CHILD SPEECH RECOGNITION AS LOW-RESOURCE AUTOMATIC SPEECH RECOGNITION

by

Fei Wu

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Master of Science.

Baltimore, Maryland

May, 2020

Abstract

This thesis investigates child speech recognition as a low-resource scenario of automatic speech recognition (ASR), and explores multiple methods to improve the performance of both hybrid and end-to-end ASR models in recognizing children's speech. Similar to ASR for adults, child speech recognition aims to transcribe the content of audio recordings into text automatically. Due to the difference in vocal characteristics, ASR models trained on only adult speech data are not adequate for recognizing child speech. With limited public available child speech corpora, recognizing child speech calls for more data-efficient methods to develop ASR systems.

In this thesis, three strategies widely used in low-resource ASR are investigated for child speech recognition:

1. Using compact model parameterization: factorized time delay neural networks (TDNN-F) are used as more data-efficient acoustic models (AM) for Deep Neural Network (DNN)-HMM hybrid ASR models;

2. Adapting models trained on out-of-domain data: transfer learning is used to

adapt end-to-end ASR model trained on adult speech for child speech recognition

3. Making creative use of available in-domain data: different data augmentation methods are applied to enhance existing child speech data to train hybrid ASR models.

Empirical results are presented on several publicly available data sets, and are compared with previously published results on the same data sets.

First Reader: Dr. Sanjeev Khudanpur

Second Reader: Dr. Leibny Paola García-Perera

External Examiner: Dr. Daniel Povey

Acknowledgments

I would like to express my sincere gratitude to my advisors, Dr. Sanjeev Khudanpur, Dr. Daniel Povey, and Dr. Leibny Paola García-Perera for their advice and guidance throughout the time I am at Johns Hopkins University. I would like to thank Dr. Povey for letting me join his lab, even though I had no prior experience in speech processing. I can not thank Dr. García-Perera enough for putting up with all my procrastination and fumbles, and being available and supportive whenever I need academic and career advice. I am grateful that Dr. Khudanpur always points the direction when I am lost, and always has a remedy, even at the last minute before a submission deadline.

I am incredibly lucky to have many wonderful coworkers and friends at Johns Hopkins. I would like to thank Tongfei Chen, Arya McCarthy, Yunmo Chen, Desh Raj, Yiming Wang, and Jiamin Xie, for all the intellectual discussion and their pleasant companionship, without which my graduate experience would not have been as lively as it is. All those late nights we shared will always be a flame, shining in the darkness of lost time.

ACKNOWLEDGMENTS

I can not possibly express enough gratitude to my beloved parents, who has been supporting me all along the way and in all ways imaginable.

Dedication

To my grandmothers.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Automatic speech recognition (ASR) describes the task of transcribing the content of a speech recording to written text using computers. The task is traditionally decomposed as two sub-tasks, acoustic modeling (AM) and language modeling (LM), where the first task focuses on mapping a sequence of acoustic features to possible sequences of phonetic units, and the latter on transducing a sequence of phonetic units into meaningful sentences. For acoustic modeling, Hidden-Markov-Model (HMM)-based systems have long been popular (Jelinek, 1976; Levinson, Rabiner, and Sondhi, 1983). HMMs are typically used with either Gaussian Mixture Model (GMM) (Stuttle, 2003) or deep neural network (DNN) (Yu and Deng, 2016), and called GMM-HMM systems and hybrid ASR systems respectively. For language modeling, traditional n-gram

LMs (Goodman, 2001) are usually used in the form of weighted finite-state transducers (WFST) (Mohri, Pereira, and Riley, 2008), composed with the HMM in AM; while recurrent neural network LMs (RNNLM) (Mikolov et al., 2010) are typically used to rescore the outputs of the AM (Deoras et al., 2011; Xu et al., 2018). Recently, with the advances in sequence modeling using deep learning, pure neural ASR systems based on the sequence-to-sequence model (Sutskever, Vinyals, and Le, 2014; Bahdanau, Cho, and Bengio, 2014), also referred to as end-to-end (E2E) ASR, are being increasingly studied (Chorowski et al., 2014; Graves and Jaitly, 2014; Chan et al., 2016). They typically require more training data than hybrid systems, but offer a certain simplification in system design and development. Hybrid systems and E2E systems will be investigated in this thesis.

1.2 Motivation

In this thesis, child speech recognition is investigated as a low-resource scenario of ASR using both the hybrid approach and end-to-end approach.

Advances in ASR make human-computer communication much easier and enable applications of digital voice assistants in fields such as home automation, customer service, and medical assistance (Vajpai and Bora, 2016). Some of these applications can benefit children, especially pre-school children who cannot yet communicate through written text. For example, automatic reading assessment (Zechner, Evanini, and Laitusis, 2012; Evanini and Wang, 2013) and interactive reading tutor (Mostow, 2012) help children learn both first and foreign languages with less guidance from teachers. However, performance of such applications are greatly limited by the unsatisfactory accuracy in child speech recognition. Potamianos and Narayanan (2003) showed that the word error rate (WER) for child speech recognition can be up to 2–5 times higher than ASR for adults, even when using the same ASR model.

Challenges in child speech recognition come from both acoustic modeling and language modeling (Li and Russell, 2002), as well as the insufficiency of training data. Previous research reports a higher variance in acoustic features among different speakers due to the development of the vocal tract, which makes ASR for child speech harder than for adults. Speech recordings of pre-teen speakers also contain more hesitation, inaccurate pronunciation and syntax, challenging the LM trained on well-written text. While adult speech corpora are not well matched for training ASR systems for child speech due to the difference in acoustic features, such as formant distribution, publicly available child speech corpora are quite limited. For instance, some well-established adult English speech corpora, e.g. LibriSpeech (Panayotov et al., 2015), have more than 1000 hours of annotated recordings, while prevalent child speech corpora in English have less than 100 hours of annotated data (Claus et al., 2013). The problem is even more acute in other languages.

This thesis focuses on the third challenge in child speech recognition, the lack of training data, as challenges in AM or LM can be greatly alleviated, if not solved, with enough data. Liao et al. (2015), utilizing 459 million frames or more than 1000 hours of private data, achieved a competitive WER on child speech compared to performance of ASR for adults. However, not all researchers have the luxury of such a large corpus, which calls for more data-efficient solutions to child speech recognition.

1.3 Related Work

Efforts in improving accuracy of child speech recognition has been made regarding to many different aspects of this challenging task, the majority of which falls into one or more of the following categories:

- Finer annotation of the data
- Data augmentation
- More robust feature extraction or feature-level adaptions
- Data-efficient structure for neural models
- Utilize out-of-domain data by using multi-task learning or transfer learning techniques

Finer annotation improves child speech ASR by providing more information to the corpus. For example, child speech datasets are divided into multiple subsets based on the age of the speakers to lower the inter-speaker variance introduced by vocal tract development (Gerosa et al., 2009). Batliner et al. (2005) and Beckman et al. (2017)

used sub-word (phonemes, etc) transcripts for a more accurate annotation of audio data, including the mistakes. Sub-word transcription is particularly important for detecting mistakes and hesitation in child speech (Zechner, Evanini, and Laitusis, 2012) or speech of non-native speakers (Leung, Liu, and Meng, 2019).

Data augmentation improves child speech recognition by synthetically constructing more training data from the available training set. Adding different noises (white noise, babbling noise, music, etc) and simulated reverberation to the original data set creates multiple copies of training data, and shows improvement in ASR for low-resource languages (Snyder, Chen, and Povey, 2015; Wang et al., 2019; Pulugundla et al., 2018; Ko et al., 2017). Vocal Tract Length Perturbation (VTLP) (Jaitly and Hinton, 2013; Kim et al., 2019) and spectrum augmentation (SpecAugment) (Park et al., 2019) are popular data augmentation techniques used in end-to-end ASR, which modify the original data on the feature level to create modified copies of the original data. Both augment the data by warping the frequency axis, and SpecAugment also masks out some frequency. Speech data synthesized by using text-to-speech (TTS) (Hayashi et al., 2019), speech conversion, or generative adversarial network (GAN) techniques has also been used for data augmentation (Hu, Tan, and Qian, 2018; Sheng, Yang, and Qian, 2019). Qian et al. (2016) and Fainberg et al. (2016) also shows that, performance of child speech recognition can be improved by simply augment the training set by including adult female speech.

Feature level adaptation refers to the effort in better feature extraction and feature-level adaptation. Perceptual linear predictive (PLP) (Hermansky, 1990) has shown improvement in child speech recognition when first introduced to the speech community. Vocal tract length normalization (VTLN) normalizes the spectral distribution of different speakers by a transformation on the feature-level.

In the context of DNNs, effort has been made to look for more **data-efficient model structures**. While effective on large datasets, neural models tend to suffer more than traditional models in a low-resource scenario. Factorized time-delay neural networks (TDNN-F)(Povey et al., 2018; Pulugundla et al., 2018) provides a more data-efficient neural model structure for ASR for low-resource languages, with the intuition that neural models suffer more from data insufficiency due to its large number of parameters compared to the traditional models. Further details of TDNN-F are discussed in Chapter 2.

Transfer learning or multi-task learning is also widely use to utilize out-of-domain data for neural model training. For hybrid systems, several different strategies (Shivakumar and Georgiou, 2020; Qian et al., 2016) have been presented to improve child speech recognition by pretraining a time delay neural network (TDNN) using sufficient adult speech data, and then fine-tuning it with limited child speech data. Multilingual training utilizes speech data in different languages to provide more data for training AM in hybrid systems (Ghoshal, Swietojanski, and Renals, 2013; Huang et al., 2013; Vu et al., 2014) or end-to-end ASR systems (Kannan et al., 2019) using multi-task training techniques. Success in multilingual ASR has also been introduced to improve low-resource ASR using either hybrid (Sahraeian and Van Compernolle, 2016) or end-to-end systems (Zhou, Xu, and Xu, 2018).

1.4 Dataset

All the experiments were conducted on one or more of these corpora: CMU_Kids (Eskenazi, Mostow, and Graff, 1997; Eskenazi and Mostow, 2006), and CSLU_Kids(Shobaki, Hosom, and Cole, 2000; Shobaki, Hosom, and Cole, 2007). This section briefly describes the datasets and shows some statistics of the corpora.

Both CMU_Kids[1] and CSLU_Kids[2] are publicly available corpora recording school-age children reading prompted sentences or phrases. The CMU_Kids corpus contains 5180 utterances, recording 76 speakers reading 1 sentence per utterance.The scripted part of the CSLU_Kids corpus contains around 1118 speakers saying a single word or reading a short phrase in each utterance. The prompted scripts are used as the reference transcript of the utterances. Different speakers might read the same sentence in different recordings. Table 1.1 shows duration, number of speakers, number of different prompt texts, and vocabulary size of both corpora. Figure 1.1 shows the data percentage distributed over sentence length.

