Computer Analysis of Children's Non-Native English Speech for Language Learning and Assessment

By

Mengjie Qian

A thesis submitted to
the University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

Department of Electronic, Electrical and Systems Engineering
School of Engineering
College of Engineering and Physical Sciences
University of Birmingham
January 2021

Abstract

Children's ASR appears to be more challenging than adults' and it's even more difficult when it comes to non-native children's speech. This research investigates different techniques to compensate for the effects of non-native and children on the performance of ASR systems. The study mainly utilises hybrid DNN-HMM systems with conventional DNNs, LSTMs and more advanced TDNN models. This work uses the CALL-ST corpus and TLT-school corpus to study children's non-native English speech.

Initially, data augmentation was explored on the CALL-ST corpus to address the lack of data problem using the AMI corpus and PF-STAR German corpus. Feature selection, acoustic model adaptation and selection were also investigated on CALL-ST. More aspects of the ASR system, including pronunciation modelling, acoustic modelling, language modelling and system fusion, were explored on the TLT-school corpus as this corpus has a bigger amount of data. Then, the relationships between the CALL-ST and TLT-school corpora were studied and utilised to improve ASR performance.

The other part of the present work is text processing for non-native children's English speech. We focused on providing accept/reject feedback to learners based on the text generated by the ASR system from learners' spoken responses. A rule-based and a machine learning-based system were proposed for making the judgement, several aspects of the systems were evaluated. The influence of the ASR system on the text processing system was explored.

To my parents

Acknowledges

First and foremost, I would like to express my deepest gratitude to my supervisors, Professor Martin Russell and Dr Peter Jančovič, for their professional advice, continuous support and kind encouragement. Their immense knowledge and extraordinary experience have inspired me in all the time of my academic research and daily life. With their guidance, I become not only a better researcher but also a better person.

I would also like to thank my brilliant colleagues of the Speech Group at the University of Birmingham, Dr Linxue Bai, Guy Cooper, Dr Evangelia Fringi, Dr Roozbeh Nabiei, Yikai Peng, Chloe Seivwright, Dr Philip Weber and Xizi Wei for their company and encouragement.

I wish to acknowledge Thiago Fraga da Silva, Andrew Breen and Antonio Bonafonte for their guidance during my two internships at Amazon, also big thanks to my friends and colleagues from the fourth floor of the Amazon Cambridge office for a wonderful time we spent together. I also wish to extend my gratitude to Professor Ailbhe Ní Chasaide, Dr Neasa Ní Chiarain and Harald Berthelsen from Trinity College Dublin for their encouragement and inspiration during my time working with them as a part-time Research Assistant.

I am also grateful to my great friends, Chen Chen, Yuanchang Chen and Miaomiao Ma for overcoming the timezone differences and sharing all the happiness and toughness with me. A special thank goes to Dr Linxue Bai and her husband Dr Yongjing Wang, whom I regard as my family to me, for their generous help ever since I came to the UK.

Finally, I would like to thank my parents, my brother and my sister-in-law for their love, understanding and support. My biggest thanks to my parents Mr Renbao Qian and Mrs Zhengping Zhang who always encourage me to pursue my dreams, support all my decisions and believe in my abilities to achieve my goals.

Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
BERT	Bidirectional Encoder Representations from Transformers
BP	Back-Propagation
BPC	Broad Phone Class
CA	Correct Accept
CALL	Computer-Assisted Language Learning
CBOW	Continuous Bag-of-Word
CD	Contrastive Divergence
CEFR	Common European Framework of Reference for Languages
CMLLR	Constrained Maximum Likelihood Linear Regression
CMN	Cepstral Mean Normalisation
CNN	Convolutional Neural Network
CR	Correct Reject
DBN	Deep Belief Network
DCT	Discrete Cosine Transform
DNN	Deep Neural Network
ELMo	Embeddings from Language Models
EM	Expectation-Maximization
FA	False Accept

FBANK	Mel-scale Filter Banks
fMLLR	feature-space Maximum Likelihood Linear Regression
FR	False Reject
FST	Finite-State Transducer
GFA	Gross False Accept
GloVe	Global Vectors for Word Representation
GMM	Gaussian Mixture Model
GOP	Goodness of Pronunciation
HMM	Hidden Markov Model
ICALL	Intelligent Computer Assisted Language Learning
IHM	Independent Headset Microphone
L1	First Language/Native Language
L2	Second Language
LDA	Linear Discriminant Analysis
LPC	Linear Prediction Cepstral
LPCC	Linear Prediction Cepstral Coefficient
LSTM	Long Short-Term Memory
LVCSR	Large Vocabulary Continuous Speech Recognition
MBR	Minimum Bayes Risk
MDM	Multiple Distant Microphone
MFCC	Mel-Frequency Cepstral Coefficients
MLLR	Maximum Likelihood Linear Regression
MLLT	Maximum Likelihood Linear Transform
MLP	Multi-layer Perceptron
MMI	Maximum Mutual Information

MPE	Minimum Phone Error
NLP	Natural Language Processing
PCA	Principal Component Analysis
PER	Phone Error Rate
PFA	Plain False Accept
PLP	Perceptual Linear Prediction
PV-DBOW	Distributed Bag-of-Words version of Paragraph Vector
PV-DM	Distributed Memory Model of Paragraph Vector
RBM	Restricted Boltzmann Machine
RNN	Recurrent Neural Network
RNNLM	Recurrent Neural Network Language Model
SA	Scoring Accuracy
SAT	Speaker Adaptive Training
SBE	Standard British English
SDM	Single Distant Microphone
SGD	Stochastic Gradient Descent
SLT	Spoken Language Technology
sMBR	state-level Minimum Bayes Risk
SNR	Signal-to-Noise Ratio
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TDNN	Time-Delay Neural Network
TF-IDF	Term Frequency - Inverse Document Frequency
VTLN	Vocal Tract Length Normalization
WER	Word Error Rate

Chapter 1

Introduction

1.1 Research Background

Learning a second language is becoming increasingly important in today's global world. Spoken Computer-Assisted Language Learning (CALL) is a promising approach to learning a foreign language as it provides learners with a flexible and practical environment to practise spoken language. In general, spoken CALL systems utilises automatic speech recognition (ASR) systems to process learner's spoken responses and text processing systems to assess the texts produced by the ASR system for error detection and feedback generation (Johnson and Valente, 2009; K. Lee et al., 2014). Hence, the ASR system and language assessment system are two essential components of a CALL application.

The availability of large amounts of training data and large computational resources promote the improvement of ASR systems and made them usable in many application domains. Recent researches have demonstrated that ASR systems can achieve performance level on par with human transcribers for some tasks (Protalinski, 2017). However, ASR systems still present deficiencies when applied to educational applications, e.g. spoken Computer-

Assisted Language Learning (CALL) systems. As it is for language learning, the target speakers of a CALL system are non-native speakers, or non-native children in most cases. In such cases, both factors – non-native and children – make ASR more challenging. Cheng et al. (2015) reported a WER of 33.0% for open-ended spoken responses produced by K-12 (kindergarten to grade 12) English learners. Y. Qian et al. (2017) achieved a WER of 23.2% for responses from adolescent English learners representing a range of first language backgrounds. Knill et al. (2020) reported a WER of 15.67% for Italian children taking English language tests, which is the best performing system among all the submissions for the Shared Task on Non-Native Children's Speech Recognition (Gretter et al., 2020b). Although the progress over the years is remarkable, the word error rate is still quite high compared to what has been achieved on adults' speech – 4.9% WER as reported in Protalinski (2017).

The difficulties for non-native children's speech recognition lie in many aspects. Firstly, there are smaller amounts of data available for non-native speakers, and the data is even more limited when it comes to children. Secondly, non-native speech often incorporates several phenomena that can greatly reduce ASR performance, such as mispronunciations, code-switching, incorrect grammar, false starts, partial words, etc. (Gretter et al., 2020b). Moreover, children's speech is linguistically different from adults' speech at many levels, e.g. acoustic, prosodic, lexical, morphosyntactic and pragmatic. Specifically, the differences are caused by three factors: (a) physiological differences, e.g. children have shorter vocal tracts, (b) cognitive differences, e.g. children are at different stages of language acquisition, this also brings more variabilities in children's speech, (c) behavioural differences, such as whispered speech, fiddling with the microphone, wobbling their body, moving their heads (Gretter et al., 2020b; M. Qian et al., 2016). The difficulties and techniques of ASR for children's speech and non-native speech are briefly reviewed in Section 2.3.3 and Section 2.3.4, respectively.

Automatic language assessment systems assess learner's language in terms of fluency, pronunciation, use of grammar and vocabulary for either their spoken speech or written text. One of the aims is to provide feedback to learners to help them improve their language

skills. The feedback can be delivered in multiple stages, for instance, accept/reject, highlight errors, propose more accurate answers (Magooda and Litman, 2017). Among these, providing accept/reject feedback is the first stage and has to be the most accurate. Hence, this thesis targets at assessing the acceptance of spoken responses in a speech-enabled CALL application for prompt-response pairs.

The focus of this thesis is on analysing children's non-native English speech for language learning and assessment. One of the main goals of this research is to improve the ASR performance for non-native children's English speech. Specifically, experiments are performed on German-speaking and Italian-speaking children's speech corpora. The investigation has been conducted at several aspects including data augmentation, feature selection, model adaptation, model selection, language model rescoring, system fusion, etc. Another goal of this work is to design proper systems for assessing the acceptance of children's spoken responses given prompt-response pairs in a language learning scenario. Rule-based system and machine learning-based systems have been explored using data collected from a CALL-SLT system where German-Swiss teenagers are practising spoken English. To gain a better understanding of the difficulties of children's speech and the differences between children's corpora, acoustic space visualisation has been proposed which represents different children's datasets in a 2-dimensional feature space. Comparisons at the ASR performance level and text level are also performed on two non-native children's English speech corpora.

1.2 Research Questions and Contributions

The research described in this thesis provides some insights from multiple aspects that are of critical importance in language learning and assessment. The major research questions are presented below.

1.2.1 Research Question 1

In the scenario of limited in-domain resource, what techniques are beneficial to improve ASR performance for children's non-native speech recognition?

The research has been conducted on the CALL-ST corpus which contains around 13.6 hours of English recordings from 12-15 years old German-Swiss children interacting with a CALL-SLT application. The overall amount of recordings is small, considering the data was distributed over three years for three spoken CALL Shared Tasks, the in-domain training data is very limited for each shared task.

The first aspect explored in this work is what out-of-domain corpora are appropriate for adding to the training data, what percentage of these corpora should be added and what particular combination of the corpora gives the best performance for the task. The second aspect is about feature selection and adaptation. The objective is to analyses different features' capability of representing children's speech and their suitability for acoustic modelling in speech recognition. Another question explored in this study is that whether a more advanced neural network, e.g. Long Short-Term Memory (LSTM) which has been demonstrated useful in large vocabulary continuous speech recognition (LVCSR) tasks, is also useful with limited in-domain data. Other techniques, such as acoustic model adaptation and sequence-discriminative training, have also been investigated. The related work will be presented in Chapter 4.

1.2.2 Research Question 2

What techniques can be applied to develop text processing systems for assessing the correctness of children's responses in a spoken CALL application?

This work has been performed on the CALL-ST corpus where the data are prompt-response

pairs, the spoken responses have been converted to text using an ASR system which has been explored in Chapter 4. To provide the accept/reject feedback to the students, a rule-based system based on a reference grammar and a machine learning-based system utilising a classifier and the sentence similarity features are developed. Several aspects of the machine learning-based system have been explored, e.g. what embeddings are suitable for representing the prompts and responses, which similarity algorithm is better for evaluating the distance between the prompt and the responses, what classifier is the best for the task. Particularly, is the word order important in word embedding based sentence similarity calculation for short sentences, i.e. the CALL-ST data? Since the text assessment is based on the hypothesis from a speech recogniser, what is the relationship between the ASR performance and the overall system performance? Does a hypothesis with a lower WER always result in a better assessment score? These studies are presented in Chapter 5.

1.2.3 Research Question 3

In the scenario of adequate in-domain data and an advanced neural network, what techniques are beneficial to improve ASR performance for children's non-native speech?

