054, 1992.

[55] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 5, no. 3, pp. 257–265, 1997.

[56] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Proc. ICASSP*, vol. 11, 1986, pp. 49–52.

[57] V Valtchev, J. Odell, P. Woodland, and S. Young, "MMIE training of large vocabulary recognition systems," *Speech Commun.*, vol. 22, no. 4, pp. 303 –314, 1997.

[58] D. Povey and P. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Proc. ICASSP*, vol. 1, 2002, pp. I–105–I–108.

[59] D. Povey, "Discriminative training for large vocabulary speech recognition," PhD thesis, University of Cambridge, 2005.

[60] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition.," in *Interspeech*, Citeseer, 2006.

[61] J. Kaiser, B. Horvat, and Z. Kacic, "A novel loss function for the overall risk criterion based discriminative training of hmm models," in *Proc. ICSLP*, 2000.

[62] D. Povey and B. Kingsbury, "Evaluation of proposed modifications to mpe for large scale discriminative training," in *Proceedings of ICASSP*, vol. 4, 2007, pp. IV–321–IV–324.

[63] K. Na, B. Jeon, D.-I. Chang, S.-I. Chae, and S. Ann, "Discriminative training of hidden markov models using overall risk criterion and reduced gradient method," in *Fourth European Conference on Speech Communication and Technology*, 1995.

[64] J. Kaiser, B. Horvat, and Z. Kacic, "A novel loss function for the overall risk criterion based discriminative training of hmm models," in *Sixth International Conference on Spoken Language Processing*, 2000.

[65] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *Proc. ICASSP*, 2008, pp. 4057–4060.

[66] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks.," in *INTERSPEECH*, 2013, pp. 2345–2349.

[67] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[68] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. ICASSP*, 2013, pp. 7893–7897.

[69] Z. Huang, J. Li, S. M. Siniscalchi, I.-F. Chen, C. Weng, and C.-H. Lee, "Feature space maximum a posteriori linear regression for adaptation of deep neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[70] Z. Huang, S. M. Siniscalchi, I.-F. Chen, J. Li, J. Wu, and C.-H. Lee, "Maximum a posteriori adaptation of network parameters in deep models," in *Interspeech*, 2015.

[71] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE transactions on speech and audio processing*, vol. 2, no. 2, pp. 291–298, 1994.

[72] Z. Huang, J. Li, S. Siniscalchi, *et al.*, "Rapid adaptation for deep neural networks through multi-task learning," in *Interspeech*, 2015.

[73] J. Li, M. L. Seltzer, X. Wang, R. Zhao, and Y. Gong, "Large-scale domain adaptation via teacher-student learning," in *INTERSPEECH*, 2017.

[74] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size DNN with output-distribution-based criteria.," in *INTERSPEECH*, 2014, pp. 1910–1914.

[75] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. D. Mori, "Linear hidden transformations for adaptation of hybrid ann/hmm models," *Speech Commun.*, vol. 49, no. 10, pp. 827 –835, 2007.

[76] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011, pp. 24–29.

[77] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *Proc. ICASSP*, 2014, pp. 6359–6363.

[78] Y. Zhao, J. Li, and Y. Gong, "Low-rank plus diagonal adaptation for deep neural networks," in *Proc. ICASSP*, 2016, pp. 5005–5009.

[79] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. ASRU*, 2013, pp. 55–59.

[80] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[81] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code," in *Proc.ICASSP*, 2013, pp. 7942–7946.

[82] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1713–1725, 2014.

[83] D. Serdyuk, K. Audhkhasi, P. Brakel, B. Ramabhadran, S. Thomas, and Y. Bengio, "Invariant representations for noisy speech recognition," in *Proc. NIPS Workshop*, 2016.

[84] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[85] M. Rahim, C.-H. Lee, and B.-H. Juang, "Discriminative utterance verification for connected digits recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 5, no. 3, pp. 266–277, 1997.

[86] R. Rose, "Keyword detection in conversational speech utterances using hidden markov model based continuous speech recognition," *Computer Speech & Language*, vol. 9, no. 4, pp. 309 –333, 1995.

[87] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[88] Q. Fu, Y. Zhao, and B.-H. Juang, "Automatic speech recognition based on non-uniform error criteria," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 780–793, 2012.

[89] C. Weng and B.-H. Juang, "Adaptive boosted non-uniform mce for keyword spotting on spontaneous speech," in *Proc. ICASSP*, 2013, pp. 6960–6964.

[90] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiat, S. Kombrink, P. Motlicek, Y. Qian, K. Riedhammer, K. Vesely, and N. T. Vu, "Generating

exact lattices in the wfst framework," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 2012, pp. 4213–4216.

[91] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *Proc. ASRU*, IEEE Catalog No.: CFP11SRW-USB, IEEE Signal Processing Society, Dec. 2011.