[1] Available at `https://catalog.ldc.upenn.edu/LDC97S63`

[2] Available at `https://catalog.ldc.upenn.edu/LDC2007S18`

Corpus	Duration (h)	Number of Speakers	Number of Sentences	Vocabulary Size
CMU_Kids	9.1	76	356	876
CSLU_Kids	69.3	1118	317	562

Table 1.1: Statistics of CMU_Kids and CSLU_Kids

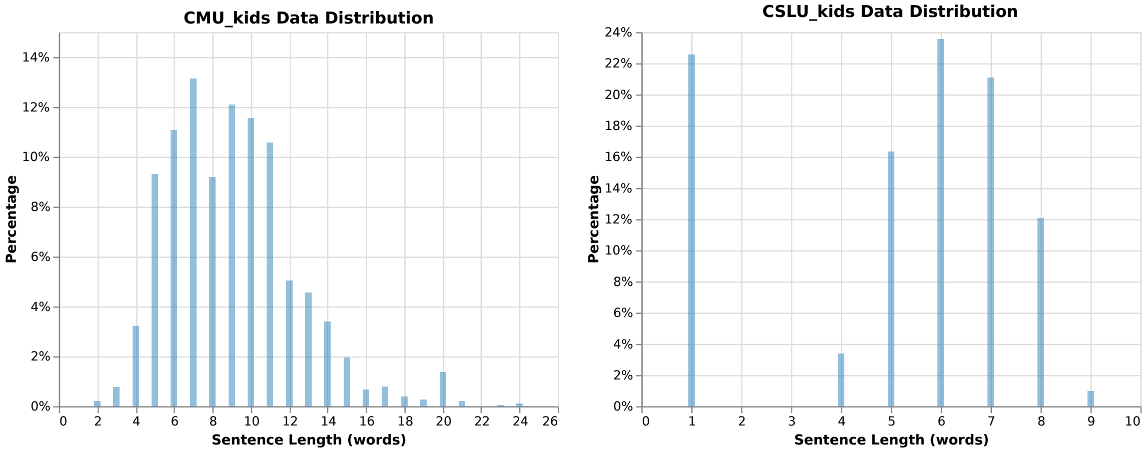

Figure 1.1: Data Distribution By Sentence Length

1.5 Contribution and Organization

In this thesis, a hybrid DNN-HMM ASR system for child speech recognition was built using the Kaldi toolkit (Povey et al., 2011), and an end-to-end ASR system was built using the Espresso toolkit (Wang et al., 2019). The main contribution of this thesis is summarized as below:

- Achieves state-of-the-art WER result on two public available child speech corpora, CMU_Kids (Eskenazi, Mostow, and Graff, 1997) and CSLU_Kids (Shobaki, Hosom, and Cole, 2000), by using a TDNN-F-HMM hybrid ASR system. The code for implementing this system has been merged into Kaldi and made available to the public.

- This thesis is, to the best of our knowledge, one of the earliest explorations of performing child speech recognition using end-to-end neural model.
- Data augmentation using additive noises and reverberation is explored for hybrid system.

The rest of this thesis is organized as follows. Chapter 2 discusses the implementation details of TDNN-F based hybrid system, and presents the results of this systems on multiple child speech corpora, along with the effect of data augmentation. Chapter 3 describes the end-to-end ASR system for child speech recognition, and presents the results with different training strategies. Lastly, Chapter 4 summarized the work presented in this thesis, and discuss possible directions for future work.

Chapter 2

Hybrid for Child Speech

2.1 Introduction

Time-delay neural network (TDNN) has long been a prevalent structure for hybrid acoustic model (AM), and on small datasets, ASR systems with such AM still hold the state-of-the-art performance. To further improve the ASR performance for low-resource languages, which normally have less than 100 hours of training data, factorized time-delay neural network (TDNN-F) was proposed as a data-efficient alternative of TDNN. Povey et al. (2018) and Pulugundla et al. (2018) both suggested similar model structure, known as TDNN-F in the former, and low-rank TDNN in the latter. The former also introduced extra training criterion for this structure to improve its performance. In computer vision (CV) literature, similar structure is also proven to be effective on convolutional neural networks (CNN), known as the bottleneck block (Lin, Chen, and Yan, 2013; Szegedy et al., 2015). In this Chapter, we applied TDNN-F to child speech recognition, and successfully improved the performance. The main results in this Chapter also appear in a refereed publication (Wu et al., 2019).

The rest of this Chapter is organized as follows. Section 2.2 describes the HMM-based hybrid system, and how we tried to improve it with TDNN-F. Section 2.3 includes details of all the experiments. Section 2.4 presents the performance of TDNN-F systems with augmentation and VTLN, and compares it with the baseline TDNN system. Finally, Section 2.5 summarizes the work in this Chapter.

2.2 TDNN-F in Hybrid System for Child Speech Recognition

Hidden Markov Models (HMM) have been popular for acoustic modelling (AM) since the 1980s. In this Chapter, we use a triphone Deep Neural Network (DNN)-HMM hybrid model for acoustic modeling, where each triphone represents a phone with its immediate left and right neighbour phones and each HMM models a triphone using 3 states. The emission probabilities of all HMM states are given by one single DNN, mapping an audio frame, represented as feature vector, to a probability distribution over all HMM states. Figure 2.3 illustrates TDNN and TDNN-F hybrid systems. More details about Figure 2.3 and the training recipe for hybrid systems can be found in Section 2.3.2 and Section 2.3.3. In this Section, we explore using TDNN-F as a data-efficient alternative for TDNN in the hybrid system. To better illustrate TDNN-F, we first discuss TDNN, and then introduce TDNN-F as its refinement.

Waibel et al. (1989) first proposed the usage of time-delay neural network (TDNN) for phoneme recognition. Peddinti, Povey, and Khudanpur (2015) then incorporated a deep TDNN into the DNN-HMM framework for better ASR performance. As shown in Figure 2.1, each TDNN layer passes forward the input frame at current time step t with its left and right neighbours within a certain context window, which is similar to a fully-connected 1-dimensional convolutional neural network (CNN). With a deep structure, TDNN has the ability to integrate information in a wide context window. Since the context windows for adjacent steps overlap with each other, it is not necessary to feed every frame to the next layer. For computational efficiency, we skip some frames based on the context window, so that the frames passed forward (shown in blue in Figure 2.1) have less or no overlapping information.

TDNN-F further improves the computation efficiency of the network by using singular value decomposition (SVD), decomposing the weight matrix of each layer into an approximation as the product of two lower rank matrices (Povey et al., 2018):

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathbf{T}} = \mathbf{M}\mathbf{N}\,, \tag{2.1}$$

where $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is a non-negative, rectangular diagonal matrix, $\mathbf{M} \in \mathbb{R}^{m\times k}$, and $\mathbf{N} \in \mathbb{R}^{k\times n}$. By choosing a suitable value of $k \leq \min\{m, n\}$, we can easily reduce the

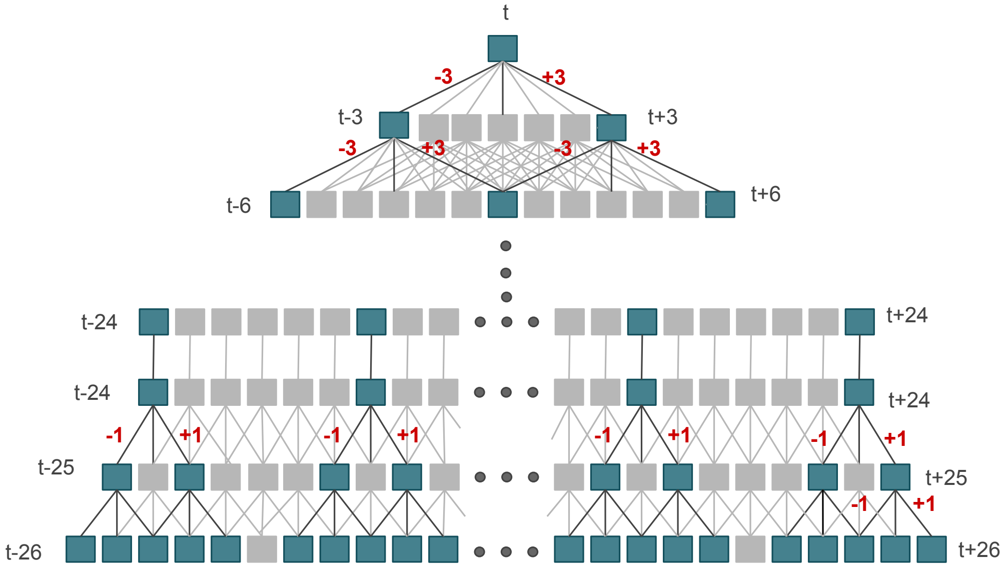

Figure 2.1: TDNN with sub-sampling

total number of parameters for this transformation. For this approach, we need to ensure that one of the two sub matrices is close to a semi-orthogonal matrix, as it is the equivalent of $\mathbf{U\Sigma}$ (or $\mathbf{\Sigma V^T}$). When training the network, after every a few updates of the whole network, we specifically update $\mathbf{N}$ with SGD using an additional objective function to guarantee that $\mathbf{N}$ is not too far from being semi-orthogonal (Povey et al., 2018). Since k is much smaller than m and n, TDNN-F can also be considered as introducing an extra bottleneck layer into the traditional TDNN. Figure 2.2 shows the factorized version of the top layer from Figure 2.1. Povey et al. (2018) also also illustrates that residual connection and a 3-stage splicing structure can further improve the performance of TDNN-F by inserting two (instead of one) bottleneck layers, and allows convolution in between the bottleneck layers.