The studies have been done on the TLT-school corpus which consists of 49 hours English recordings from 9-16 years old Italian children taking English language tests. As there are more in-domain data, a more advanced acoustic model – Time-Delay Neural Network (TDNN) is employed as the baseline model for acoustic modelling. This work focuses on the closed-track speech recognition task, which only uses in-domain data for system development. The investigation has covered many aspects of speech recognition, trying to improve the ASR performance in general and address the problems caused by the factors of non-native and children of the speech. In particular, is pronunciation modelling for non-native, non-English or non-speech voices useful for improving ASR performance? Does transfer learning between

subsets of the training data help to achieve a better overall performance? To what extent can the language model be improved with a limited resource? How to perform system fusion for ASR systems and is it beneficial? These questions have been explored in Chapter 6.

1.2.4 Research Question 4

What are the relationships between different children's speech corpora? How can we utilise these relationships to obtain better ASR systems for children's non-native speech?

Previously, the CALL-ST corpus and the TLT-school corpus have been utilised to explore different aspects of the ASR system. The children in these two corpora have different native language and age range. The recording conditions and transcription rules are also different. Are these two corpora complementary to each other? What aspects are in common and different between these two corpora? Can we use them to obtain a better ASR system for both of the corpora? The investigation has been discussed in Chapter 7.

1.3 Thesis Outline

The outline of this thesis is listed in this section, with brief introductions to each chapter. Chapter 2 and 3 are background chapters, Chapter 4 to 7 describe the experiments and findings, and Chapter 8 concludes the thesis.

Background chapters

- Chapter 2 reviews the theories and algorithms that are relevant to this thesis. In particular, Computer-Assisted Language Learning (CALL) is reviewed in Section 2.2. Then, Automatic Speech Recognition(ASR) is introduced and reviewed in Section 2.3.

including two special cases, namely ASR for children and ASR for non-native speakers. Section 2.4 and Section 2.5 provide full descriptions of the GMM-HMM based and hybrid DNN-HMM based acoustic modelling techniques in ASR systems. Various acoustic training criteria are described in Section 2.6. Two advanced neural networks for acoustic modelling – Long-short Term Memory (LSTM) and Time-Delay Neural Network (TDNN) – are explained in Section 2.7 and Section 2.8, respectively. Section 2.9 introduces topological manifold and its use in speech recognition. Section 2.10 reviews the techniques for extracting word embeddings and document embeddings that are employed in text processing systems covered in Chapter 5.

- Chapter 3 provides the full description of the speech corpora used in our speech recognition and text processing systems, namely the CALL-ST corpus, the TLT-school corpus, the PF-STAR corpus, the AMI corpus and the WSJCAM0 corpus. The experiments and analysis are mainly focused on the CALL-ST and TLT-school corpora, other corpora are also involved in some experiments.

Experimental chapters

- Chapter 4 addresses research question 1. This chapter explores how to improve ASR performance using limited in-domain data. Experiments and results with data augmentation, feature selection and model adaptation in a DNN-HMM system are reported. This chapter also compares the conventional DNN-HMM model with an LSTM model and compares cross-entropy training with sequence discriminative training.
- Chapter 5 addresses research question 2. This chapter provides the details of the systems developed for making judgements on children's spoken responses, and explores the influence of the ASR system on the text processing system.
- Chapter 6 addresses research question 3. This chapter presents the experiments on the TLT-school corpus, which is fairly adequate for a speech recognition task compared to the CALL-ST corpus. More aspects of the ASR system have been explored for recognising children's non-native English, including pronunciation modelling, language

modelling and system fusion.

- Chapter 7 addresses research question 4. This chapter explores the relationship between two non-native children's English speech corpora. Experiments to improve the ASR performance using two corpora are reported, visualisations and interpretations of i-vectors using t-SNE are presented, text-level analyses are performed with statistics of the transcriptions, perplexities of the language models and distribution of the vocabularies.
- Chapter 8 summarizes the major contributions in this thesis and suggests potential future extensions of current work.

1.4 Publications

Some of the ideas and results in this thesis have been published in reviewed conference papers at various stages during the period of study for this thesis. It is worth mentioning that the author has also worked on other aspects of speech recognition during the PhD which are not detailed in this thesis. The research focus of that work is slightly mismatched with the theme of this thesis, however it benefited a lot the work to be presented in this thesis and has led to a publication (M. Qian et al., 2018a). A full list of publications during author's PhD research is listed below:

1. **M. Qian**, P. Jančovič, and M. Russell, "The University of Birmingham 2019 Spoken CALL Shared Task Systems: Exploring the importance of word order in text processing", in Proc. 8th ISCA Workshop on Speech and Language Technology in Education (SLaTE), Graz, Austria, 2019, pp. 11-15.
2. C. Baur, A. Caines, C. Chua, J. Gerlach, **M. Qian**, M. Rayner, M. Russell, H. Strik, and X. Wei, "Overview of the 2019 Spoken CALL Shared Task", in Proc. 8th ISCA Workshop on Speech and Language Technology in Education (SLaTE), Graz, Austria,

2019, pp. 1-5.

3. **M. Qian**, X. Wei, P. Jančovič, and M. Russell, "The University of Birmingham 2018 Spoken Call Shared Task Systems", in Proc. Interspeech, Hyderabad, India, 2018, pp. 2374-2378.
4. **M. Qian**, L. Bai, P. Jančovič, and M. Russell, "Phone Recognition using a Non-Linear Manifold with Broad Phone Class Dependent DNNs", in Proc. Interspeech, Hyderabad, India, 2018, pp. 3753-3757.
5. C. Baur, A. Caines, C. Chua, J. Gerlach, **M. Qian**, M. Rayner, M. Russell, H. Strik, and X. Wei, "Overview of the 2018 Spoken CALL Shared Task", in Proc. Interspeech, Hyderabad, India, 2018, pp. 2354-2358.
6. D. Jülg, M. Kunstek, C. P. Freimoser, K. Berkling, **M. Qian**, "The CSU-K Rule-Based System for the 2nd Edition Spoken CALL Shared Task", in Proc. Interspeech, Hyderabad, India, 2018, pp. 2359-2363.
7. **M. Qian**, X. Wei, P. Jančovič, and M. Russell, "The University of Birmingham 2017 SLaTE CALL shared task systems", in Proc. 7th ISCA Workshop on Speech and Language Technology in Education (SLaTE), Stockholm, Sweden, 2017, pp. 91-96.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a background review of the techniques that are related to or have been employed in our work which will be presented in the later chapters of this thesis. It begins by reviewing the motivation, history and related work in Computer-Assisted Language Learning (CALL) in Section 2.2. In this thesis, the work was initially focused on the assessment issues in CALL systems. Specifically, we focused on the speech recognition and text processing components in order to judge the syntactic and semantic appropriateness of learners' spoken responses. Apart from this, a great deal of effort has been put into improving the speech recognition system to generally improve the CALL system for children. Section 2.3 covers the review for automatic speech recognition (ASR) and two special aspects in ASR – ASR for children's speech and ASR for non-native speech. Section 2.4 and Section 2.5 review two dominant approaches for ASR: the Gaussian Mixture Model - Hidden Markov Model (GMM-HMM) and Deep Neural Network - HMM (DNN-HMM) based systems. Section 2.6 to Section 2.8 cover the most popular training criteria for acoustic modelling and two neural network structures that are well used in recent years. Section 2.9 introduces the topological

manifold, another special aspect in speech recognition. Section 2.10 presents a review of the word embeddings, Word2Vec and Doc2Vec models in particular, that are used in text processing.

2.2 Computer-Assisted Language Learning

2.2.1 Motivation

With the rapid development and globalization of the world, learning a foreign language has become a high demand for people across the world. English is by far the most common studied second language (L2) in the world with 1.5 billion learners, followed by French with 83 million learners and Chinese with 30 million learners (Noack and Gamio, 2015). Practising speaking is necessary to improve the spoken proficiency for an L2 learner. However, only a small number of students have the chance to practice with a native speaker or a one-to-one instructor, most of the L2 learners are learning by classroom courses and self-study which are important for reading and writing skill but less efficient in improving spoken skills. The availability of Computer-Assisted Language Learning (CALL) systems could provide students with a more suitable, flexible and practical learning environment. Apart from written language, the development of spoken language technology makes it possible to apply CALL systems to spoken language practice.

2.2.2 History of CALL

Several scholars in the field reviewed the history of CALL from different perspectives. Gamper and Knapp (2002) review the CALL systems that address communicative skills or pronunciation using Automatic Speech Recognition (ASR) technologies. Heift (2017) provides an overview of Intelligent Computer-Assisted Language Learning (ICALL) in teach-

ing and learning in L2 written language. Tafazoli et al. (2019) focus on the advancement of technologies in the field of CALL from a chronological perspective, presenting the history of CALL from the 1950s to the 21st century. The history of CALL is often divided into three phases: structural/behaviourist CALL, communicative CALL and integrative CALL (Warschauer, 2000). Starting in the 1950s and developing through the 1970s was the Structural/Behaviourist CALL – the era of stimulus and response. The computer prompts the student with a question (stimulus) and the student gives an answer (response) by filling in the blanks or choosing from a given set of choices. Communicative CALL came in the 1980s and 1990s, communication and interaction were important in this phase. Instead of learning the language – its rules, syntax, phonemes and morphemes, students got the opportunities to actually use the language. The next phase is the Integrative CALL from 2000 onwards, which integrated general language knowledge in the first phase and the communication skills in the second phases.

2.2.3 Related Work in CALL

Over the last twenty years, the work on CALL is mainly focused on interactive tuition and assessment. From the perspective of interactive tuition, while more researches come into CALL via the disciplines of computational linguistics rather than language teaching, more techniques are incorporated into recent CALL systems, e.g. Natural Language Processing (NLP), speech recognition and speech synthesis. A system aiming at making learners pay more attention to the grammar in spoken modality was developed by Vries et al. (2013), it uses an ASR system to process L2 learner's responses. Chatbots, defined as computer programs which attempt to simulate conversations of human beings via text or voice interactions (Rouse, 2016), have been shown to have a positive impact on learning success and student satisfaction (Winkler and Soellner, 2018). J.-X. Huang et al. (2017) developed a dialogue-based chatbot for a CALL system to deal with out of topic user utterances to

make the conversation more natural between the system and the learners. Different NLP and machine learning algorithms were used and compared by Dutta (2017) for developing an intelligent chatbot to assist high school students in language learning. They found that rule-based chatbots might be more than sufficient for closed domains with a fixed set of possible questions. In all other cases, a better learning and teaching outcome can be achieved by chatbots with artificial intelligence techniques. Many CALL applications are limited to the pre-defined curricula that the application is used for teaching, while Proença et al. (2019) developed an application for learners to practice pronunciation on freely input text.

The assessment in a CALL system can be conducted at different levels, e.g. pronunciation, grammar and semantics. Witt and S. J. Young (2000) proposed a likelihood based "goodness of pronunciation" (GOP) measure to score the phone-level pronunciation for use in CALL systems. Their work indicates that a computer based pronunciation scoring system is capable of providing similar feedback to students as a human expert when a sufficient amount of speech is available. This GOP measure is evaluated in different ways by Kanters et al. (2009) to determine whether and how pronunciation error detection can be improved. They validated the use of artificially introduced pronunciation errors, which is a useful finding as the paucity of non-native material is a common problem. Minematsu (2007) argued that the approaches to pronunciation assessment such as GOP identifies learning to pronounce as learning to impersonate, which might not be pedagogically enough. An automatic system was presented by Y. Wang et al. (2018) to address the pronunciation assessment of spontaneous spoken English using a Gaussian Process grader, which can also provide the uncertainty of its prediction.

A series of spoken CALL shared tasks was released from 2016 to 2019 (Baur et al., 2017; Baur et al., 2018; Baur et al., 2019) to assess grammar and meaning correctness of learners' spoken responses. This series of tasks have a speech processing and a text processing task, participants can choose to work on either or both of the tasks. They have attracted many research groups and received a lot of positive responses. The importance of the speech

recognition system to the overall system has been verified by multiple groups (M. Qian et al., 2017; Oh et al., 2017; Axtmann et al., 2017; M. Qian et al., 2018b; Nguyen et al., 2018; Jülg et al., 2018; Gretter et al., 2019). Rule-based systems (Ateeq et al., 2018; Jülg et al., 2018), traditional classifiers with various features (Evanini et al., 2017; Caines, 2017; Magooda and Litman, 2017; Evanini et al., 2018) and DNN-based classifiers (Oh et al., 2017; Gretter et al., 2019; Sokhatskyi et al., 2019; M. Qian et al., 2019) are used for the text processing component of the tasks.