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

[93] *Cedict-on-line chinese tools*, https://www.mdbg.net/chinese/dictionary?page=cedict.

[94] P. Fung, C. Y. Ma, and W. K. Liu, "Map-based cross-language adaptation augmented by linguistic knowledge: From english to chinese," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[95] *The cmu pronouncing dictionary*, http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[96] *Sequitur g2p-a trainable grapheme-to-phoneme converter*, https://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html.

[97] *Mmseg - chinese segment on mmseg algorithm*, https://pypi.org/project/mmseg/1.3.0/.

[98] J. H. Wright, M. J. Carey, and E. Parris, "Improved topic spotting through statistical modelling of keyword dependencies," in *Proc. ICASSP*, vol. 1, 1995, 313–316 vol.1.

[99] A. L. Gorin, G. Riccardi, and J. H. Wright, "How may i help you?" *Speech Commun.*, vol. 23, no. 1-2, pp. 113–127, Oct. 1997.

[100] J. H. Wright, A. L. Gorin, and G. Riccardi, "Automatic acquisition of salient grammar fragments for call-type classification," in *EUROSPEECH*, 1997.

[101] K. Myers, M. J. Kearns, S. P. Singh, and M. A. Walker, "A boosting approach to topic spotting on subdialogues," in *Proc. ICML*, 2000, 655–662, ISBN: 1-55860-707-2.

[102] C.-H. Lee, R Carpenter, W Chou, J Chu-Carroll, W Reichl, A Saad, and Q Zhou, "A study on natural language call routing," in *Interactive Voice Technology for Telecommunications Applications, 1998. IVTTA'98. Proceedings. 1998 IEEE 4th Workshop*, IEEE, 1998, pp. 37–42.

[103] C.-H. Lee, B. Carpenter, W. Chou, J. Chu-Carroll, W. Reichl, A. Saad, and Q. Zhou, "On natural language call routing," *Speech Communication*, vol. 31, no. 4, pp. 309–320, 2000.

[104] J. Chu-Carroll and B. Carpenter, "Vector-based natural language call routing," *Computational linguistics*, vol. 25, no. 3, pp. 361–388, 1999.

[105] C. Cortes, P. Haffner, and M. Mohri, "Rational kernels: Theory and algorithms," *J. Mach. Learn. Res.*, vol. 5, pp. 1035–1062, Dec. 2004.

[106] C. Cortes, P. Haffner, and M. Mohri, "Lattice kernels for spoken-dialog classification," in *Proc. ICASSP*, vol. 1, 2003, I–628–31 vol.1.

[107] T. J. Hazen, F. Richardson, and A. Margolis, "Topic identification from audio recordings using word and phone recognition lattices," in *Proc. ASRU*, 2007, pp. 659–664.

[108] C. Weng, D. L. Thomson, P. Haffner, and B. H. F. Juang, "Latent semantic rational kernels for topic spotting on conversational speech," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1738–1749, 2014.

[109] C. Weng and B. H. Juang, "Latent semantic rational kernels for topic spotting on spontaneous conversational speech," in *Proc. ICASSP*, 2013, pp. 8302–8306.

[110] C. Fellbaum, "Wordnet," in *The Encyclopedia of Applied Linguistics*. John Wiley & Sons, Inc., 2012, ISBN: 9781405198431.

[111] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks.," in *Interspeech*, 2013, pp. 2345–2349.

[112] Andrew, H. Sak, F. de Chaumont Quitry, T. Sainath, and K. Rao, "Acoustic modelling with cd-ctc-smbr lstm rnns," in *Proc. ASRU*, 2015, pp. 604–609.

[113] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," in *Information Processing and Management*, 1988, pp. 513–523.

[114] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS*, 2013, pp. 3111–3119.

[115] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Proc. ICLR*, 2013.

[116] G. Hinton, L. Deng, D. Yu, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[117] D. Yu and J. Li, "Recent progresses in deep learning based acoustic models," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 396–409, 2017.

[118] L. Samarakoon and K. C. Sim, "Factorized hidden layer adaptation for deep neural network based acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2241–2250, 2016.

[119] Y. Miao, H. Zhang, and F. Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 11, pp. 1938–1949, 2015.

[120] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proc. ICASSP*, 2013, pp. 7947–7951.

[121] P. Swietojanski, J. Li, and S. Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1450–1463, 2016.

[122] Y. Zhao, J. Li, J. Xue, and Y. Gong, "Investigating online low-footprint speaker adaptation using generalized linear regression and click-through data," in *ICASSP*, 2015, pp. 4310–4314.

[123] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition.," in *Interspeech*, 2013, pp. 2365–2369.