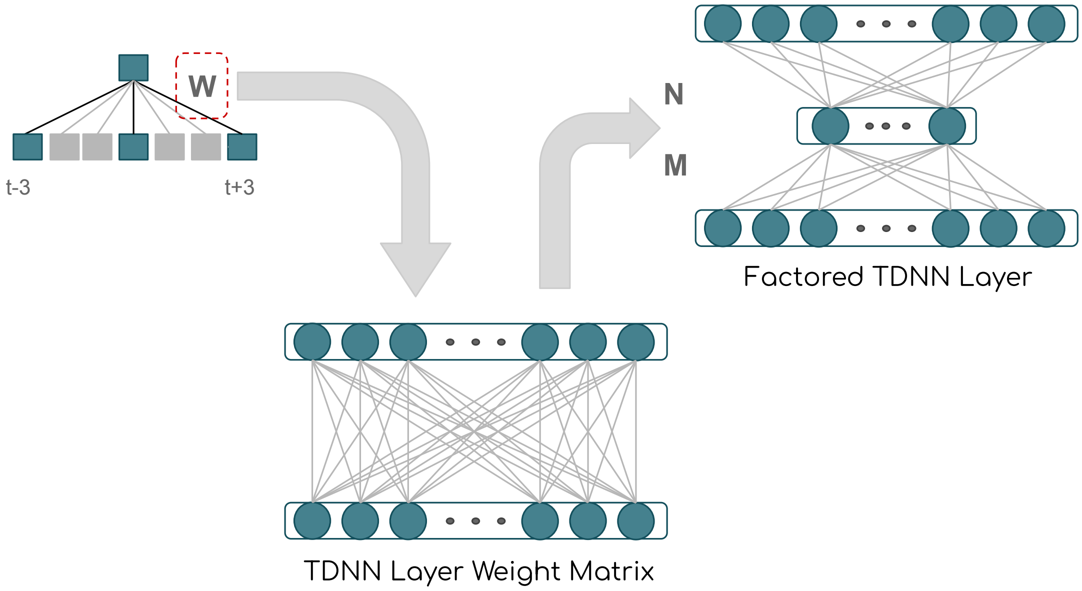

Figure 2.2: Factorized TDNN Layer

To train a TDNN-F acoustic model, a GMM-HMM model is trained to provide labeled data by aligning the frames to phone states. The hybrid model is then trained using the lattice-free maximum mutual information (LF-MMI) criterion, which maximizes the posterior probability of the transcript given the audio signal. It involves summing the joint probability over all possible word sequences allowed by the AM and LM in the system. MMI can be approximated by summing over the lattice instead of all possible sentences, but it is still computationally expensive. LF-MMI uses a phone-level language model instead of a word-level language model (backed by a lexicon), which allows a smaller frame rate, a faster decoding speed, and improvement in accuracy (Povey et al., 2016). We interpolate the LF-MMI loss with cross-entropy loss during training. An additional output layer was built to calculate the cross-entropy loss. Only the LF-MMI output layer is used during decoding.

2.3 Experiment Setup

All of the experiments were carried out using the Kaldi toolkit (Povey et al., 2011). In this Section, we provide details of the corpora, data augmentation process, and the two models we used in our experiment.

2.3.1 Corpora and Data Preparation

We trained and tested our models using 2 child speech corpora, CMU_Kids and CSLU_Kids. The details and statistics of both corpora can be found in Section 1.4. As shown in Table 2.1, we randomly reserved 30% of the utterances from each corpus for development testing. All of our models were trained on both of the corpora separately, as well as on the combined training set.

Table 2.1: Child Speech Corpora and Training/Test Splits

	CMU_Kids		**CSLU_Kids**		**Combined**	
	Train	Dev	Train	Dev	Train	Dev
# of Utterances	3621	1559	50026	21354	53647	2213
Duration (hours)	6.34	2.76	48.56	20.75	54.90	23.51

Since CMU_Kids is particularly small in size, and it is noisier than CSLU_Kids, we trained a TDNN-F model with augmented training set from CMU_Kids by adding babble noise and reverberation. Babble noise was created by combining 3 to 5 speakers from the Mixer-6 corpus (Cieri et al., 2006), and then added to the original training set. On top of additive babble noise, we simulated room reverberation using the RIRS_NOISES database (Ko et al., 2017). In total, we generated 2 augmented copies (babble noise and babble noise with reverberation) of the data and used them alongside the original, clean training data as the training set of the TDNN-F. Alignments used for the new augmented data come from their corresponding clean copies.

Mel-frequency cepstral coefficients (MFCC) were used as the front-end feature. We extracted MFCC features from a 25 ms window, and a frame rate of 10 ms. For the GMM-HMM system, 13 MFCCs and their corresponding Δ and Δ-Δ features were used. For TDNN-F system, high-resolution MFCCs were used. The input vector was a concatenation of 40-dimension cepstral coefficients of both the current and neighbor frames, and a 100-dimension i-vector (Dehak et al., 2011; Saon et al., 2013; Senior and Lopez-Moreno, 2014) of the current frame.

2.3.2 TDNN-F System

For the TDNN-F network, we followed the three-stage splicing structure[1]. As shown in Figure 2.3, we started by building a traditional 3-state left-to-right GMM-

[1] Adapted from the Kaldi-MATERIAL recipe: `https://github.com/kaldi-asr/kaldi/tree/master/egs/material/s5`

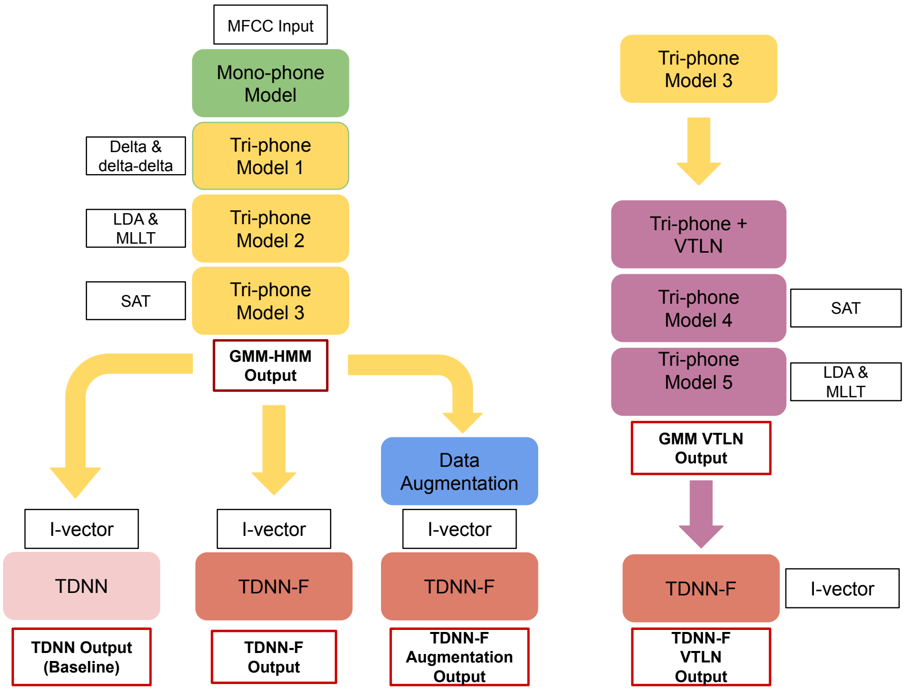

Figure 2.3: Feature Processing and Model Training Recipes

HMM triphone model to provide aligned (frame to phone state) training data for the DNN system. For better alignment, linear discriminant analysis (LDA) (Haeb-Umbach and Ney, 1992), maximum likelihood linear transformation (MLLT) estimation (Gales, 1999), and speaker adaptive training (SAT) (Anastasakos, McDonough, and Makhoul, 1997) are included.

Our model was then trained using the aligned data. High-resolution MFCC of the current (t) and neighboring ($t-1$, $t+1$) frames were concatenated with an i-vector of the current frame, and used as the input feature. After transferring the input feature into the hidden dimension (1024), 12 TDNN-F layers were used as the hidden layers. Each TDNN-F layer may be regarded as a large TDNN layer (with a dimension of 1024), followed by two bottleneck layers (equivalent of two small TDNN layers with a dimension of 256). All (large and small) hidden layers concatenate the current frame with either the left or right neighbor before forwarding it to the next layer.

On top of the hidden layers, two separate output layers were built: one for the LF-MMI objective function and another for the cross-entropy objective function. When training the model, the losses with respect to the two objective functions were interpolated, while in decoding, only the LF-MMI output was used.

2.3.3 Baseline System

To illustrate the effectiveness of TDNN-F, we also built a traditional TDNN as the baseline system.[2] The baseline network was constructed to have a comparable size (in number of parameters) as the TDNN-F. The baseline system has the same input as the TDNN-F system, and the same number of hidden layers (12). Each hidden layer is a TDNN layer with a dimension of 768, and a window size of 1 (on both sides). Since the input and output layer of the two networks are approximately the same, we compare the size of two networks by comparing the number of parameters in each hidden layer, and the baseline system is about 1.5 times in size as the TDNN-F system.

[2] Adapted from the Kaldi MiniLibriSpeech recipe: `https://github.com/kaldi-asr/kaldi/tree/master/egs/mini_librispeech/s5`

2.3.4 Language Model

The baseline model and TDNN-F model used the same lexicon and language model (LM). We used the CMU Pronunciation Dictionary as the lexicon, and the LM was a 3-gram model trained on LibriSpeech. An additional phone-level 4-gram LM was trained from the lexicon and used in the LF-MMI training for the TDNN-F.

2.4 Results and Discussion

All of our models described in Section 3 were tested with the datasets described in Section 3.1. Table 2.2 shows a comparison among all of our models. *Dev* denotes the development set from the same corpus as the training set, while *Test* denotes the development set from the combined set.

Table 2.2: WER (%) Comparison of Different Acoustic Models

	CMU_Kids		**CSLU_Kids**		**Combined**
	Dev	Test	Dev	Test	Dev/Test
GMM	29.6	75.9	32.5	38.4	36.5
GMM + VTLN	29.5	**74.9**	31.6	37.4	35.8
TDNN *(baseline)*	20.1	77.9	13.7	24.9	15.8
TDNN-F	**17.3**	77.0	**10.8**	22.4	**11.7**
TDNN-F + VTLN	17.6	78.2	13.3	**22.3**	11.9

2.4.1 Word Error Rate Analysis

The first row shows the word error rate (WER) result from the GMM-HMM system, which is used to align the training data for the baseline TDNN and TDNN-F system; the third and fourth rows show a comparison between the TDNN and the TDNN-F systems. By comparing these two rows, it can be seen that TDNN-F outperforms the baseline in almost all combinations of training and testing data. On the combined data set, the TDNN-F achieves a relative improvement of 26%.

Though TDNN-F shows great strength in the *Dev* sets, it is worth noticing that when the training set is extremely small, i.e., on CMU_Kids, TDNN-F has no such advantage against the GMM-HMM model on the *Test* set. Part of the reason is that TDNN-F, like all other neural models, suffers more from the insufficiency of training data due to the massive number of parameters. Another reason is that, comparing to CSLU_Kids, CMU_Kids has more noise. Data augmentation is introduced with the expectation that it would compensate for the insufficiency and mismatch of data. More discussion on augmentation follows in Section 4.2.