2.3 Automatic Speech Recognition

2.3.1 Introduction

Automatic speech recognition (ASR) is a task of converting speech data into written transcription. Modern ASR system consists of two main stages: feature extraction and decoding. A pronunciation dictionary, acoustic model and language model are required for the decoder. A diagram of a speech recognition system is shown in Figure 2.1.

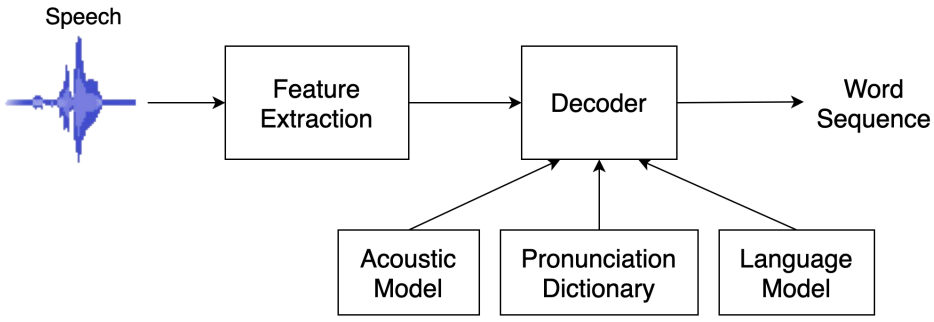


Figure 2.1: *A speech recognition system.*

The aim of feature extraction is to convert the input audio signal into a sequence of fixed size acoustic vectors X. The most popular features are Mel Frequency Cepstral Coefficients (MFCCs) (Mermelstein, 1976; S. Davis and Mermelstein, 1980), Mel-scale filter banks (FBANKs) and Perceptual Linear Predictions (PLPs) (Hermansky, 1990). The decoder aims

at finding the most probable word sequence $\hat{W} = w_1, w_2, ..., w_l$ corresponding to X, i.e.

$$\hat{W} = \underset{W}{\operatorname{argmax}} P(W|X). \tag{2.1}$$

Since it is difficult to model $P(W|X)$, the Bayes rule is applied to transform it into an equivalent one as follow:

$$P(W|X) = \frac{p(X|W)P(W)}{p(X)}, \tag{2.2}$$

where P is used for probabilities and p is used for probability densities; $p(X)$, the probability of the observation vectors, is a constant; $p(X|W)$ is determined by acoustic models and $P(W)$ is achieved by language models. Hence, Equation 2.1 can be reformed as below, which indicates that a good ASR system requires a good acoustic model and a good language model:

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

Hidden Markov model (HMM) has played a dominating role in speech recognition acoustic modelling for decades, it has been utilised in two ways: Gaussian Mixture Model based HMMs (GMM-HMMs) and Deep Neural Network based HMMs (DNN-HMMs), the details of which will be presented in Section 2.4 and Section 2.5.

The language model is to compute the prior probability $P(W) = P(w_1^l)$, which can be decomposed using the chain rule of probability:

$$\begin{aligned} P(W) &= P(w_1, w_2, ..., w_l) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)...P(w_l|w_1^{l-1}) \\ &= \prod_{k=1}^{l} P(w_k|w_1^{k-1}). \end{aligned} \tag{2.4}$$

This is often modelled as an n-gram model, where the probability of a word only depends on the previous $n-1$ words, i.e.

$$P(W) = \prod_{k=1}^{l} P(w_k|w_{k-(n-1)}^{k-1}). \tag{2.5}$$

In recent years, language models trained using neural networks start to emerge, e.g. Recurrent Neural Network Language Model (RNNLM).

The pronunciation dictionary comprises one or more phone-level transcriptions of the words that occur in the training set. It is needed in an ASR system as it provides the link between sub-word HMMs and the language model.

2.3.2 Reviews on ASR

The state-of-the-art ASR has achieved word accuracy rate on par with humans, Google claims the lowest word error rate at 4.9% in 2017 which broke the 95% threshold of human accuracy (Protalinski, 2017). Looking back at the history of ASR, although a lot of major breakthroughs have been achieved in the last 10 years due to the development of deep neural networks, the study of ASR started in the 1950s.

The first speech recognition system dates back to 1952 when Bell Laboratories designed the "Audrey" system which could recognise telephone quality digits spoken by a single individual (K. H. Davis et al., 1952). In the early 1960s, IBM developed "Shoebox" which could understand 16 spoken words in English including the ten digits and respond to simple arithmetic problems like "plus", "minus" and "total" (Dersch, 1962). Since the 1970s, more researchers became interested in ASR but the development was slow, the ASR systems were only able to handle isolated words with a small vocabulary.

A statistic method was proposed in the 1980s known as the "Hidden Markov Model" (HMM). This brought a breakthrough in ASR, making the systems capable of recognising several thousands of words. In 1987, the first system that can recognise speaker-independent continuous speech was developed – the "SPHINX" system, by Kai-Fu Lee from the Carnegie Mellon University (CMU). It was developed based on Gaussian Mixture Model (GMM) - HMM. In the early 1990s, Artificial Neural Network (ANN), a predecessor of DNN was

applied in speech recognition, but it did not outperform GMM-HMM based systems. GMM-HMM was the leading approach for speech recognition for more than 20 years, the details of a GMM-HMM based ASR system will be presented in Section 2.4.

In 2006, Hinton proposed the Deep Belief Network (DBN), in which Restricted Boltzmann Machine (RBM) was used to initialise the network (Hinton et al., 2006). In 2009, Hinton and his student Mohamed applied DBN to speech recognition, their best performing DBN model achieved a phone error rate (PER) of 23% on the TIMIT core test set which outperformed all other methods at that time (Mohamed et al., 2009). Since then, DNN has been applied to large vocabulary continuous speech recognition (LVCSR) and became the most popular method for speech recognition (Yu and Deng, 2010). The popularity of DNN in ASR not only relies on a more powerful network – more layers (deeper) and finer targets (wider), but also relies on the increase of computation power (GPU) and the availability of big data. DNN can be used for extracting front-end features, building acoustic models and language models. The conventional DNN-HMM based ASR system involves using the GMM-HMM model to provide training labels, it will be introduced in Section 2.5.

The success of applying DNN to ASR revived many approaches. Discriminative training was proposed in the 1990s, it outperforms cross-entropy training under deep neural network architectures for speech recognition. The aim of cross-entropy training is to minimize frame-level error rate, while discriminative training is aiming at minimizing sequence level error rate. These different training criteria will be explained in Section 2.6.

Deep neural networks considerably boosted ASR performance, the hybrid DNN-HMM approach together with recurrent long short-term memory (LSTM) neural network and time-delay neural networks (TDNN) marks the state-of-the-art on many tasks, covering a wide range of training set sizes. LSTM and TDNN learn sequence-level representation, they are able to capture long range and local dependencies between different time steps. An introduction to the LSTM recurrent neural network and TDNN model will be presented in

Section 2.7 and Section 2.8, respectively.

Recently, more and more alternative approaches emerge, moving gradually towards so-called end-to-end approaches. A conventional ASR requires an acoustic model, a language model and a pronunciation dictionary. Each component plays a different role and uses different technologies, it is hard to optimize the system globally. End-to-end systems directly map a sequence of input acoustic features into a sequence of graphemes or words, aiming to get rid of those components in a conventional ASR system. End-to-end approaches have shown promising results, especially using large training sets.

Another interesting topic in speech recognition is topological manifold. A number of researches have been trying to understand the behaviour of deep neural networks. It is challenging in general, but it is easier to explore low-dimensional DNNs, in which the behaviour and training can be understood with visualisations. It has been observed that different types of speech sounds lend themselves to different acoustic analysis in low-dimensional DNNs (Bai, 2018). This perspective allows us to gain deeper intuition about the behaviour of neural networks and connect neural networks to a mathematical structure called topological manifold (Olah, 2014). More descriptions on manifold are covered later in Section 2.9.

2.3.3 ASR for Children's Speech

A special case in ASR is that for children's speech, because it appears to be more challenging than for adults' speech. Speech recognition for adults has achieved performance on par with human over the last few years (Protalinski, 2017), however speech recognition for children is still significantly less accurate than that of adults (M. Russell and D'Arcy, 2007; Gerosa et al., 2009; Liao et al., 2015).

In 1996, Wilpon and Jacobsen investigated the performance of a speech recognizer across a great span of ages including children, adolescents, adults and the elderly using

a connected digit recognizer. They found that speech recognition would be a matter of having enough sufficiently representative training data as long as the ASR addresses speakers between 15 to 70 years old. The error rates increased dramatically outside this age range, even with balanced amounts of training data. Their recognizer was 170% higher for children compared to that for adults when trained with a combined dataset of all age groups and 122% higher when trained on children data only (Wilpon and Jacobsen, 1996).

This is mostly due to children's speech has more variabilities than adults'. S. Lee et al. (1999) studied the changes in magnitude and variability of duration, spectral envelope, fundamental and formant frequencies of children's speech as a function of age and gender. They found that the within-subject variability decreased with the increase in age from 5 years to 12 years, reaching adult level at an age of 15.

Several techniques were proposed to tackle the acoustic variability. Different front-end features with various setups were studied. Wilpon and Jacobsen (1996) computed the linear prediction cepstral coefficients (LPCC) with a lower LPC order for children's speech and found it work better than normal order LPCCs. Q. Li and M. J. Russell (2001) found that changing the length of the analysis window and the width of filters in the mel filter-bank have limited effect on children's speech recognition performance. However, Shivakumar et al. (2014) found it beneficial to increase the frame width and decrease MFCC coefficients as they provide some smoothing to help decrease the variability in speech. They also compared MFCCs, Perceptual Linear Prediction (PLP) cepstral coefficients and spectrum based filter bank features, MFCC outperforms the other two features in their experiments.

Vocal Tract Length Normalization (VTLN) technique was used as a standard technique in children's ASR systems to suppress acoustic variability introduced by the developing of vocal tracts in children (L. Lee and Rose, 1996; Eide and Gish, 1996; Wegmann et al., 1996). The use of VTLN in opposing directions when training a DNN-HMM model with mismatched data was explored by M. Qian et al. (2016). A warping-factor aware DNN system

was proposed by Serizel and Giuliani with posterior probabilities of VTLN warping factors appended to acoustic features as the input to a DNN (Serizel and Giuliani, 2014; Serizel and Giuliani, 2017). Deep neural networks, with various new architectures, are becoming more capable of learning internal representations that are invariant with respect to sources of variability such as the vocal tract length and shape, e.g. VTLN was not effective in an LSTM acoustic model (Liao et al., 2015).

Adapting acoustic models with Maximum Likelihood Linear Regression (MLLR) (Elenius and Blomberg, 2005; Gerosa et al., 2007) and speaker adaptive training (SAT) based on Constrained MLLR (CMLLR) were found to be effective for children's ASR (Giuliani et al., 2006; Gerosa et al., 2007). Pitch normalization for addressing the pitch mismatch between children's and adults' speech was proposed for children's ASR. It obtained a relative improvement of 9% over the baseline in a mismatched ASR and the improvement was found to be additive to that obtained with speaker normalization, VTLN and CMLLR techniques (Ghai and Sinha, 2015).

There is also variability in children's pronunciation patterns. Due to differing and partial linguistic knowledge, kids tend to mispronounce. Improvements can be gained through the use of customized dictionaries (Q. Li and M. J. Russell, 2002). Pronunciation confusion matrices for children in different age classes are presented by Shivakumar et al. (2014), they also show that data-driven pronunciation variation modelling is useful. However, part of the variations is attributed towards the phonological factors associated with language acquisition and hence the customization of dictionaries have their limitations (Fringi et al., 2015; Fringi et al., 2016; Fringi and M. J. Russell, 2018).

Acoustic modelling using DNN models outperforms that using GMM-HMM models for adults' speech recognition (Mohamed et al., 2009; Yu and Deng, 2010). When it is applied to children's speech recognition, transfer learning is a useful approach to make use of a DNN trained with mismatched data to tackle the lack of data problem. Shivakumar and Georgiou

(2020) investigated different transfer learning adaptation techniques, specifically to address the acoustic variability and pronunciation variability which are two major factors degrading children ASR. Their results suggest that all the variabilities present between children and adults are concentrated at the top (pronunciation level) and bottom (acoustic level) layers of the DNN.