[124] Y. Zhao, J. Li, K. Kumar, and Y. Gong, "Extended low-rank plus diagonal adaptation for deep and recurrent neural networks," in *ICASSP*, 2017, pp. 5040–5044.

[125] G. Saon, G. Kurata, T. Sercu, *et al.*, "English conversational telephone speech recognition by humans and machines," *Proc. Interspeech*, 2017.

[126] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[127] M. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75 –98, 1998.

[128] M. J. F. Gales, "Cluster adaptive training of hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 417–428, 2000.

[129] L. Samarakoon and K. C. Sim, "Factorized hidden layer adaptation for deep neural network based acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2241–2250, 2016.

[130] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1713–1725, 2014.

[131] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 55–59.

[132] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third chime speech separation and recognition challenge: Dataset, task and baselines," in *Proc. ASRU*, 2015, pp. 504–511.

[133] T. Hori, Z. Chen, H. Erdogan, J. R. Hershey, J. L. Roux, V. Mitra, and S. Watanabe, "The merl/sri system for the 3rd chime challenge using beamforming, robust feature extraction, and advanced speech recognition," in *Proc. ASRU*, 2015, pp. 475–481.

[134] J. Li, D. Yu, J.-T. Huang, and Y. Gong, "Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM," in *Proc. SLT*, IEEE, 2012, pp. 131–136.

[135] H. Erdogan, T. Hayashi, J. R. Hershey, T. Hori, C. Hori, W.-N. Hsu, S. Kim, J. Le Roux, Z. Meng, and S. Watanabe, "Multi-channel speech recognition: Lstms all the way through," in *CHiME-4 workshop*, 2016.

[136] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A.-r. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc.ASRU*, IEEE, 2011, pp. 30–35.

[137] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. INTERSPEECH*, 2012.

[138] L. Deng, J. Li, J.-T. Huang, *et al.*, "Recent advances in deep learning for speech research at Microsoft," in *ICASSP*, 2013, pp. 8604–8608.

[139] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, *Robust Automatic Speech Recognition: A Bridge to Practical Applications*. Academic Press, 2015.

[140] F. B. H. Sak A. Senior, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Interspeech*, 2014.

[141] J. Li, R. Zhao, Z. Chen, *et al.*, "Developing far-field speaker system via teacher-student learning," in *Proc. ICASSP*, 2018.

[142] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. INTERSPEECH*, 2011.

[143] Z. Huang, J. Li, S. M. Siniscalchi, I.-F. Chen, J. Wu, and C.-H. Lee, "Rapid adaptation for deep neural networks through multi-task learning.," in *Interspeech*, 2015, pp. 3625–3629.

[144] P. Swietojanski and S. Renals, "Differentiable pooling for unsupervised speaker adaptation," in *Proc. ICASSP*, 2015, pp. 4305–4309.

[145] Z. Q. Wang and D. Wang, "Unsupervised speaker adaptation of batch normalized acoustic models for robust asr," in *Proc. ICASSP*, 2017, pp. 4890–4894.

[146] D. Yu, A. Eversole, M. Seltzer, K. Yao, Z. Huang, B. Guenter, O. Kuchaiev, Y. Zhang, F. Seide, H. Wang, *et al.*, "An introduction to computational networks and the computational network toolkit," *Microsoft Technical Report MSR-TR-2014–112*, 2014.

[147] M. Delcroix, T. Yoshioka, A. Ogawa, Y. Kubo, M. Fujimoto, N. Ito, K. Kinoshita, M. Espi, S. Araki, T. Hori, and T. Nakatani, "Strategies for distant speech recognitionin reverberant environments," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, pp. 1–15, 2015.

[148] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*. Springer Science & Business Media, 2008, vol. 1.

[149] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE assp magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[150] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 260–276, 2010.

[151] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1529–1539, 2007.

[152] H Erdogan, J. Hershey, S Watanabe, M Mandel, and J Le Roux, "Improved mvdr beamforming using single-channel mask prediction networks," 2016.

[153] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "Blstm supported gev beamformer front-end for the 3rd chime challenge," in *Proc. ASRU*, 2015, pp. 444–451.

[154] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *Proc. ICASSP*, 2016, pp. 5745–5749.

[155] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and Andrew, "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms," in *Proc. ASRU*, 2015, pp. 30–36.

[156] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, and M. Bacchiani, "Factored spatial and spectral multichannel raw waveform cldnns," in *Proc. ICASSP*, 2016, pp. 5075–5079.

[157] J. T. Geiger, F. Weninger, J. F. Gemmeke, M. W?llmer, B. Schuller, and G. Rigoll, "Memory-enhanced neural networks and nmf for robust asr," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 6, pp. 1037–1046, 2014.