Since previous research (Tuerk and Robinson, 1993; Andreou, Kamm, and Cohen, 1994; Eide and Gish, 1996; Lee and Rose, 1998) has shown the effectiveness of VTLN on GMM-HMM models for recognizing child speech, we also explored applying VTLN along with TDNN-F. We find however, by comparing the first and last two rows in Table 2.2, that while VTLN does improve the WER performance of GMM-HMM model as reported, it does no have a significant effect on TDNN-F.

2.4.2 Performance on Small Dataset

To further study how models are effected when the training set is extremely small, i.e., around 6 hours in our case, we compare the performance of different models (GMM, TDNN, TDNN-F) trained on only CMU_Kids. We examine WER separately on the CMU_Kids and CSLU_Kids development sets to study the effect of matched versus mismatched noise conditions in training and test, showing how well the models generalize.

Table 2.3: WER (%) for Very Training Set and Mismatched Test Sets

	CMU_Kids	**CSLU_Kids***(Mismatched)*
GMM	29.5	**85.4**
TDNN *(baseline)*	20.1	89.8
TDNN-F	17.3	89.1
TDNN-F + Aug	**16.0**	89.0

As seen in Table 2.3, GMM outperforms both neural models in the mismatched scenario as expected because it has less parameters. Though showing no significant advantage over TDNN in the mismatched setting, augmentation does compensate for the shortage in training data, and improve the WER on the matched test set.

2.4.3 Error Analysis

Table 2.2 shows that TDNN-F is generally a better model than the baseline, but it gives limited information about the kind of mistakes that the models made. Table 2.4 presents the details of the three types of mistakes measured in WER: insertion, deletion and substitution. For each model, the first row shows the number of mistaken words that fall into each category, followed by the percentage of each category in the total WER in the second row. For TDNN-F, we also show their relative improvement compared to the baseline.

Table 2.4: ASR Error Breakdown of Child Speech

	-	Total	Ins	Del	Sub
TDNN	# Errors	11560	2168	3210	6182
(baseline)	Fraction (%)	—	18.8	27.8	53.5
TDNN-F	# Errors	8552	1681	2794	4077
	Fraction (%)	—	19.7	32.7	47.7
	Improvement (%)	26	22	13	34

Compared to ASR systems for adults that have a similar WER as our TDNN-F system, a larger portion of the total word errors in child speech comes from insertions and deletions. For example, in one of the example recipes for LibriSpeech from Kaldi, the total WER is around 12%, and substitutions make up 80% of the errors. This is not surprising, considering that both the test sets we use contain children trying to read, and most of them are still learning how to read. Here is an example transcript given by the TDNN-F system and its reference transcript provided by the CMU_Kids corpus:

> **TDNN-F** : If a lightning storm comes there are four things you can do to *say stay sick help stay help* stay safe.
> **Reference**: If a lightning storm comes there are four things you can do to stay safe.

When measuring WER, this sentence is ruled to have 6 insertion errors. But when we listened to the recording, we found that the speaker was clearly struggling with the phrase “stay safe”, as she attempted it multiple times, and hesitated between saying “stay health (healthy)” and “stay safe”. This points to potential transcription issues and their impact, albeit likely to be small, on the measured WER in the two corpora we are using.

2.5 Summary

By comparing the factorized TDNN with different traditional and state-of-the art systems, we demonstrate the efficacy of TDNN-F for the task of automatically recognizing child speech. We build a TDNN-F system that outperforms its alternatives in datasets with various sizes. We explored the impact of vocal track length normalization (VTLN) and data augmentation on the performance of TDNN-F systems. Though effective with traditional models like GMM-HMM, VTLN has no significant impact on TDNN-F. When trained with an extremely small dataset, data augmentation helps improve the performance of TDNN-F on test data in the same channel condition as the training set.

Chapter 3

End-to-end Child Speech Recognition

3.1 Introduction

Inspired by seq2seq (Sutskever, Vinyals, and Le, 2014; Bahdanau, Cho, and Bengio, 2014) and transformer-based models (Vaswani et al., 2017) , two popular purely neural sequence-to-sequence models first introduced to tackle the task of machine translation (MT), recent effort in en-to-end style ASR systems has shown success in large corpora (Graves and Jaitly, 2014; Chorowski et al., 2014; Wang et al., 2019; Chan et al., 2016; Dong, Xu, and Xu, 2018; Karita et al., 2019). In low-resource scenario, however, purely neural models suffers even more than the hybrid models from the data shortage. Previous work tried to address this by multitask learning. Zhou, Xu, and Xu (2018) performed multilingual training by annotating the utterances in different languages with the same sub-word unit set, so that these extra data can be utilized to train the model for low-resource ASR.

This chapter aims to explore applying end-to-end ASR systems to child speech recognition, and to tackle the low-resource problem by using transfer learning techniques to fine-tune end-to-end ASR models pretrained on sufficient adult speech data. The rest of this chapter is organized as follows. Section 3.2 describes the 'seq2seq with attention' model in details, along with its modification for ASR and transfer learning methods used to adapt a pretrained model for child speech recognition.

3.2 Model Structure

End-to-End ASR has developed a few different flavors in the recent years. In this section, we focus on the most prevalent recurrent seq2seq with attention structure, and transfer learning methods to adapt pretrained seq2seq ASR model.

3.2.1 Seq2seq with Attention Model in MT

First introduced to the MT community, seq2seq model with attention mechanism (Sutskever, Vinyals, and Le, 2014; Bahdanau, Cho, and Bengio, 2014) has shown strength in the task of transducing a sentence in the source language to its translation in the target language. Both the input and output sentences are considered as a sequence of words, and hence the name, seq2seq. This model has an encoder-decoder structure, where the core component of both parts is an RNN. In this thesis, we use Long-Short Term Memory (LSTM) for its ability to handle longer sequences than regular RNN.

Figure 3.1 illustrates the seq2seq model with attention mechanism for MT. On the encoder side, each word in the input sentence is first transformed by a linear layer into a fixed-size vector, called word embedding. The LSTM layers on the top of the embedding layers, taking one word at a time, encodes the word based on the embedding and its current status, and then update the status for future steps [1]. As a result, the encoder outputs a sequence of encoded vectors,$\mathbf{h}_{1:l}$, with l being the number of words in the input sentences. On the decoder side, at each time step t, the LSTM takes the output and hidden state from the previous step $t-1$, and computes its current hidden state $\vec{s}_t$. The attention module uses $\vec{s}_t$ as the query, the encoder-side hidden states, $\mathbf{h}_{1:L}$ as the keys and values, and outputs the final hidden representation as a weighted sum of the values, where the weights are computed using some scoring function that considers the relevance between the query and each keys[2]. Finally, a linear layer is used to transform the output of the attention module to a probability over the vocabulary in the target language.

[1] Refer to Appendix A for more details about LSTM

[2] Refer to Appendix B for more details about attention

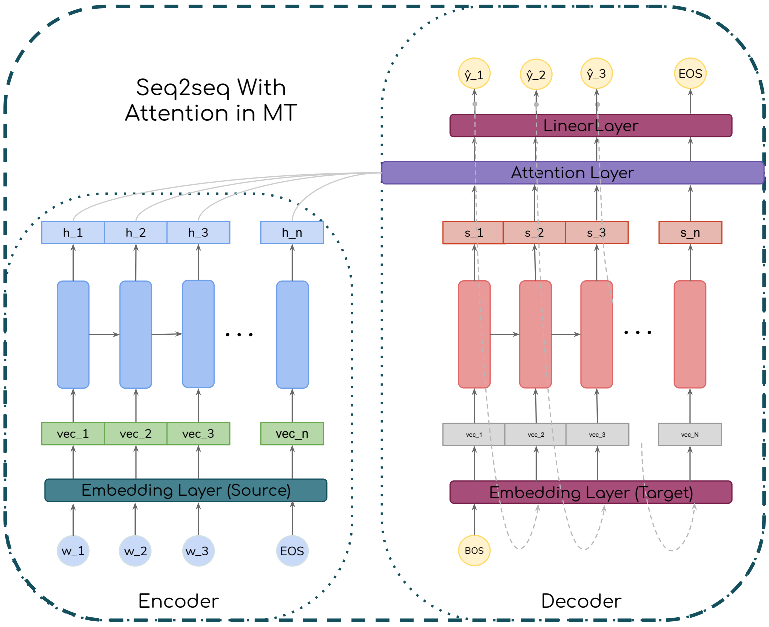

Figure 3.1: Seq2seq Model for MT

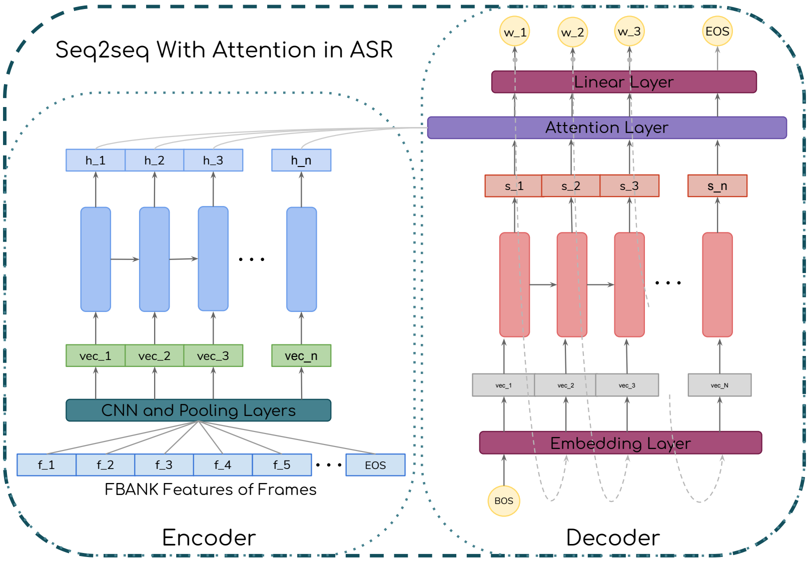

Figure 3.2: Seq2seq Model for ASR

In training time, the decoder takes the input word, embedded by the target language embedding layer, from the gold translation provided in the training data, instead of the previous output predicted by itself. All the components in this model can be trained using a cross entropy loss between the predicted and gold translation.

3.2.2 Seq2seq with Attention Model in ASR

Similar to MT, ASR can be describe as a sequence-to-sequence modeling task: to "translate" a sequence of utterance frames to its transcript as a sequence of words. However, in order to apply the seq2seq with attention model to ASR, some modifications are needed.