Recently, more children's speech has become available and the network architecture has advanced, it is possible that training a model with only children's speech may give comparable performance. Using only children's data, a TDNN model trained with MFCC features appended with i-vectors obtained a WER of 7.5% on a non-native children speech recognition task (Gretter et al., 2019).

2.3.4 ASR for Non-Native Speech

Recognition of non-native speech is another difficult ASR task. The recognition accuracy is usually drastically lower when using a system trained on native speech to recognise non-native speech than to recognise native speech (Z. Wang et al., 2003; Matassoni et al., 2018). Non-native speech presents higher acoustic and linguistic variability compared to native speech, because it is characterized by pronunciation errors, accented pronunciation, lexical and syntactical errors that mainly depend on the proficiency level of the target language as well as on the cross-language interference between the speaker's mother tongue and the target language (Gerosa and Giuliani, 2004). Witt (2012) has given an in-depth review of automatic error detection in pronunciation training, in which pronunciation errors are divided into phonemic error and prosodic error, each with a few sub-types. She also pointed out the low reliability of pronunciation error detection by human experts on individual phoneme level. Likewise, automated error detection is not that reliable either, consequently the correlation between the two is even less.

Many researches have been done to tackle pronunciation and acoustic variability in non-native speech for automatic speech recognition. A finite-state transducer (FST) -based approach has been previously explored to better model the lexicon patterns in non-native speakers (Livescu and Glass, 2000), this approach reduced the WER from 20.9% to 18.8% on a non-native test set. Oh et al. (2007) analysed the pronunciation variability between native English speech and non-native English speech spoken by Koreans using both a knowledge-based approach and a data-driven approach, then performed acoustic model adaptation based on the pronunciation variability at state-tying step of the ASR system. Imseng et al. (2011) proposed to use Kullback-Leibler divergence based hidden Markov models (KL-HMM) to handle the acoustic variability present in multi-accented non-native speech. Estimating multilingual phoneme posterior probabilities with a multilayer perceptron and using the probabilities as input feature to the KL-HMM, their system has shown good performance in low-resource conditions. A hybrid approach combining acoustic modelling and pronunciation modelling was proposed by Bouselmi et al. (2007) to enhance the ASR performance for non-native speech. Lately, DNN-based transfer learning techniques were analysed and discussed for training effective acoustic models for non-native children's speech starting from children's native speech with limited non-native audio material (Matassoni et al., 2018).

2.4 HMM based speech recognition using GMMs

2.4.1 Introduction

Hidden Markov model based system has been the mainstream system for speech recognition since it was proposed in the 1980s. HMM is a statistical Markov model in which the goal is to learn about the unobservable ("hidden") states by visible observations. It is assumed that the hidden states follow a Markov process and the probability of a particular observation at time t does not depend on previous observations and only depend on the current state.

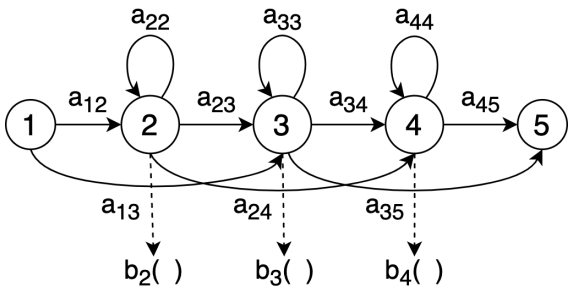


Figure 2.2: *A simple Left-Right 5-state HMM (S. Young et al., 2002).*

A left to right 5-state HMM is illustrated in Figure 2.2. The middle three states are emitting states that have output probability distribution associated with them, the entry and the exit states are non-emitting states. An HMM is specified by the following components (Poritz, 1988; Rabiner, 1989):

$Q = q_1 q_2 ... q_N$	a set of N **states**.
$O = o_1 o_2 ... o_T$	a sequence of T **observations**.
$\pi = \{\pi_i\}$	**initial state probability distribution**, $\sum_{i=1}^{N} \pi_i = 1$, π_i is the probability that the Markov chain starts in state i, some states j may have $\pi_j = 0$.
$A = \{a_{ij}\}$	a **transition probability** matrix, where a_{ij} represents the probability of a transition from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1, \forall i$.
$B = \{b_i(o_t)\}$	**emission probabilities**, also called **output probabilities** or **observation likelihoods**, where $b_i(o_t)$ is the observation PDF given HMM state i for a D-dimensional observation vector o_t.

Each emission probability $b_i(o_t)$ can be described with a multivariate Gaussian mixture model. An HMM λ is composed of the weights, mean vectors and covariance matrices of all Gaussian mixture components for each state. HMMs are characterized by three fundamental problems:

- **Problem 1 (Likelihood)**: Given an HMM λ and an observation sequence O, finds the probability that the observed sequence was produced by this model. The probability

is denoted as $P(O|\lambda)$ and is estimated with the forward algorithm.

- **Problem 2 (Decoding)**: Given an HMM λ and an observation sequence O, finds the best hidden state sequence Q. The Viterbi algorithm is used in this process.
- **Problem 3 (Learning)**: Given an observation sequence O and the set of states in the HMM, learns the HMM parameters that maximize the likelihood $P(O|\lambda)$. The standard algorithm for HMM training is the forward-backward algorithm, also known as Baum-Welch algorithm (Baum, 1972), a special case of the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

2.4.2 Context-dependent Triphones and Decision Tree

In a speech recognition task, an HMM can be assigned to each whole word, but it will need a huge amount of HMMs when the vocabulary size is big and there may not be enough data to model every word. Also, the trained model cannot be applied to a new word. To address this problem, the HMMs are initially applied to a higher granularity for speech – the phones, which are called the "monophone" models. In a monophone model, each phone is modelled with an HMM with a number of states. For English, there are forty to sixty phones according to different dictionaries. Usually, silence is modelled with five states and non-silence phones are modelled with three states. The acoustic model has fewer parameters to be trained with monophone models and any new word can be formed by concatenating the HMMs of the phone units.

Since the surrounding phone context greatly influences the pronunciation of each phone, context-dependent triphone models are used for acoustic modelling to capture the coarticulation effect. If the left phone is 'l' and the right phone is 'r', the current phone 'x' is modelled as 'l-x+r' in the context of a triphone model. There will be up to M^3 triphones for a system with M phones and there may not be enough samples to train each of them. In fact, most of these triples will never appear in the training data due to phonological

constraints. A standard approach to tackle the data sparsity is to apply tree-based state tying (S. J. Young et al., 1994).

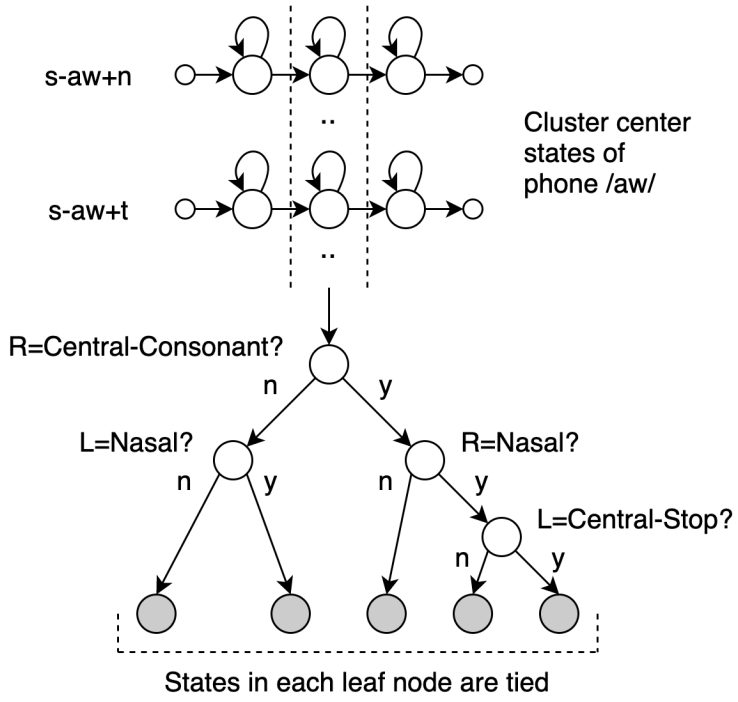


Figure 2.3: *Decision tree-based state tying (S. Young et al., 2002).*

The basic idea of the decision tree is to find similar triphones and share parameters between them. The tree based clustering uses a top-down approach to split a complete set of triphones to smaller clusters by asking questions about the samples. It can be applied at the triphone level (model sharing) or state level (state sharing). When state sharing is applied to all the triphones, one tree is constructed for each state of each triphone. Figure 2.3 illustrates an example of state tying – tying the centre states of all triphones of phone /aw/ (as in "out"). All states trickle down the tree and end up at one of the shaded terminal nodes (leaf nodes) depending on the answers to the questions. All of the states in the same leaf nodes are tied.

2.5 DNN-HMM based ASR systems

2.5.1 Introduction

The combination of artificial neural network (ANN) and HMMs as an alternative architecture for speech recognition started from the beginning of 1990s. It did not become popular until the early 2000s when DNN-HMMs were successfully applied to model context-dependent states by fine-tuning a pre-trained deep belief network (DBN).

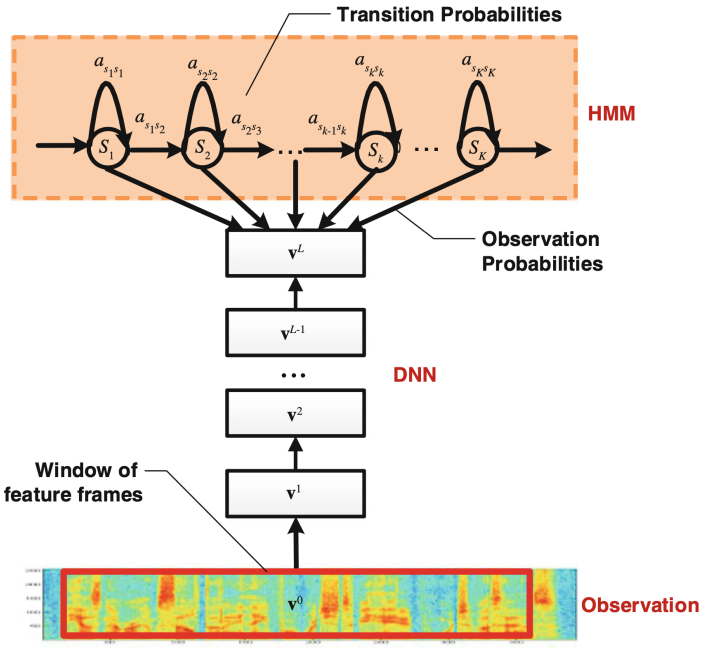


Figure 2.4: *The architecture of a DNN-HMM hybrid system (Yu and Deng, 2016).*

The architecture of a DNN-HMM hybrid system is illustrated in Figure 2.4. The HMM states correspond to the output of the neural network nodes (senones). In the case of a GMM-HMM model, a different GMM is used to model each different state as the GMM is a generative model. The DNN-HMM system is different from GMM-HMM system in that DNN is a discriminative model. Hence, a single DNN can be trained to estimate the conditional posterior probabilities for all states. Specifically, the i_{th} node in the output layer

is the posterior probability for state i given observation x:

$$o_i(x) = P(i|x). \tag{2.6}$$

The likelihood $p(x|i)$ is required during the decoding process, hence Bayes rule is applied to obtain the likelihood from the posterior probability:

$$p(x|i) = \frac{P(i|x)p(x)}{P(i)}, \tag{2.7}$$

where $P(i) = \frac{T_i}{T}$, estimated from the training set, is the prior probability of each state (senone), T_i and T are the number of frames labelled as state i and the total number of frames, respectively. $p(x)$ is independent of the word sequence and can be ignored, which leads to the scaled likelihood:

$$\overline{p}(x|i) = \frac{P(i|x)}{P(i)}. \tag{2.8}$$

Another difference between DNN-HMM and GMM-HMM is that the input to the DNN is typically not a single frame but a window of $[t-c, t+c]$ frames of input features, where t is the current frame and c is the context size. In a standard DNN-HMM system, a GMM-HMM is needed to provide the decision tree and the input labels for DNN training. The state-level forced alignments are usually generated using the Viterbi algorithm.