[158] F. Weninger, J. Geiger, M. W?llmer, B. Schuller, and G. Rigoll, "Feature enhancement by deep {lstm} networks for {asr} in reverberant multisource environments," *Computer Speech & Language*, vol. 28, no. 4, pp. 888 –902, 2014.

[159] H. Erdogan, J. R. Hershey, S. Watanabe, and J. L. Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 708–712.

[160] B. Li, T. N. Sainath, R. J. Weiss, K. W. Wilson, and M. Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," in *Proc. Interspeech*, 2016.

[161] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[162] A. Graves, A. r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.

[163] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling.," in *Interspeech*, 2012, pp. 194–197.

[164] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks.," in *ICML*, vol. 14, 2014, pp. 1764–1772.

[165] H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, and J. Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4280–4284.

[166] D. Bahdanau, J. Chorowski, D. Serdyuk, Y. Bengio, *et al.*, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 4945–4949.

[167] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[168] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: A next-generation open source framework for deep learning," in *Proc. NIPS Workshop*, 2015.

[169] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

VITA

Zhong Meng received his B.S. degree from the School of Information Science and Engineering, Southeast University (with the highest honor) in 2012 and his M.S. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology in 2014. He was a research intern with Microsoft Research, Mitsubishi Electric Research Labs and AT&T Labs Research in 2017, 2016 and 2015 respectively. He was awarded the Best Paper Nomination in IEEE ASRU 2017.

In addition to the "discriminative and adaptive training for speech recognition and understanding" work presented in this thesis, Zhong Meng is also actively engaged in the research regarding speech enhancement, speaker recognition and etc. The publications during his Ph.D. study in Georgia Institute of Technology are listed below.

1. **Zhong Meng**, Jinyu Li, Yifan Gong, Biing-Hwang (Fred) Juang, "Cycle-Consistent Speech Enhancement", in Proc. InterSpeech 2018

2. **Zhong Meng**, Jinyu Li, Yifan Gong, Biing-Hwang (Fred) Juang, "Adversarial Feature-Mapping for Speech Enhancement", in Proc. InterSpeech 2018

3. **Zhong Meng**, Jinyu Li, Yifan Gong, Biing-Hwang (Fred) Juang, "Adversarial Teacher-Student Learning for Unsupervised Domain Adaptation", in Proc. ICASSP 2018

4. **Zhong Meng**, Jinyu Li, Zhuo Chen, Yong Zhao, Vadim Mazalov, Yifan Gong, Biing-Hwang (Fred) Juang, "Speaker-Invariant Training via Adversarial Learning", in Proc. ICASSP 2018

5. **Zhong Meng**, Zhuo Chen, Vadim Mazalov, Jinyu Li, Yifan Gong, "Unsupervised Adaptation With Domain Separation Networks for Robust Speech Recognition", in Proc. ASRU 2017, **Best Paper Nomination**

6. **Zhong Meng**, Biing-Hwang (Fred) Juang, "Minimum Semantic Error Cost Training of Deep Long Short-Term Memoy Networks for Topic Spotting on Conversational Speech", in Proc. InterSpeech 2017 **(Oral)**

7. **Zhong Meng**, Biing-Hwang (Fred) Juang, ”Non-Uniform MCE Training of Long Short-Term Memory for Keyword Spotting”, in Proc. InterSpeech 2017 **(Oral)**

8. **Zhong Meng**, Shinji Watanabe, John Hershey, Hakan Erdogan, ”Deep Long Short-Term Memory Adaptive Beamforming Networks for Multichannel Robust Speech Recogntion”, in Proc. ICASSP 2017

9. **Zhong Meng**, Biing-Hwang (Fred) Juang, ”Non-Uniform MCE Training of Deep Neural Networks for Keyword Soptting”, in Proc. InterSpeech 2016

10. **Zhong Meng**, Biing-Hwang (Fred) Juang, ”Statistical Modeling of Speakers Voice with Temporal Co-Location for Active Voice Authentication”, in Proc. InterSpeech 2016 **(Oral)**

11. **Zhong Meng**, M Umair Bin Altaf, Biing-Hwang (Fred) Juang, ”Active Voice Authentication”, submitted to ACM Transactions on Privacy and and Security

12. **Zhong Meng**, M Umair Bin Altaf, Biing-Hwang (Fred) Juang, ”Non-Uniform Discriminative of Deep Neural Network Acoustic Models for Keyword Spotting”, submitted to IEEE/ACM Transactions on Audio, Speech, and Language Processing

13. Hakan Erdogan, Tomoki Hayashi, John R. Hershey, Takaaki Hori, Chiori Hori, Wei-Ning Hsu, Suyoun Kim, Jonathan Le Roux, **Zhong Meng**, Shinji Watanabe **(Alphabetical Order)**, ”Multi-Channel Speech Recognition: LSTMs All the Way Through”, The 4th International Workshop on Speech Processing in Everyday Environments