Firstly, the input sequence in ASR is a sequence of utterance frames represented as feature vectors. The number of frames in an utterance is normally much larger than the number of words in a sentence. Long sequences are known to be challenging to RNNs (Sutskever, Vinyals, and Le, 2014; Bahdanau, Cho, and Bengio, 2014). In order to alleviate the impact of long sequences to LSTM, the word embedding layers are replaced by a few CNN layers and optionally pooling layers to down sample the input sequence before sending it to the LSTM(Chorowski et al., 2014).

Secondly, the decoder in the seq2seq model can be considered as a small LM trained exclusively on the training data. It is hard to get as many words as an MT corpus form the transcripts of an ASR corpus, which is a great disadvantage for training the decoder. Though not favorable when introduced to the MT community

(Gulcehre et al., 2015), the idea of using extra LM trained on extra data to help decoding gained its popularity in the speech community. Chorowski and Jaitly (2016), Kannan et al. (2018), and Hori, Cho, and Watanabe (2018) showed that incorporating extra LM significantly improve the performance of end-to-end ASR. In this chapter, we include the external LM by using shallow-fusion (Gulcehre et al., 2015), defined as below:

$$y_t* = \underset{y_t \in \mathcal{V}}{\operatorname{argmax}} \log p_{ASR}(y_t|\mathbf{x}, \mathbf{y_{1:t-1}}) + \lambda \log p_{LM}(y_t|\mathbf{y_{1:t-1}}) \tag{3.1}$$

At each time stamp t, the prediction, y_t*, is made by interpolating the distribution over vocabulary ($\mathcal{V}$), given by the seq2seq ASR model ($\log p_{ASR}(y_t|\mathbf{x}, \mathbf{y_{1:t-1}})$), and the same distribution given by the external LM ($\log p_{LM}(y_t|\mathbf{y_{1:t-1}})$. The external LM weight, λ, is tuned as a hyper parameter.

3.2.3 Fine-tuning Pretrained seq2seq Model for Child Speech Recognition

Though successful and achieved state-of-the-art results in large corpora, such as LibriSpeech, seq2seq model does not work well in smaller ones. The WSJ corpus (Garofalo et al., 1993; “CSR-II (WSJ1) complete” 1994), for example, has around 200 hours of data, on which none end-to-end ASR system could outperform the hybrid models yet. Our child speech dataset has less than half of training data in WSJ corpus, and we noticed in preliminary experiments that well established seq2seq with attention ASR models did not converge on the child speech data sets, and resulted in a WER over 90%. To developed workable end-to-end ASR system for child speech recognition, we fine-tuned two seq2seq models trained on two adult speech corpora respectively, LibriSpeech and WSJ. The significant difference in model sizes and amount of pre-training data also create an interesting caparison.

Instead of fine-tuning the entire seq2seq with attention model, it is interesting to consider the function of each component, and fine-tune some components accordingly. Prior to the seq2seq structure, predominant end-to-end ASR was to use an RNN plus a connections temporal classification (CTC) (Graves et al., 2006; Graves and Jaitly, 2014), which can be regarded as a seq2seq model but with the decoder replaced by a CTC module. CTC-based ASR models have worse performance than the seq2seq models in general, and it heavily relies on external LM re-scoring for better performance. This suggests that the decoder can be regarded as a weaker and conditional LM, which is trained, exclusively on the transcripts of the speech corpus, to generate a sentence given the acoustic feature encoded by the encoder. The transcripts in a speech corpus normally have fewer words than an MT corpus, which might be part of the reasons why external LMs do not significantly facilitate seq2seq MT, but are effective on seq2seq for ASR. In low-resource scenario like child speech recognition, it might be worthwhile to freeze all the parameters in decoder and relies on the external LM when decoding. In addition, the fact that CTC-based models can work, though less satisfactory, without external LM means that the encoder LSTM might also have the power to model the linguistic content (syntactic, and even semantic features) instead of only the acoustic features. So we also tried freezing the LSTM in the encoder to see if it improves the overall performance in low-resource scenario. To make sure the external LM is not completely out-of-domain, we fine-tuned the external LM with the transcripts of the child speech training set.

3.3 Experiment Setup

Experiments in this chapter are implemented using Espresso (Wang et al., 2019), an end-to-end ASR toolkit. We pretrained two seq2seq with attention models, one pretrained on LibriSpeech data, and the other on WSJ data. As the amount of data in the two corpora differs, the two models are also different in sizes. Two external LM are trained on these two corpora respectively. We follow the exemplary recipes[3] for LibriSpeech and WSJ provided in Espresso to train the seq2seq models and their corresponding external LMs.

3.3.1 Data and Feature Extraction

We used the same child speech corpora, CMU_Kid and CSLU_Kids, as described in section 1.4. The train-dev-test split is different than the one used in Chapter

[3] https://github.com/freewym/espresso/tree/master/examples

2. RNN decoders are powerful but also prone to overfitting with small datasets. If there are overlapping sentences in the dev and train set, the WER result will not be an accurate evaluation of the model. Both corpora we use have a limited set of sentences, and one sentence might be read by different speakers for multiple times. So we preserved 15% of sentences in each corpus for test set, 15% for dev set, and 70% for train set to make sure that no two utterances in two different sets will have the same transcript. Table 3.1 shows more details about the train, dev and test set used in this chapter.

Table 3.1: Train-Dev-Test Split for End-to-End Child Speech Recognition

Subset	**Number of Sentences**	**Duration (h)**
Train	481	52.81
Dev	104	13.34
Test	89	12.26

For all the experiments in this chapter, we use the Mel frequency filter bank (fbank) features and pitch feature extracted by Kaldi. Features are extracted for each 25ms-wide frame, with a stride of 10ms. For the fbank feature, 80 Mel bins are used, which gives a 80-dimensional vector as the fbank feature. It is then concatenated with a 3-dimensional pitch features.

3.3.2 External Language Model

External LMs for both corpora are LSTM language models. Details of both models are shown in Table 3.2.

Table 3.2: External LSTM Language Models

External LM	Vocabulary Unit	Vocabulary Size	LSTM Layers	Hidden Size	Total Parameters	Training Data
LM-Lib	BPE	5000	4	800	24.5M	102M BPEs
WordLM-WSJ	65000	3	1200	112.6M	37.9M words	

The LM trained on LibriSpeech (LM-Lib) uses Byte Pair Encoding (BPE) (Sennrich, Haddow, and Birch, 2016) as the vocabulary units, and has a vocabulary size of 5,000 BPEs. This model has 4 layers of LSTM, with a hidden size of 800 dimension, result around 24.5 million parameters in total. It is trained on the transcripts of the training set provided in the LibriSpeech corpus, which has around 102 million BPEs. To use this LM by shallow-fusion, the decoder in the seq2seq model has to use the same BPE vocabulary.

The LM trained on WSJ (WordLM-WSJ) uses words as the vocabulary unit, and has a vocabulary size of 65,000 words. The model has 3 LSTM layers with hidden size of 1,200, and around 112.6 million parameters in total. It is trained on the transcripts of the training set in WSJ, containing around 37.9 million words in total. To incorporate this LM for seq2seq ASR decoding, the decoder can use a character vocabulary, and fuse with the LM using the look-ahead word-based LM fusion (Hori, Cho, and Watanabe, 2018).

3.3.3 Seq2seq with Attention Models

We pretrained two seq2seq attention ASR models, seq2seq-Lib and seq2seq-WSJ, using the LibriSpeech and WSJ corpus respectively. Both models take sequence of fbank features as the encoder input. The pre-LSTM CNN layers are the same for both models. Four 2-dimensional CNN layers were used to down sample the input sequence to one fourth of the original length in both dimensions, namely the feature dimension and time dimension. As a result, the 83-dimensional input sequence is transformed into 128 channels of 20-dimensional vector sequence that is about a quarter of its original length. For the encoder, seq2seq-Lib has 4 layers of bi-directional LSTM with hidden size of 1024, while seq2seq WSJ uses 3 layers of bi-directional LSTM with a hidden size of 320. For the attention decoder, both model use Bahdanau attention Bahdanau, Cho, and Bengio (2014). seq2seq-Lib has 3 layers of LSTM in its decoder, with a hidden size of 1024; and seq2seq-WSJ has 3 layers of LSTM with a hidden size of 320 in the decoder. In order to allow shallow-fusion with the corresponding external LM, seq2seq-Lib decoder has a vocabulary of 5000 BPEs, and seq2seq-WSJ decoder has a character level vocabulary. A more detailed model structure can be found at `https://github.com/freewym/espresso/blob/master/espresso/models/speech_lstm.py`.

3.3.4 Fine-tuning of Pretrained Models

We tuned the external LMs and seq2seq models on the training set of CMU_Kids and CSLU_Kids. To see how different components affect the seq2seq models, we tried fine-tune the entire model, only the encoder, and only the CNN layers on both models. ?? shows the details of the 6 fine-tuned seq2seq models.

Table 3.3: Fine-tuning Seq2seq Models

Model Structure	Model	Frozen Component	Initial Learning Rate
seq2seq-Lib	Lib-Unfrozen	None	$2.5e^{-5}$
	Lib-Dec	Decoder	
	Lib-DecEnc	Decoder+Encoder-LSTM	
seq2seq-WSJ	WSJ-Unfrozen	None	$3e^{-4}$
	WSJ-Dec	Decoder	
	WSJ-DecEnc	Decoder+Encoder-LSTM	

3.4 Results and Analysis

3.4.1 Fine-tuning External Language Model

Table 3.4: Perplexity on Dev Set Before and After Fine-tuning LM

	Before		**After**	
	Original Dev	Kids Dev	Original Dev	Kids Dev
Word LM (WSJ)	71	253	98	114
BPE LM (LibriSpeech)	37	1794	171	38

As discuss in section 3.1, ASR corpora normally has less written data then MT corpora, and external LMs trained on extra data help improve ASR results. Before we apply pre-trained LMs to out End2end child speech system, we also fine-tuned the external LMs described in 3.3 with the train set of CMU_Kids and CSLU_Kids data. Table 3.4 shows the perplexity on the child speech dev set (Kids Dev in the table) before and after fine-tuning. For reference, we also include the perplexity on the original dev sets (WSJ-test_dev93 and LibriSpeech-dev) used to train the LMs. As shown in the table, both LMs trained on adult speech text performs poorly on the child speech text without fine-tuning, especially when compared to the perplexity on their original dev set. After fine-tuning, the BPE LM achieves a comparable perplexity on child speech dev set (38) as on its original dev set (37). The Word LM, however, though much improved after fine-tuning (253 → 114), the performance on child speech dev set is still significantly worse than on the original dev set (71).