A neural network with at least two hidden layers can be called a deep neural network. There are multiple types of neural networks, e.g. multilayer perceptron (MLP), recurrent neural network (RNN), convolutional neural network (CNN), etc. Sometimes DNN specifically refer to neural networks trained by layer-wise training techniques. The deep belief network (DBN) is a classical deep learning structure which has a dominant role in speech recognition, it's also an important structure used in this thesis. More descriptions about DBN-DNN will be covered in the next section.

2.5.2 DBN-DNN with RBM pre-training

It has been empirically confirmed that a pre-trained DNN performs better than a DNN that uses random initialisation. The two-stage training procedure is a classical approach to train a DNN. In the first stage, a set of RBMs are trained, each has a visible layer and a latent layer. In the second stage, a DBN is formed by stacking the RBMs, then the output layer is added on top of the DBN, followed by fine-tuning the whole network with the back-propagation algorithm.

RBM

The restricted Boltzmann machine (RBM) is a stochastic generative neural network (Ackley et al., 1985; Hinton et al., 2006). An RBM is a variant of the Boltzmann machines with no visible-visible or hidden-hidden connections. It can be regarded as a two-layer neural network with an input layer (visible layer) and a hidden layer, as illustrated in Figure 2.5. The hidden unit usually takes binary values and follow Bernoulli distributions. The visible units may take binary or real values depending on the input types, resulting in a Bernoulli-Bernoulli RBM or Gaussian-Bernoulli RBM.

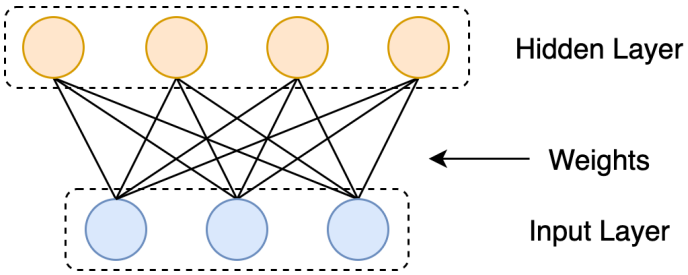


Figure 2.5: *An example of restricted Boltzmann machines (Yu and Deng, 2016).*

In the RBM, an energy is assigned to every configuration of visible vector v and hidden vector h. In the case of a Bernoulli-Bernoulli RBM, the energy is

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{a}^T\mathbf{v} - \mathbf{b}^T\mathbf{h} - \mathbf{h}^T\mathbf{W}\mathbf{v}, \tag{2.9}$$

where **a** and **b** are, respectively, the visible and hidden layer bias vectors, **W** is a weight matrix connecting the visible and hidden layer. If the visible units take real values, the RBM is often called the Gaussian-Bernoulli RBM and the assigned energy is

$$E(\mathbf{v}, \mathbf{h}) = -\frac{1}{2}(\mathbf{v}\text{-}\mathbf{a})^T(\mathbf{v}\text{-}\mathbf{a}) - \mathbf{b}^T\mathbf{h} - \mathbf{h}^T\mathbf{W}\mathbf{v}. \tag{2.10}$$

To train an RBM is to learn the parameters **W**, **a** and **b**. RBMs are usually trained using the Contrastive Divergence (CD) learning procedure (Hinton, 2002; Hinton, 2012). The approximation made by one-step CD algorithm is usually enough, although there could be more steps.

DBN-DNN and Fine-tuning

A DBN is a stack of RBMs, in which the hidden layer of previous RBM performs as the visible layer in the current RBM (Hinton et al., 2006). An example of a DBN-DNN with three hidden layers is illustrated in Figure 2.6.

Firstly, a Gaussian-Bernoulli RBM (GRBM) is trained to model real-valued acoustic coefficients with a window of frames. Then the states of the hidden units of the GRBM are used as the data to train an RBM, this can be repeated as many times as desired to produce multiple hidden layers (Hinton et al., 2012). The information about the HMM states, which the acoustic model will need to discriminate, are not used in the training of these GRBM and RBMs. Then the stack of RBMs is converted to a DBN by replacing the undirected connections of lower level RBMs by top-down, directed connections. Finally, a "softmax" output layer is added on top of the DBN with the weights between them initialised to zeros. The units in the output layers are the possible HMM states. The DBN-DNN is then discriminatively trained using the standard error back-propagation, this is called the fine-tuning process. The idea of fine-tuning can also be applied to adapt a trained model to a particular subset of the training set.

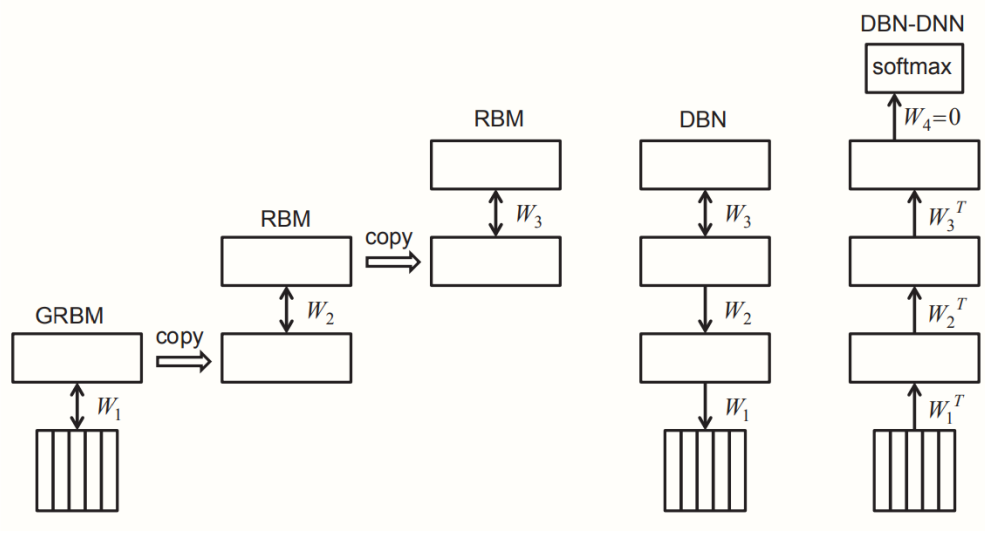


Figure 2.6: *The construction of a DBN-DNN with three hidden layers (Hinton et al., 2012).*

2.6 Acoustic Model Training Criteria

2.6.1 Introduction

In a DNN-HMM hybrid system, the DNN is trained to estimate the posterior probabilities for the HMM states. Specifically, for an observation o_{ut} corresponding to time t in utterance u, the output $y_{ut}(s)$ of the DNN for the HMM state s is obtained using the softmax activation function:

$$y_{ut}(s) \triangleq P(s|o_{ut}) = \frac{exp\{a_{ut}(s)\}}{\sum_{s'} exp\{a_{ut}(s')\}}, \tag{2.11}$$

where $a_{ut}(s)$ is the activation at the output layer corresponding to state s. The networks are trained to optimize a given training objective function using the standard error back-propagation (BP) (Rumelhart et al., 1986).

Cross-entropy has been used as the objective function in standard DNNs for speech recognition for decades. However, it is a frame-based criterion and speech recognition cares more about the sequence level correctness. The idea of using sequence-discriminative training in neural networks was proposed in the early 1990s (Bridle and Dodd, 1991; Krogh

and Riis, 1999), it did not become popular in speech recognition until the 2010s because of the computational constraints. Since then, various sequence-discriminative training criteria, like maximum mutual information (MMI), minimum phone error (MPE) and state-level minimum Bayes risk (sMBR), have been shown consistent gains for neural networks over cross-entropy training (Kingsbury, 2009; G. Wang and Sim, 2011; Kingsbury et al., 2012). Veselý et al. (2013) implemented sequence-discriminative training for neural networks in the Kaldi toolkit (Povey et al., 2011) – a widely used open source toolkit in speech recognition, speaker recognition, speaker diarisation, etc. Compared to a DNN with cross-entropy training, different sequence-discriminative criteria haven shown to decrease the word error rates by 7-9% relatively, on average. The implementation in Kaldi makes it possible for the wider community to apply sequence-discriminative training to different speech tasks.

2.6.2 Cross-entropy Training

In general, cross-entropy measures the relative entropy between two probability distributions over the same set of events. If $p(x)$ is the true distribution and $q(x)$ is the estimated distribution, the cross-entropy is defined as:

$$H(p,q) = -\sum_{\forall x} p(x) log(q(x)). \tag{2.12}$$

In neural networks, if $\mathbf{y}$ is the ground truth and $\hat{\mathbf{y}}$ is the estimate (output of the last layer), then the cross-entropy is

$$F_{CE} = -\mathbf{y} \cdot log(\hat{\mathbf{y}}), \tag{2.13}$$

where $\cdot$ is the inner product. Usually, $y = 1$ for correct samples and $y = 0$ for incorrect samples, the cross-entropy becomes the negative log posterior:

$$F_{CE} = -log(\hat{\mathbf{y}}). \tag{2.14}$$

Speech recognition is a multi-class classification problem, it is common to use the negative log posterior as the objective, which is also the expected cross-entropy between the

distribution represented by the reference labels and the predicted distribution $y(s)$:

$$F_{CE} = -\sum_{u=1}^{U}\sum_{t=1}^{T_u} logy_{ut}(s_{ut}), \tag{2.15}$$

where s_{ut} is the reference state label at time t for utterance u.

2.6.3 MMI

The maximum mutual information (Bahl et al., 1986) criterion in speech recognition calculates the mutual information between the distributions of the observations and word sequences:

$$F_{MMI} = \sum_{u} log\frac{p(O_u|S_u)^k P(W_u)}{\sum_W p(O_u|S)^k P(W)}, \tag{2.16}$$

where $O_u = \{o_{u1}, ..., o_{uT_u}\}$ is the sequence of all observations, W_u is the word-sequence in the reference for utterance u, $S_u = \{s_{u1}, ..., s_{uT_u}\}$ is the sequence of states corresponding to W_u and k is the acoustic scaling factor. The numerator $p(O_u|S_u)^k P(W_u)$ is the likelihood of data given correct word sequence. The denominator is the total likelihood of the data given all possible sequences, the sum is taken over all possible word sequences by the full acoustic and language models in recognition. The numerator and denominator lattices were generated for each training utterance to get the estimate. The objective of F_{MMI} is optimised by making the correct word sequence likely (maximizing the numerator) and all other word sequences unlikely (minimizing the denominator).

Boosted MMI

The boosted MMI is modified from MMI objective, to boost the likelihood of paths that contain more errors. The modified criterion is:

$$F_{BMMI} = \sum_{u} log\frac{p(O_u|S_u)^k P(W_u)}{\sum_W p(O_u|S)^k P(W) e^{-bA(W,W_U)}}, \tag{2.17}$$

where b is the boosting factor, $A(W, W_U)$ is the raw accuracy of word sequence W with respect to the reference W_u.

LF-MMI

The idea of lattice-free MMI is to get rid of the denominator lattice and to do the summation over all possible label sequences instead. To make its computation feasible, Povey et al. (2016) use a phone-level n-gram language model in place of the word-level language model and computes the objective function using neural network outputs at one third the standard frame rate. Xiong et al. (2017) use a mixed-history acoustic unit language model, where the probability of transitioning into a new context-dependent phonetic state (senone) is conditioned on both the senone and phone history. In their model, consecutive framewise occurrences of a single senone are compressed into a single occurrence, a variable-length N-gram language model is estimated from this data and the history state consists of the previous phone and previous senones within the current phone.

2.6.4 MPE and sMBR

The minimum Bayes risk (MBR) criterion (Povey, 2005; Kaiser et al., 2002; Gibson and Hain, 2006) are similar to MMI, except the objective of F_{MMI} is to minimize the expected sentence error, while MBR criteria are designed to minimize the expected error corresponding to different granularities of labels (phone level or state level). The objective function is:

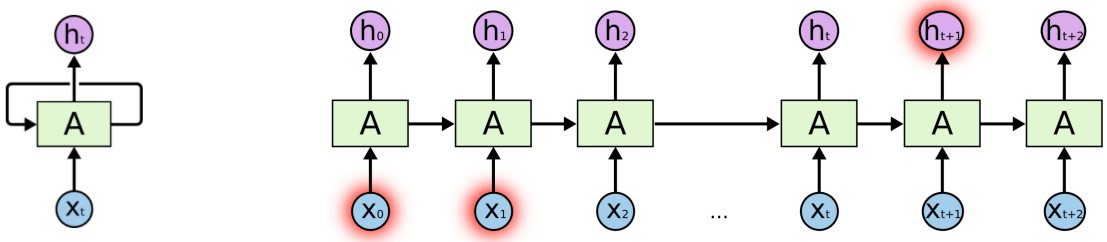
$$F_{MBR} = \sum_u log \frac{\sum_W p(O_u|S)^k P(W) A(W, W_u)}{\sum_{W'} p(O_u|S)^k P(W')} \tag{2.18}$$

where $A(W, W_u)$ is the raw accuracy – the number of correct phone labels (for MPE) and state labels (for sMBR) corresponding to the word sequence W with respect to that corresponding to the reference W_u. The denominator of F_{MBR} is the same as F_{MMI}, the $A(W, W_u)$ in the numerator improve the granularity from sentence to state/phone.

2.7 Long Short-Term Memory

2.7.1 Recurrent Neural Network

Recurrent Neural Network (RNN) is a kind of neural network that specializes in processing sequences. Traditional neural network assumes that all inputs (and outputs) are independent of each other, it does not memorize the past data and there is no future scope. The decisions are made based on the current input (or inputs within a context window). In some cases, the previous inputs are needed to predict the next output. RNNs are capable of handling sequential data, accepting the current data and previously input data.



(a) *RNNs have loops.*

(b) *An unrolled recurrent neural network (Olah, 2015).*

Figure 2.7: *(a) A recurrent neural network and (b) its unfolded structure.*

A typical recurrent neural network and how it looks like when unrolled are illustrated in Figure 2.7. RNN has loops which allow information to be passed from one step of the network to the next. Given an input sequence $x = (x_1, ..., x_T)$, a standard RNN computes the hidden state h_t and output y_t at time t by the following equation (Graves et al., 2013):

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \tag{2.19}$$

$$y_t = W_{hy}h_t + b_y, \tag{2.20}$$

where W denotes weight matrices and b denotes the bias vector. As Figure 2.7b shows, both the current data point and the previous hidden states are feed to the network to calculate the current hidden states.

In theory, RNNs can make use of the information in long sequences. In practice, they suffer from the vanishing gradients problem, hence are limited to looking back only a few steps (Hochreiter, 1991; Bengio et al., 1994). The gradients carry information that is needed for updating RNN parameters and when the gradient becomes smaller and smaller, no real learning is done as the parameter updates become insignificant. Long Short-Term Memory (LSTM) networks, a special type of RNN, was proposed to tackle the vanishing gradient problem (Hochreiter and Schmidhuber, 1997).

2.7.2 LSTM Networks

LSTMs are explicitly designed to avoid the long-term dependency problem. As introduced in Section 2.7.1, RNNs have a looped structure and the repeating module is quite simple (cell 'A' in Figure 2.7). LSTMs also have the form of a chain of repeating modules (cells) but the repeating module is more complicated than a standard RNN. The structure of an LSTM cell is depicted in Figure 2.8.

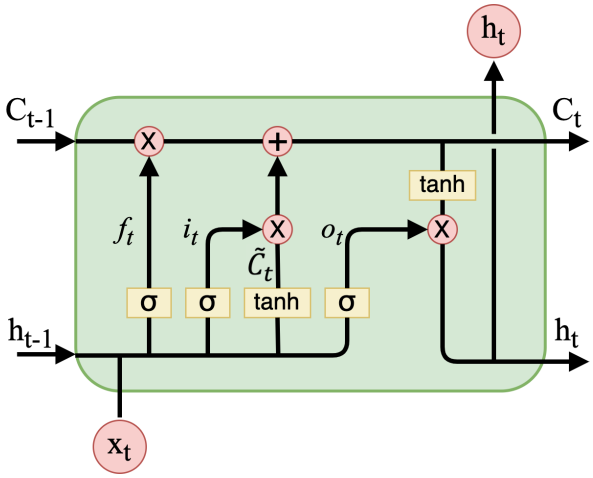


Figure 2.8: *Structure of the LSTM cell (Olah, 2015).*

An LSTM network has three gates that update and control the cell states: the forget gate, input gate and output gate. It also has a cell state to store information of the current cell. In Figure 2.8, the horizontal line running through the top of the diagram is the cell

state. The forget gate, on the left of the cell, decides what information from the previous cell state is worth remembering and forgets the irrelevant stuff. This is implemented using a sigmoid function:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \tag{2.21}$$

The input gate controls what new information will be stored in the cell state, the output of this gate is the element-wise product of the outputs of a sigmoid layer and a tanh layer. The sigmoid layer (Equation 2.22) decides what values to update and the tanh layer (Equation 2.23) creates the candidate values, $\tilde{C}_t$, that could be added to the state:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{2.22}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C). \tag{2.23}$$

After the forget gate and the input gate, the cell state can be updated:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t. \tag{2.24}$$

In the last step, the output gate filters the cell state and decides what will be passed as input in the next time step. It is implemented using a sigmoid function and the final output of the cell is h_t:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{2.25}$$

$$h_t = o_t * tanh(C_t). \tag{2.26}$$

LSTMs have been widely used and perform well on a large variety of problems. Many variants of LSTMs have been created, one of the most popular variants was introduced by Gers and Schmidhuber (2000), in which each gate layer looks at the cell states as well as the input and hidden sates:

$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f), \tag{2.27}$$

$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i), \tag{2.28}$$

$$\tilde{C}_t = tanh(W_C \cdot [C_{t-1}, h_{t-1}, x_t] + b_C). \tag{2.29}$$

2.8 Time-Delay Neural Network

2.8.1 Introduction

Recurrent neural network structures have been shown to efficiently model the long time dependencies in acoustic events, but the training of RNNs takes much longer compared to feed-forward networks due to the sequential nature of the learning algorithm. A time-delay neural network (TDNN) is also capable of modeling the long range temporal dependencies. The training and decoding time of a TDNN is comparable to a standard feed-forward neural network and the word error rates (WER) is comparable to an LSTM based neural network.

The TDNN was first proposed in 1989 for phoneme recognition (Waibel et al., 1989). In a task of recognising phonemes "B", "D" and "G", a TDNN model achieved a correct rate of 98.5% compared to 93.7% with the best HMM model. The popularity of TDNNs was picked up in 2015 when a more efficient setup was proposed, which is known as a TDNN with the sub-sampling scheme (Peddinti et al., 2015). A data-efficient alternative of TDNN was both suggested by Povey et al. (2018) and Pulugundla et al. (2018) in 2018, known as factorized time-delay neural network (TDNN-F) in the former and low-rank TDNN in the latter. They both have similar structures, while the former also introduced an extra training criterion for this structure to improve its performance. TDNN-F can efficiently reduce the computational cost and has been shown to do better than conventional DNN-HMM and TDNN systems (Povey et al., 2018; Pulugundla et al., 2018; F. Wu et al., 2019).

2.8.2 Neural Network Structure

In the previous section, we introduced how acoustic modelling with a neural network works. The neural network has the ability to model richer representations and the ability to use context to model the probability distribution of the observation likelihoods. When processing

a wider temporal context, the initial layer of a standard DNN learns an affine transform from the entire temporal context. However, in a TDNN architecture, each layer processes a context window from the previous layer and they are operated at different temporal resolution. It can be considered as a stack of 1D convolutional neural networks. The initial transforms are learnt on narrow contexts and the deeper layers learn from a wider context. Hence the higher layers are capable of learning wider temporal relationships.

The transforms in the TDNN architecture are tied across time steps and for this reason they are seen as a precursor to the convolutional neural networks (CNN). Due to tying, the lower layers of the network are updated by a gradient accumulated over all the time steps of the input temporal context during back-propagation (Peddinti et al., 2015). Thus the lower layers of the network are forced to learn translation invariant feature transforms. The tying of transforms also increases the statistical strength of the update.

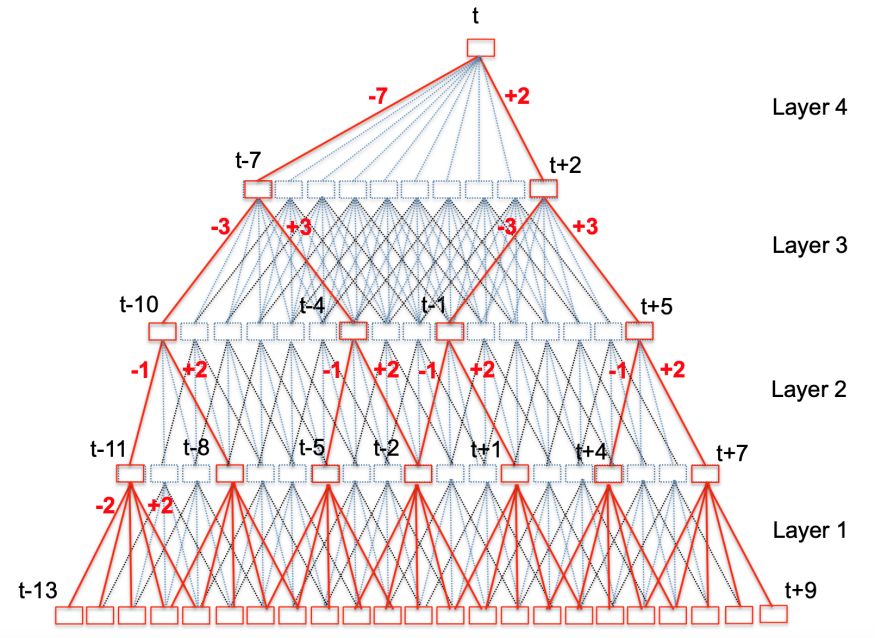


Figure 2.9: *An example of the TDNN model. The computation in TDNN with sub-sampling (red) and without sub-sampling (blue+red) (Peddinti et al., 2015).*

An example of the TDNN model is presented in Figure 2.9. Each rectangular block

stands for multiple nodes. A block in the input layer, t-th hidden layer and output layer presents D-dimensional input features, H_t hidden nodes and O output nodes, respectively. Figure 2.10a is a sub-component of the second layer in the TDNN architecture presented in Figure 2.9, the frame at time t-10 is calculated from the frame at time [t-11, t-8] in the previous layer. Figure 2.10b shows that each block contains multiple nodes and each coloured line represents an $N \times M$ weight matrix, where M and N are the numbers of nodes in the previous layer and current layer. Figure 2.10c presents a detailed version of Figure 2.10b where each weight matrix is unfolded.

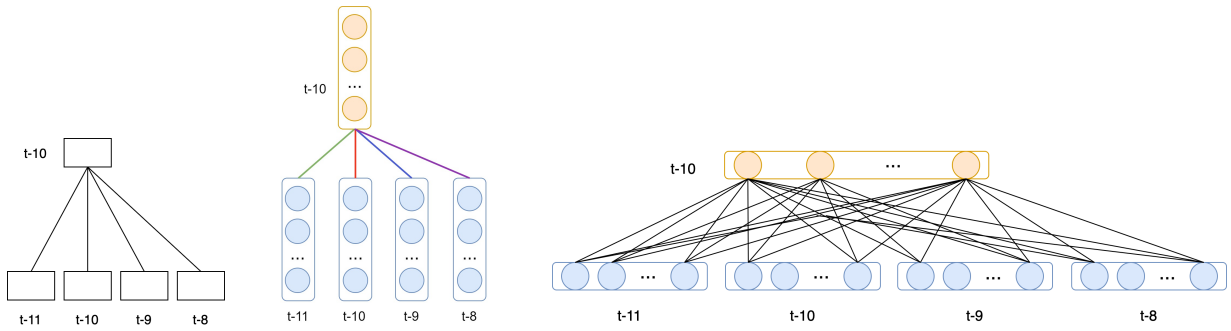


Figure 2.10: *Understanding the TDNN architecture. (a) A sub-component of a TDNN model (simplified version), (b) an illustration and (c) the unfolded version of the sub-component in the TDNN architecture.*

2.8.3 Sub-sampling

A major issue in a TDNN architecture is the linear increase in parameters with the increase in the input temporal context used to compute one output frame. Besides, each hidden activation is computed at all time steps, this results in large overlaps between input contexts of activations at neighboring time steps. Assuming that neighboring activations are correlated, a sub-sampling strategy was proposed to alleviate the problem of linear increase both in parameters and computation with the increase in input context (Peddinti et al., 2015).