We then used the tuned LMs to help the tuned seq2seq models using shallow-fusion when decoding. Figure 3.3 shows different LM weights and the WER results on the child speech dev set. All curves have a U-shape, suggesting that the external LM can help decoding to some degree, and the optimal LM weight can be found by hyper-parameter tuning.

3.4.2 Effect of Freezing Model Components

Figure 3.3 also shows how freezing different components responds to different external LM-weight in WER performance. For both structures, a large LM weight

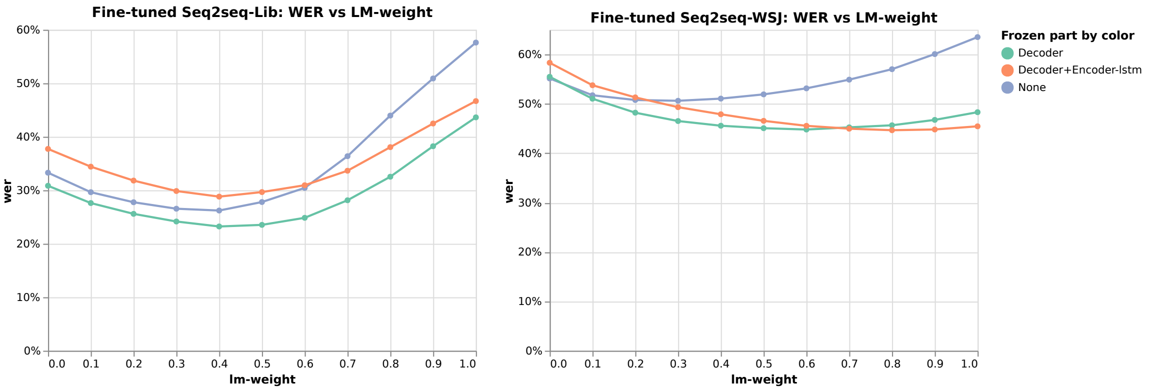

Figure 3.3: WER vs LM-weight

hurts the performance of unfrozen model the most. For seq2seq-WSJ, external LM almost did not help the unfrozen model—the optimal LM weight is 0.1. For seq2seq-Lib, there is an abrupt performance drop when LM weight reaches 0.5 for the unfrozen model, while the other two has a more smooth and consistent increase in WER when LM weight is getting too large. We hypothesize that this is because the seq2seq model, especially the decoder, is over-fitted to the training data, so when forced to use an external LM, the performance of the unfrozen models start to fall apart. While freezing the decoder during fine-tuning does improve overall performance for both structures, it is interesting to see that freezing both decoder and encoder-LSTM has a comparable performance. The WER difference between Lib-Dec and Lib-DecEnc decreases when the LM weight goes up, and WSJ-DecEnc even slightly outperform WSJ-Dec at the optimal LM weight point. This suggests that the encoder LSTM might also learned non-acoustic information and functions as a weak LM. When have limited data and small model, freezing both decoder and encoder LSTM can prevent over-fitting.

Table 3.5: WER Results of Fine-tuned Models

Fine-tuned Model	**Dev (WER %)**	**Test (WER %)**
Lib-Unfrozen	26.2	26.9
Lib-Dec	**23.2**	**24.2**
Lib-DecEnc	28.8	29.6
WSJ-Unfrozen	50.6	53.2
WSJ-Dec	44.8	**50.5**
WSJ-DecEnc	**44.7**	50.7

Table 3.5 shows the WER result on dev and test set of all fine-tuned models, using optimal LM-weights obtained on the dev set. For both pretrained models, freezing some components during fine-tuning outperforms fine-tuning the entire models without freezing any parameters. With large pre-trained model (seq2seq-Lib), freezing only the decoder gives the best performance (Lib-Dec) on both dev and test set. The WER on dev and test set are close, suggesting that the model generalizes well. When the pre-trained model is small (seq2seq-WSJ), the WER on test set is around 3–6% higher than the WER on dev set, which means the model is probably over-fitting. In such case, freezing just the decoder and freezing both decoder and encoder LSTM have about the same performance.

3.4.3 Error Analysis

Figure 3.4 breaks down the WER, and shows the substitution, insertion and deletion error by reference sentence length. All six models have a particularly high WER on the one-word utterances. Part of the reason is that decoder and external LM model language as a sequence of words, and are not able to help much the decoding when the sentence has only one word. Another reason is that the compared to WSJ and LibriSpeech, utterances in child speech corpora are much shorter, respecting the speakers' reading and comprehension ability. The latter also affects the number of insertion and deletion errors when using different fine-tuning strategies. Freezing more parameters during fine-tuning forced the model to stay closer to its pretraining adult speech data, while fine-tuning the entire model allows to lean towards the child speech data. As a result, models fine-tuned with encoder LSTM and decoder frozen (WSJ-DecEnc and Lib-DecEnc) have more insertion errors than the other models, and the models fine-tuned with no components frozen (WSJ-Unfrozen and Lib-Unfrozen) have more deletion errors than the others.

3.5 Summary

In this chapter, we explore applying end-to-end models to the task of child speech recognition. Though they perform poorly when trained with only child speech, seq2seq with attention can be improved by first training on sufficient adult speech data, and then fine-tuning on small child speech dataset. The external LM used to improve the pretrained model also needs to be tuned on child speech transcripts. The performance can be further improved by freezing some components during the fine-tuning stage to avoid overfitting. With a large pretrained model, freezing the decoder gives the best result, but with smaller pretrained models, freezing decoder and freezing both decoder and encoder LSTM have similar performance.

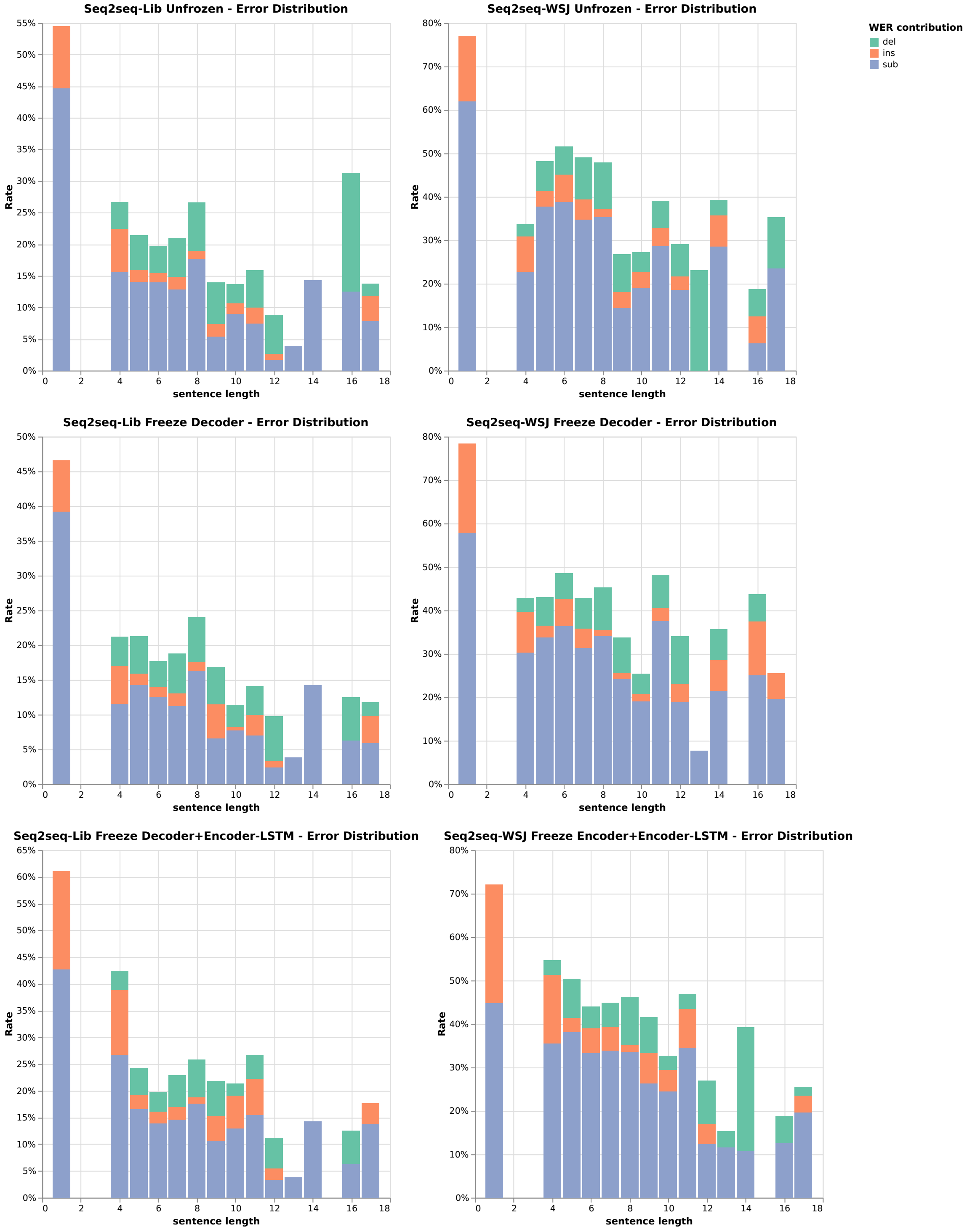

Figure 3.4: Error Distribution by Sentence Length

Chapter 4

Conclusion and Future Work

4.1 Conclusion

In this thesis, we explored child speech recognition as a low-resource Automatic Speech Recognition (ASR) problem. Hybrid and end-to-end systems were built and tested on two publicly available child speech corpora, CMU_Kids and CSLU_Kids.

By adopting a data-efficient model structure, a factorized time delay neural network (TDNN-F), we successfully improved the performance of the hybrid system, and achieved state-of-the-art result on the above-mentioned corpora. The recipe for training this hybrid system has been made available as a recipe of Kaldi [1] for future research. By using transfer learning techniques, we utilized adult speech data for training end-to-end ASR models for child speech, and built the first working end-

[1] Available at `https://github.com/kaldi-asr/kaldi/tree/master/egs/cmu_cslu_kids`

to-end system for child speech recognition. We also demonstrated that freezing the decoder in Seq2seq model during the fine-tuning stage helps prevent over-fitting and improves the overall performance.

4.2 Future Work

As future research, we would like to address the problem of transcription errors and imperfect readings in the audio data as discussed in section 2.4.3. A biased language model, or a sub-word-level adjustment might be some good directions to go for identifying and cleaning up the reference transcripts of child speech. Another issue we would like to explore is to tackle the problem of over-short sentences. Child speech utterance are normally much shorter than adult speech in sentence length, but common language models (LM) for hybrid and end-to-end systems are mainly trained on adult speech transcripts or even written text. Such LMs have limited power to model short transcripts in child speech data. By adding a classifier for short utterances, we can apply different LMs for short and long utterances, which can contribute to an over-all performance improvement.