In the proposed sub-sampling approach, no more than two frames, e.g. only the

frames at the edges of the temporal convolution kernel, are used as an input at each hidden layer. The context of the existing filters is increased to model wider temporal contexts and this does not increase the number of parameters due to the sub-sampling. However, it is important to make sure that the information from all the frames in the input context is used to compute each output frame.

Table 2.1: *Context specification of TDNN in Figure 2.9 (Peddinti et al., 2015).*

Layer	Input context	Input context with sub-sampling
1	$[-2, 2]$	$[-2, 2]$
2	$[-1, 2]$	$\{-1, 2\}$
3	$[-3, 3]$	$\{-3, 3\}$
4	$[-7, 2]$	$\{-7, 2\}$
5	$\{0\}$	$\{0\}$

A TDNN without sub-sampling (blue + red) and a TDNN with sub-sampling (red) are shown in Figure 2.9. It shows that sub-sampling computes activations for just a fraction of the time steps, as opposed to the standard TDNN which computes activations for each time step. Layerwise context specification, corresponding to this TDNN, is shown in Table 2.1. The notation $[-1, 2]$ means splicing together frames $t - 2$ through $t + 2$, which could be written as context $\{-1, 0, 1, 2\}$, and the notation $\{-1, 2\}$ means splicing together the input at the current frame minus 1 and the current frame plus 2. The same context modelling capability is maintained even though less activations are computed.

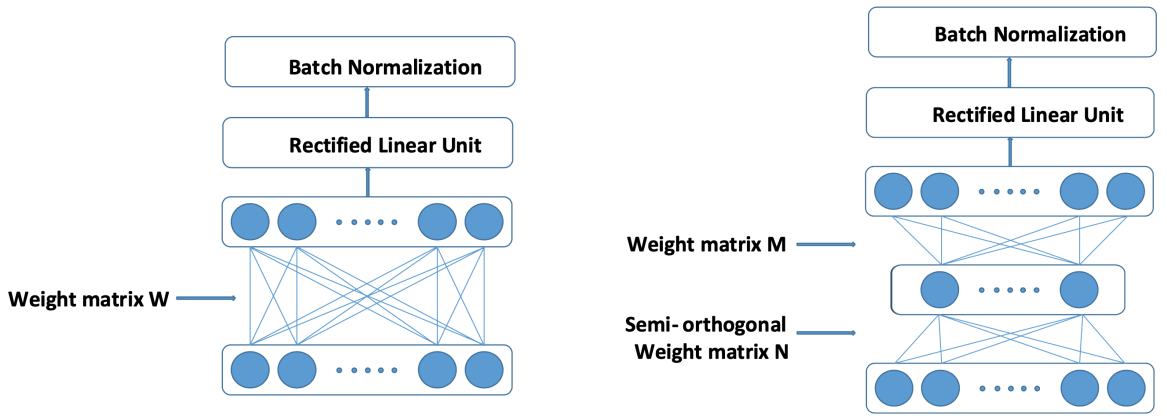
2.8.4 Factorized Time-Delay Neural Network

Factorized time-delay neural network (TDNN-F) improves the computation efficiency of TDNN 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).

$$W = U\Sigma V^T = MN,$$

where $\Sigma \in \mathbb{R}^{m\times n}$ is a non-negative, diagonal matrix, $M \in \mathbb{R}^{m\times k}$ and $N \in \mathbb{R}^{k\times n}$. The total number of parameters for this transformation can be reduced by choosing a suitable value of $k \leq min\{m, n\}$. It needs to ensure that one of the two sub-matrices is close to a semi-orthogonal matrix, as it is the equivalent of $U\Sigma$ or ΣV^T. Since k is much smaller than m and n, TDNN-F can also be considered as introducing an extra linear bottleneck layer into the traditional TDNN. Figure 2.11 shows the difference of a normal TDNN layer and a factorized TDNN layer.



(a) *One standard feed-forward layer.*

(b) *Factorized layer with semi-orthogonal constraint.*

Figure 2.11: *A normal TDNN layer and a factorized TDNN layer (Povey et al., 2018).*

A few training strategies can be applied to further improve the performance of a TDNN-F network (Povey et al., 2018):

- **Factorizing the convolution**: A TDNN-F with a constrained 2x1 convolution followed by a 2x1 convolution performs better than a constrained 3x1 convolution followed by 1x1 convolution. The former is equivalent to a TDNN layer splicing 2 frames followed by a TDNN layer splicing 2 frames, and the latter is equivalent to a TDNN layer splicing 3 frames followed immediately by a feed-forward layer. This is referred to as "factorizing the convolution".

- **3-stage splicing**: 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. If the hidden layer dimension is 1280 and the bottleneck dimension is 256, the dimension goes from 1280→256→256→1280 within one layer.
- **Training criteria interpolation**: Training using both lattice-free maximum mutual information (LF-MMI) and cross-entropy has been shown to be useful (F. Wu et al., 2019). A TDNN-F acoustic model is trained using LF-MMI loss interpolated with cross-entropy loss which is calculated from an additional output layer.
- Others: Dropout, factorizing the final layer, skip connections, etc., are also helpful.

2.9 Topological Manifolds

State-of-the-art ASR systems use a single deep neural network (DNN) to define a mapping f from the "acoustic space" A to the space P of vectors of phone (or senone) posterior probabilities, which are integrated into the Viterbi decoder for speech recognition. Although this is a single continuous mapping, in practice the DNN is trained to approximate a discontinuous function f whose outputs typically jump between 0 and 1 across phone state boundaries. Therefore, it may be advantageous to think of f as a *set* of continuous functions $\{f_1, \cdots, f_J\}$, with each function f_j defined on a subset $A_j \subset A$ of the acoustic space and $\bigcup_{j=1}^{J} A_j = A$. In this case, the appropriate mathematical structure is a non-linear topological manifold. There have been few studies that have shown the benefits of coding the acoustic speech signal in a non-linear manifold space (Jansen and Niyogi, 2006; Subramanya and Bilmes, 2009; Y. Liu and Kirchhoff, 2013).

In mathematics, an n-dimensional manifold is a topological space that is locally equivalent to n dimensional real Euclidean space $\mathbb{R}^n$ (J. Lee, 2010). A simple example of a 1-dimensional manifold is a circle C in the plane, any point on C has a neighbourhood that

can be "straightened out" to be an open interval in $\mathbb{R} = \mathbb{R}^1$. However, C cannot be embedded as a whole as a subset of $\mathbb{R}^1$ (Bai et al., 2017).

In context of speech analysis, the manifold structure provides a tool to exploit the fact that different phonetic classes employ different production mechanisms and are best described by different types of features. Intuitively, one might hope that the subsets A_j correspond to broad phonetic categories. The idea of phone-dependent feature extraction is well-established. For example, while vocal tract resonance frequencies provide a natural description of vowels, unvoiced consonants are better described in terms of duration and mean energies in key frequency bands (F. Li et al., 2010; F. Li et al., 2012; K. Stevens and Blumstein, 1978; Heinz and K. N. Stevens, 1961; Raphael, 1972; Wilde, 1995; Khasanova et al., 2014). There are also a number of studies that use broad phone class (BPC) -dependent classifiers to focus on subtle differences between phones within a BPC (Scanlon et al., 2007).

A two-level *linear* computational model motivated by these considerations is presented by H. Huang et al. (2016). The first level comprises a set of discriminative linear transforms, one for each of a set of overlapping BPCs, that are used for feature extraction. The transforms are obtained using variants of linear discriminant analysis (LDA). Each transform is applied to an acoustic feature vector and k-nearest neighbour methods are used to estimate probabilities of BPCs and phones, which are then combined in the second level to estimate posterior probabilities and hence to classify the acoustic vector. This two-level linear classifier obtained slightly better results compared to a single transform on frame-level phone classification experiments on TIMIT (Garofolo et al., 1993).

Inspired by these observations, Bai et al. (2017) introduced a two-level *non-linear* model, referred to as a BPC-DNN. Their premise was that it would be advantageous to replace a single 'global' DNN with several BPC-dependent DNNs. In the first level of the BPC-DNN, several small, separate DNNs were applied to different BPCs. For each BPC, a DNN was trained to map acoustic features onto a vector of posterior probabilities of the

phones or senones within the BPC, plus an "outside-BPC" class. In the second level the outputs of these DNNs were passed as input to another DNN, the fusion network, which transformed them into a single phone or senone posterior probability vector for frame level classification. The BPC-DNN is related to Wu and Gales' multi-basis adaptive neural network (MBANN) (C. Wu and Gales, 2015), in which parallel component DNNs correspond to different speaker types.

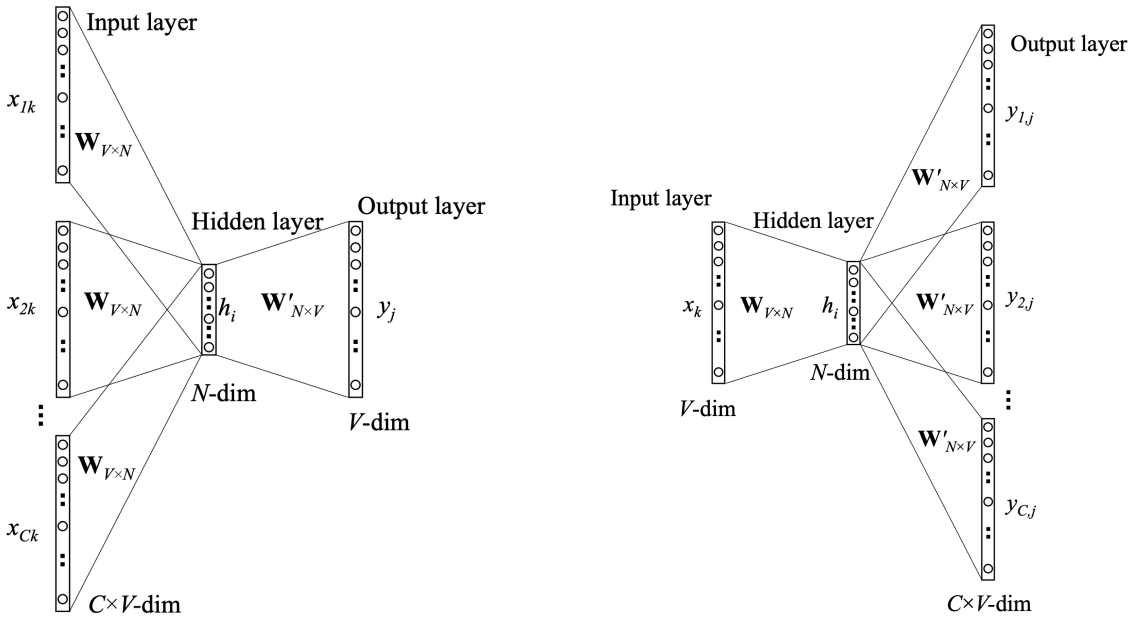
An obstacle to the application of a topological manifold model to acoustic speech analysis is the need to cover the acoustic space A with phonetically meaningful subsets A_i on which the "feature extraction" transforms f_i are defined. In the approach described here, this problem is avoided by applying the BPC-dependent DNN for a particular BPC to the whole of A but only mapping frames corresponding to phones in the BPC to the correct phone class. All other frames are mapped to the "outside-BPC" category.

It was shown in Bai et al. (2017) that the BPC-DNN model gives statistically significant improvements in phone-classification of feature vectors, compared with a single global DNN. We extended this work to full phone recognition by passing the output of the fusion network to a Viterbi decoder in M. Qian et al. (2018a). Our work confirms that the improvement in frame phone classification accuracy using BPC-DNNs can be extended to phone recognition. Specifically, a reduction in phone error rate of 6% relative to a conventional DNN is obtained using a BPC-DNN with fewer parameters. The details of this related work are included in Appendix A.