Appendix A

Mathematical Details of LSTM

Long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) is widely used to solve sequence modeling problems. In seq2seq models for ASR, LSTM is used to encode a sequence of input vectors, $\mathbf{x} = (\vec{x_t})_{t=1}^{t=l}$, resulting in a sequence of hidden vector, $\mathbf{h} = (\vec{h_t})_{t=1}^{t=l}$. For simplicity, we define $\mathrm{L}_*(\cdot)$ to be the forward function of a linear transformation with learnable weight matrix W_*, and bias b_*, i.e. $\mathrm{L}_*(\vec{x}) = W_* \cdot \vec{x} + b_*$.

Figure A.1 (Olah, 2015) illustrates the details of an LSTM cell and how it works on the input sequence. At each time step t, the input vector $\vec{x_t}$ is first transformed into a candidate update, $\widetilde{\vec{c_t}}$. The previous cell state, $\vec{c}_{t-1}$ is then updated based on the candidate update, forget factor f_t, and input factor i_t. Finally, previous hidden state h_{t-1} is updated. Equation A.1 shows how LSTM updates cell state $\vec{c_t}$ and hidden state $\vec{h}_t$.

$$\widetilde{\vec{c}_t} = \tanh(\mathrm{L}_C([\vec{h}_{t-1}, \vec{x}_t])) \tag{A.1a}$$

$$f_t = \sigma(\mathrm{L}_f([\vec{h}_{t-1}, \vec{x}_t])) \tag{A.1b}$$

$$i_t = \sigma(\mathrm{L}_i([\vec{h}_{t-1}, \vec{x}_t])) \tag{A.1c}$$

$$\vec{c}_t = f_t * \vec{c}_{t-1} + i_t * \widetilde{\vec{c}_t} \tag{A.1d}$$

$$o_t = \sigma(\mathrm{L}_o([\vec{h}_{t-1}, \vec{x}_t])) \tag{A.1e}$$

$$h_t = o_t * \tanh(\vec{c}_t) \tag{A.1f}$$

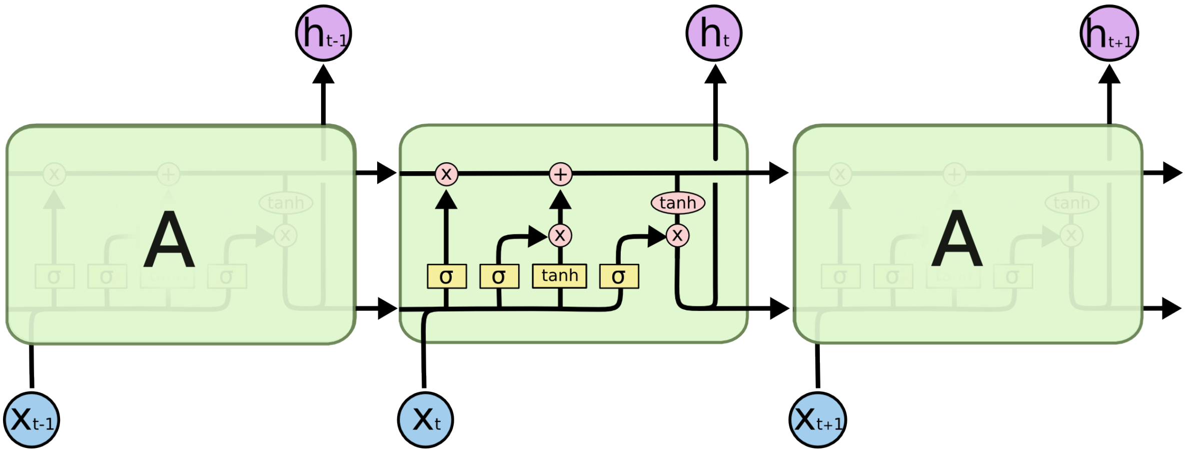

Figure A.1: LSTM Cell (Olah, 2015)

Appendix B

Mathematical Details of Attention Mechanism

Bahdanau attention (Bahdanau, Cho, and Bengio, 2014) was proposed to improve the seq2seq model by introducing context vector to the decoder side. The context vector is computed as a weighted sum of all hidden states on the encoder structure, $\mathbf{h}_{1:l}$. The weight at decoder time step i for the encoder hidden state $\vec{h}_t$, α_{it}, is a normalized score between the encoder hidden state $\vec{h}_t$ and the decoder hidden state $\vec{s}_i$, and the score is given by a learnable linear layer, $\mathrm{L_a}(\cdot)$ and a learnable vector $\vec{v}$. Equation B.1 show the computation of context vector $\vec{c_i}$.

$$e_{it} = \vec{v} \cdot \tanh(\mathrm{L}_a([\vec{s}_i, \vec{h}_t])) \tag{B.1a}$$

$$\alpha_{it} = \frac{\exp e_{it}}{\sum_t \exp e_{it}} \tag{B.1b}$$

$$\vec{c}_i = \sum_t \alpha_{it} \vec{h}_t \tag{B.1c}$$

Bibliography

Anastasakos, T., J. McDonough, and J. Makhoul (1997). "Speaker adaptive training: a maximum likelihood approach to speaker normalization". In: *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing.* Vol. 2, 1043–1046 vol.2.

Andreou, Andreas G., Terri Kamm, and Judith Levy Cohen (1994). "Experiments in vocal tract normalization". In:

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473.*

Batliner, Anton, Mats Blomberg, Shona D'Arcy, Daniel Elenius, Diego Giuliani, Matteo Gerosa, Christian Hacker, Martin Russell, Stefan Steidl, and Michael Wong (2005). "The PF_STAR children's speech corpus". In: *Ninth European Conference on Speech Communication and Technology.*

Beckman, Mary E, Andrew R Plummer, Benjamin Munson, and Patrick F Reidy (2017). "Methods for eliciting, annotating, and analyzing databases for child speech development". In: *Computer speech & language* 45, pp. 278–299.

Chan, William, Navdeep Jaitly, Quoc Le, and Oriol Vinyals (2016). "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)*. IEEE, pp. 4960–4964.

Chorowski, Jan, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio (2014). "End-to-end continuous speech recognition using attention-based recurrent nn: First results". In: *arXiv preprint arXiv:1412.1602*.

Chorowski, Jan and Navdeep Jaitly (2016). "Towards better decoding and language model integration in sequence to sequence models". In: *arXiv preprint arXiv:1612.02695*.

Cieri, Christopher, Walt Andrews, Joseph P Campbell, George Doddington, Jack Godfrey, Shudong Huang, Mark Liberman, Alvin Martin, Hirotaka Nakasone, Mark Przybocki, et al. (2006). *The mixer and transcript reading corpora: Resources for multilingual, crosschannel speaker recognition research.* Tech. rep. MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB.

Claus, Felix, Hamurabi Gamboa Rosales, Rico Petrick, Horst-Udo Hain, and Rüdiger Hoffmann (2013). "A survey about databases of children's speech." In: *INTERSPEECH*, pp. 2410–2414.

"CSR-II (WSJ1) complete" (1994). In: *Linguistic Data Consortium, Philadelphia, vol. LDC94S13A*.

Dehak, N., P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet (2011). “Front-End Factor Analysis for Speaker Verification”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4, pp. 788–798.

Deoras, Anoop, Tomáš Mikolov, Stefan Kombrink, Martin Karafiát, and Sanjeev Khudanpur (2011). “Variational approximation of long-span language models for LVCSR”. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, pp. 5532–5535.

Dong, Linhao, Shuang Xu, and Bo Xu (2018). “Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, pp. 5884–5888.

Eide, Ellen and Herbert Gish (1996). “A parametric approach to vocal tract length normalization”. In: *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings.* Vol. 1. IEEE, pp. 346–348.

Eskenazi, M and J Mostow (2006). *The CMU KIDS Speech Corpus (LDC97S63).*

Eskenazi, Maxine, Jack Mostow, and David Graff (1997). “The CMU kids corpus”. In: *Linguistic Data Consortium* 11.

Evanini, Keelan and Xinhao Wang (2013). “Automated speech scoring for non-native middle school students with multiple task types.” In: *INTERSPEECH*, pp. 2435–2439.

Fainberg, Joachim, Peter Bell, Mike Lincoln, and Steve Renals (2016). “Improving Children’s Speech Recognition Through Out-of-Domain Data Augmentation”. In: *INTERSPEECH*, pp. 1598–1602.

Gales, M. J. F. (1999). “Semi-tied covariance matrices for hidden Markov models”. In: *IEEE Transactions on Speech and Audio Processing* 7.3, pp. 272–281.

Garofalo, John, David Graff, Doug Paul, and David Pallett (1993). “CSR-I (WSJ0) Complete”. In: *Linguistic Data Consortium, Philadelphia.*

Gerosa, Matteo, Diego Giuliani, Shrikanth Narayanan, and Alexandros Potamianos (2009). “A review of ASR technologies for children’s speech”. In: *Proceedings of the 2nd Workshop on Child, Computer and Interaction*, pp. 1–8.

Ghoshal, A., P. Swietojanski, and S. Renals (2013). “Multilingual training of deep neural networks”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7319–7323.

Goodman, Joshua (2001). “A bit of progress in language modeling”. In: *arXiv preprint cs/0108005.*

Graves, Alex, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber (2006). “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks”. In: *Proceedings of the 23rd international conference on Machine learning*, pp. 369–376.

Graves, Alex and Navdeep Jaitly (2014). “Towards end-to-end speech recognition with recurrent neural networks”. In: *International conference on machine learning*, pp. 1764–1772.

Gulcehre, Caglar, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio (2015). “On using monolingual corpora in neural machine translation”. In: *arXiv preprint arXiv:1503.03535*.

Haeb-Umbach, R. and H. Ney (1992). “Linear discriminant analysis for improved large vocabulary continuous speech recognition”. In: *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 13–16 vol.1.

Hayashi, Tomoki, Shinji Watanabe, Yu Zhang, Tomoki Toda, Takaaki Hori, Ramon Astudillo, and Kazuya Takeda (Feb. 2019). “Back-Translation-Style Data Augmentation for end-to-end ASR”. English. In: *2018 IEEE Spoken Language Technology Workshop, SLT 2018 - Proceedings*. 2018 IEEE Spoken Language Technology Workshop, SLT 2018 - Proceedings. 2018 IEEE Spoken Language Technology Workshop, SLT 2018 ; Conference date: 18-12-2018 Through 21-12-2018. Institute of Electrical and Electronics Engineers Inc., pp. 426–433.

Hermansky, Hynek (1990). “Perceptual linear predictive (PLP) analysis of speech”. In: *the Journal of the Acoustical Society of America* 87.4, pp. 1738–1752.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural computation* 9.8, pp. 1735–1780.

Hori, T., J. Cho, and S. Watanabe (2018). “End-to-end Speech Recognition With Word-Based Rnn Language Models”. In: *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 389–396.

Hori, Takaaki, Jaejin Cho, and Shinji Watanabe (2018). “End-to-end speech recognition with word-based RNN language models”. In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, pp. 389–396.

Hu, H., T. Tan, and Y. Qian (2018). “Generative Adversarial Networks Based Data Augmentation for Noise Robust Speech Recognition”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5044–5048.

Huang, Jui-Ting, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong (2013). “Cross-language Knowledge Transfer using Multilingual Deep Neural Network with Shared Hidden Layers”. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

Jaitly, Navdeep and Geoffrey E Hinton (2013). “Vocal Tract Length Perturbation (VTLP) improves speech recognition”. In: *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*. Vol. 117.

Jelinek, F. (1976). “Continuous speech recognition by statistical methods”. In: *Proceedings of the IEEE* 64.4, pp. 532–556.

Kannan, Anjuli, Arindrima Datta, Tara N. Sainath, Eugene Weinstein, Bhuvana Ramabhadran, Yonghui Wu, Ankur Bapna, Zhifeng Chen, and Seungji Lee (2019). “Large-Scale Multilingual Speech Recognition with a Streaming End-to-End Model”.

In: *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019.* Ed. by Gernot Kubin and Zdravko Kacic. ISCA, pp. 2130–2134.

Kannan, Anjuli, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhijeng Chen, and Rohit Prabhavalkar (2018). "An analysis of incorporating an external language model into a sequence-to-sequence model". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, pp. 1–5828.

Karita, Shigeki, Xiaofei Wang, Shinji Watanabe, Takenori Yoshimura, Wangyou Zhang, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, and Ryuichi Yamamoto (2019). "A Comparative Study on Transformer vs RNN in Speech Applications". In: *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019, Singapore, December 14-18, 2019.* IEEE, pp. 449–456. URL: https://doi.org/10.1109/ASRU46091.2019.9003750.

Kim, Chanwoo, Minkyu Shin, Abhinav Garg, and Dhananjaya Gowda (2019). "Improved vocal tract length perturbation for a state-of-the-art end-to-end speech recognition system". In: *Proc. Interspeech.* Vol. 2019, pp. 739–743.

Ko, Tom, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur (2017). "A study on data augmentation of reverberant speech for robust speech recognition". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, pp. 5220–5224.

Lee, L. and R. Rose (1998). “A frequency warping approach to speaker normalization”. In: *IEEE Transactions on Speech and Audio Processing* 6.1, pp. 49–60.

Leung, W., X. Liu, and H. Meng (2019). “CNN-RNN-CTC Based End-to-end Mispronunciation Detection and Diagnosis”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8132–8136.

Levinson, S. E., L. R. Rabiner, and M. M. Sondhi (1983). “An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition”. In: *The Bell System Technical Journal* 62.4, pp. 1035–1074.

Li, Qun and Martin J. Russell (2002). “An analysis of the causes of increased error rates in children’s speech recognition”. In: *INTERSPEECH.* 7.

Liao, Hank, Golan Pundak, Olivier Siohan, Melissa K. Carroll, Noah Coccaro, Qi-Ming Jiang, Tara N. Sainath, Andrew W. Senior, Françoise Beaufays, and Michiel Bacchiani (2015). “Large vocabulary automatic speech recognition for children”. In: *INTERSPEECH.* 9.

Lin, Min, Qiang Chen, and Shuicheng Yan (2013). “Network in network”. In: *arXiv preprint arXiv:1312.4400.*

Mikolov, Tomáš, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur (2010). “Recurrent neural network based language model”. In: *Eleventh annual conference of the international speech communication association.*

Mohri, Mehryar, Fernando Pereira, and Michael Riley (2008). “Speech recognition with weighted finite-state transducers”. In: *Springer Handbook of Speech Processing*. Springer, pp. 559–584.

Mostow, Jack (2012). “Why and How Our Automated Reading Tutor Listens”. In: 4.

Olah, Christopher (2015). *Understanding LSTM Networks*. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`. [Online; accessed 11-May-2020].

Panayotov, V., G. Chen, D. Povey, and S. Khudanpur (2015). “Librispeech: An ASR corpus based on public domain audio books”. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210.

Park, Daniel S., William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le (2019). “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition”. In: *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*. Ed. by Gernot Kubin and Zdravko Kacic. ISCA, pp. 2613–2617.

Peddinti, Vijayaditya, Daniel Povey, and Sanjeev Khudanpur (2015). “A time delay neural network architecture for efficient modeling of long temporal contexts”. In: *Sixteenth Annual Conference of the International Speech Communication Association*. 15.

Potamianos, A. and S. Narayanan (2003). “Robust recognition of children’s speech”. In: *IEEE Transactions on Speech and Audio Processing* 11.5, pp. 603–616.

Povey, Daniel, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur (2018). “Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks.” In: *Interspeech*, pp. 3743–3747.

Povey, Daniel, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. (2011). *The Kaldi speech recognition toolkit.* Tech. rep. IEEE Signal Processing Society.

Povey, Daniel, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur (2016). “Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI.” In: *Interspeech*, pp. 2751–2755.

Pulugundla, Bhargav, Murali Karthick Baskar, Santosh Kesiraju, Ekaterina Egorova, Martin Karafiát, Lukás Burget, and Jan Cernockỳ (2018). “BUT System for Low Resource Indian Language ASR.” In: *Interspeech*, pp. 3182–3186.

Qian, M., I. McLoughlin, W. Quo, and L. Dai (2016). “Mismatched training data enhancement for automatic recognition of children’s speech using DNN-HMM”. In: *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 1–5.

Qian, Yao, Xinhao Wang, Keelan Evanini, and David Suendermann-Oeft (2016). “Improving DNN-Based Automatic Recognition of Non-native Children Speech with

Adult Speech". In: *Workshop on Child Computer Interaction*, pp. 40–44. URL: http://dx.doi.org/10.21437/WOCCI.2016-7.

Sahraeian, R. and D. Van Compernolle (2016). "A study of rank-constrained multilingual DNNS for low-resource ASR". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5420–5424.

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

Senior, A. and I. Lopez-Moreno (2014). "Improving DNN speaker independence with I-vector inputs". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 225–229.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (Aug. 2016). "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725.

Sheng, P., Z. Yang, and Y. Qian (2019). "GANs for Children: A Generative Data Augmentation Strategy for Children Speech Recognition". In: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 129–135.

Shivakumar, Prashanth Gurunath and Panayiotis Georgiou (2020). "Transfer learning from adult to children for speech recognition: Evaluation, analysis and recommendations". In: *Computer Speech & Language* 63, p. 101077.

Shobaki, Khaldoun, John-Paul Hosom, and Ronald Cole (2007). "CSLU: Kids' speech version 1.1". In: *Linguistic Data Consortium*.

Shobaki, Khaldoun, John-Paul Hosom, and Ronald A Cole (2000). "The OGI kids' speech corpus and recognizers". In: *Sixth International Conference on Spoken Language Processing*.

Snyder, David, Guoguo Chen, and Daniel Povey (2015). "Musan: A music, speech, and noise corpus". In: *arXiv preprint arXiv:1510.08484*.

Stuttle, Matthew Nicholas (2003). "A Gaussian mixture model spectral representation for speech recognition". PhD thesis. University of Cambridge.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*, pp. 3104–3112.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tuerk, Christine and Tony Robinson (1993). “A new frequency shift function for reducing inter-speaker variance”. In: *Third European Conference on Speech Communication and Technology*.

Vajpai, Jayashri and Avnish Bora (2016). “Industrial Applications of Automatic Speech Recognition”. In: *International Journal of Engineering Research and Applications* 6.3, pp. 88–95.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). “Attention is all you need”. In: *Advances in neural information processing systems*, pp. 5998–6008.

Vu, N. T., D. Imseng, D. Povey, P. Motlicek, T. Schultz, and H. Bourlard (2014). “Multilingual deep neural network based acoustic modeling for rapid language adaptation”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7639–7643.

Waibel, A., T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang (1989). “Phoneme recognition using time-delay neural networks”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.3, pp. 328–339.

Wang, Y., T. Chen, H. Xu, S. Ding, H. Lv, Y. Shao, N. Peng, L. Xie, S. Watanabe, and S. Khudanpur (2019). “Espresso: A Fast End-to-End Neural Speech Recognition Toolkit”. In: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 136–143.

Wang, Yiming, David Snyder, Hainan Xu, Vimal Manohar, Phani Sankar Nidadavolu, Daniel Povey, and Sanjeev Khudanpur (2019). “The JHU ASR System for VOiCES from a Distance Challenge 2019”. In: *Proc. Interspeech 2019*, pp. 2488–2492.

Wu, Fei, L. Paola García-Perera, Daniel Povey, and Sanjeev Khudanpur (2019). “Advances in Automatic Speech Recognition for Child Speech Using Factored Time Delay Neural Network”. In: *INTERSPEECH*.

Xu, Hainan, Tongfei Chen, Dongji Gao, Yiming Wang, Ke Li, Nagendra Goel, Yishay Carmiel, Daniel Povey, and Sanjeev Khudanpur (2018). “A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5929–5933.

Yu, Dong and Li Deng (2016). *AUTOMATIC SPEECH RECOGNITION: A DEEP LEARNING APPROACH*. Springer.

Zechner, Klaus, Keelan Evanini, and Cara Laitusis (2012). “Using automatic speech recognition to assess the reading proficiency of a diverse sample of middle school students”. In: *Third Workshop on Child, Computer and Interaction*.

Zhou, Shiyu, Shuang Xu, and Bo Xu (2018). “Multilingual End-to-End Speech Recognition wit A Single Transformer on Low-Resource Languages”. In: *CoRR* abs/1806.05059.

Vita

Fei Wu received her B.E. in Electrical and Electronics Engineering from Stevens Institute of Technology, Hoboken, NJ in 2017, and her B.S. in Electronics and Information Engineering from Beijing Institute of Technology, Beijing, China in 2018. She joined the Master of Science in Engineering, Computer Science program at Johns Hopkins University in the fall of 2018, where she joined Professor Daniel Povey and Professor Sanjeev Khudanpur's lab in Center of Language an Speech Processing. Her research focuses on child speech recognition, and mispronunciation detection for child speech.