2.10 Word and Document Embeddings

Word embedding is a vector representation of document vocabularies. Traditional embeddings are frequency-based, e.g. count vectors, Term Frequency - Inverse Document Frequency (TF-IDF) vectors and co-occurrence vectors. With these representations, words with similar

meanings do not necessarily have similar representations. Recently proposed prediction-based embeddings are capable of capturing the context of a word in a document, semantic and syntactic similarity, the relation with other words, etc. Word2Vec (Mikolov et al., 2013a) is one popular technique to learn prediction-based embeddings using a shallow neural network. It can be obtained using a skip-gram or continuous bag-of-word (CBOW) algorithm. The neural network architecture of both algorithms are illustrated in Figure 2.12 (Rong, 2014).



(a) *CBOW* (b) *Skip-gram*

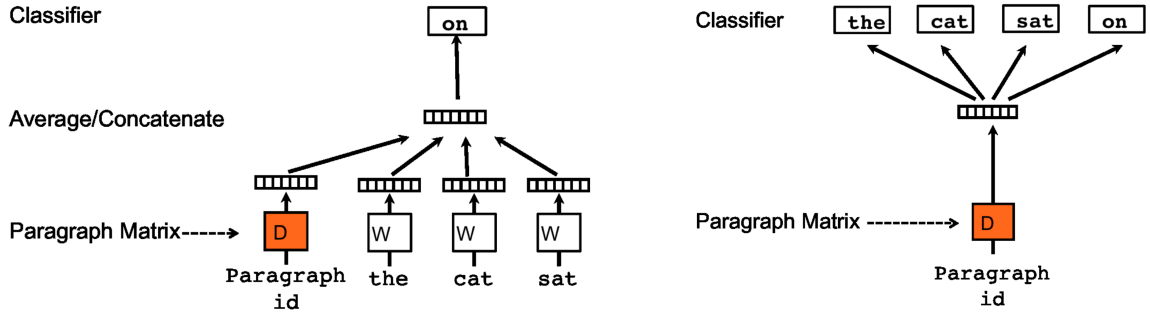
Figure 2.12: *Neural network model structures for learning Word2Vec embeddings. The CBOW architecture predicts the current word based on the context and the Skip-gram architecture predicts surroundings given the current word (Rong, 2014).*

In a CBOW architecture, each word in the input layer is encoded into a one-hot vector, the number of vectors depends on the context size. The weight between the input layer and the hidden layer can be represented by a $V \times N$ matrix W, where V is the vocabulary size and N is the hidden layer size. Each row of W is the N-dimension vector representation v_w of the associated word of the input layer. The hidden layer of the CBOW model takes

the concatenation, average or sum of the vectors of the input context words. The weight between the hidden layer and the output layer is a different weight matrix W' which is an $N \times V$ matrix. Note, matrix W' is not the transpose of matrix W. The j-th column of matrix W' is vector v'_{wj} associating the j-th word in the output layer. Vector v_w and v'_w are two representations of the word w, often referred to as "input vector" and "output vector". In most cases, the input vector has been used as the embedding for the word. The CBOW model can be considered as predicting the current word based on the context. While the skip-gram model is the opposite, it predicts the surroundings given the current word. The input vector of the skip-gram model is also one-hot encoded vector. Instead of having one multinomial distribution on the output layer, a skip-gram model has C multinomial distributions where C is the context size. Each output is computed with the same weight matrix W'.

The models described above have been extended to go beyond word level and achieve phrase level or sentence level representations. Mikolov et al. (2013b) proposed two models, namely the Distributed Memory Model of Paragraph Vector (PV-DM) and the Distributed Bag of Words version of Paragraph Vector (PV-DBOW), to train Paragraph Vectors (also known as "Doc2Vec"), which are capable of constructing representations of input sequence of variable length. The frameworks of these two models are presented in Figure 2.13. The PV-DM model is inspired by the CBOW model in Word2Vec, it randomly samples fixed-length consecutive words from a paragraph using a sliding window and predicts a centre word from the randomly sampled words by taking the context words and a paragraph id as input. Every paragraph is encoded into a one-hot vector in the input layer and represented by a column in matrix D which is between the input layer and the hidden layer. The paragraph vector is shared across all contexts sampled from the same paragraph but not across paragraphs. The word vector matrix W is shared across paragraphs. The PV-DBOW model is different from the PV-DM model. It ignores the context words in the model and forces the model to predict words randomly sampled from the paragraph in the output. The PV-DM model takes into consideration of the word order, at least in a small context. While the PV-DBOW

model is not able to capture the word order.



(a) *PV-DM* (b) *PV-DBOW*

Figure 2.13: *Frameworks for learning Paragraph Vectors (Doc2Vec). The paragraph vector in a PV-DM model can act as a memory of the topic of the paragraph. In a PV-DBOW model, the paragraph vector is trained to predict the words in a context window (Mikolov et al., 2013b).*

In the recent decade, different types of word embeddings are developed and they all have their advantages. For example, the Global Vectors for Word Representation (GloVe) is able to capture global statistics (Pennington et al., 2014). Both of the Word2Vec and GloVe embeddings are context-independent embeddings, while Bidirectional Encoder Representations from Transformers (BERT) and Embeddings from Language Models (ELMo) generate different word embeddings for a word considering the context the word has been used (Devlin et al., 2019; Peters et al., 2018). Besides, BERT learns embeddings for subwords and ELMo is character-based, both of them are capable of generating embeddings for unknown or out-of-vocabulary words. This thesis mainly used the Word2Vec and Doc2Vec embeddings, hence, the details of other embedddings will not be presented here.

Chapter 3

Speech Corpora

This chapter describes the speech corpora used in the thesis. Two children's speech corpora were involved to analysis the English speech of German-speaking Swiss children and Italian-speaking children, namely the CALL Shared Task corpus (CALL-ST) and the TLT-school corpus. The AMI corpus and PF-STAR corpus were used to deal with the lack of data problem in ASR where the target is speech recognition for non-native children. The WSJCAM0 was used in officially provided speech recognition system in the CALL Shared Tasks.

3.1 Spoken CALL Shared Tasks and the CALL-ST Corpus

The CALL Shared Task corpus (CALL-ST) includes English responses from German-speaking Swiss teenagers interacting with the CALL-SLT[1] system (Rayner et al., 2010; Rayner et al., 2014; Baur, 2015), which has been under development at the University of Geneva since

[1]SLT: Spoken Language Technology

2009[2]. It was collected in 15 classes at 7 different schools in 2014 and early 2015 (Baur et al., 2017). The teenagers involved are school students in their first to third year of learning English aged 12 to 15 years. Three Spoken CALL Shared Tasks (ST) were held in 2017, 2018 and 2019, respectively, based on the recordings from the CALL-ST corpus.

Section 3.1.1 will introduce the collection of the CALL-ST corpus and Section 3.1.2 will introduce the three Spoken CALL Shared Tasks. The data release, the judgement labels of the data and the scoring metric of the shared tasks will be presented in Section 3.1.3, Section 3.1.4 and Section 3.1.5, respectively.

3.1.1 Corpus Collection

Figure 3.1: *CALL-SLT interface (Baur et al., 2016).*

The collected corpus is based on a textbook commonly used in German-speaking Switzerland that consists of 8 lessons. Each prompt in the course is a combination of a written text in L1 (German) and a multimedia file in L2 (English). The screenshot in Figure 3.1 presents a typical example of the data collection process: the system plays a short animated clip with an English native speaker asking a question in English (top) and

[2]http://callslt.unige.ch/demos-and-resources/

simultaneously displays the German text which indicates how the student is supposed to answer in L2 (middle). Students could ask for help by pressing the question-mark icon (bottom right), an example response will show in the bottom pane and also be played in audio form. Once the student has spoken into the headset or onboard microphone, the system will perform speech recognition and then match the recognised utterance against the prompt's specification of correct answers. Each prompt permits many variations of responses, emphasising the communication approaches rather than minor grammatical or pronunciation flaws in the utterances. The recording sampling rate is 8kHz with 16 bit sample resolution.

The collected corpus contains 38,771 spontaneous speech recordings in total. Subsets of the corpus were selected and released for the three Spoken CALL Shared Tasks, data release will be described in Section 3.1.3.

3.1.2 Spoken CALL Shared Tasks

The first shared task (ST) for Computer-Assisted Language Learning (CALL), referred to as "2017 Spoken CALL Shared Task" or "ST1", was led by the University of Geneva with support from the University of Birmingham and Radboud University using the data from the CALL-ST corpus (Baur et al., 2017). Following the success of the first edition, the above consortium of universities along with the University of Cambridge introduced the second edition of the ST in 2018 (Baur et al., 2018) and the third edition in 2019 (Baur et al., 2019), referred to as "2018 Spoken CALL Shared Task" or "ST2" and "2019 Spoken CALL Shared Task" or "ST3", respectively.

The three shared tasks all used data from the CALL-ST corpus, but with different subsets. The detailed description of the released datasets will be presented in Section 3.1.3. The items in the data are prompt-response pairs, where the prompt is a piece of German text and the response is an utterance spoken in English and recorded as an audio file. The

challenge of the task is to label pairs as "accept" or "reject", accepting responses which are grammatically and linguistically correct and rejecting those incorrect either in grammar or meaning according to the judgments created either by a panel of human listeners or a combination of machines and human annotators. The approaches to annotating the judgments for the shared tasks will be described in Section 3.1.4.

The shared task system consists of a speech processing and a text processing component as depicted in Figure 3.2. The first part is an ASR system to convert a given audio recording of the student response into a text. The second part is a text processing component which takes the transcribed response and makes a judgment of whether the utterance is accepted or rejected according to the language and meaning annotations.

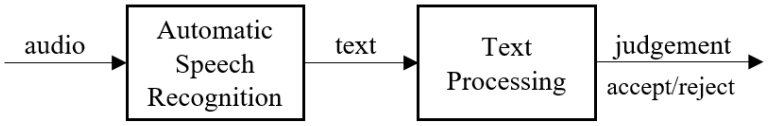


Figure 3.2: *The structure of a Spoken CALL Shared Task System.*

Each shared task consists of two versions: a speech-processing version and a text-processing version. The aim of the two versions are the same, but they have different items provided as system input. In the speech-processing version of the CALL shared task, each item consists of an identifier, a German text prompt and an audio file containing an English language response. For the text-processing version, there is an extra text string representing the speech recognition result of the audio file. The recognition texts for the text-processing version in ST1 were obtained from either the official baseline Kaldi ASR system or the Nuance ASR used in the original CALL-SLT system, while an improved Kaldi ASR system was used to produce the recognition texts for the text-processing version of ST2 and ST3. This thesis is mainly concerned with the speech-processing version but the text-processing version has also been improved. The work concerned with the speech processing component and text processing component will be presented in Chapter 4 and Chapter 5, respectively.

3.1.3 Data Release for Shared Tasks

Each data in the released corpus consists of a German prompt, an English recording responded by the students along with a transcription of the recording and the language and meaning judgments of the response. The transcription and judgments of the test sets were hidden until the end of the challenges. The number of speakers, utterances and total length of each dataset are given in Table 3.1.

Table 3.1: *Statistics of the Shared Task 1 (ST1), Shared Task 2 (ST2) and Shared Task 3 (ST3) datasets.*

Dataset	Release	#Speakers	#Utterance	Duration
ST1-TRAIN	ST1	66[1]	5222	4.80
ST1-TST	ST1	25[1]	996	0.89
ST2-TRAIN	ST2	58	6698	5.99
ST2-TST	ST2	-	1000	0.91
ST3-TST	ST3	-	1000	0.99
Total	-	-	14916	13.58

[1] The speaker information was reported in (Baur et al., 2017) but not available in the released corpus.

The training (ST1-TRAIN) and test (ST1-TST) sets of the 2017 Spoken CALL Shared Task (ST1) were released in July 2016 and March 2017, respectively. The training set contains 5,222 utterances and the test set contains 996 utterances. In October 2017 and February 2018, the training (ST2-TRAIN) and test (ST2-TST) data of the second edition of the CALL Shared Task (ST2) were released. Only a test set (ST3-TST) were released in April 2019 for Shared Task 3 (ST3). Among all of these datasets, ST2-TRAIN is the only dataset that provides the speaker IDs for each utterance. The number of speakers for ST1-TRAIN and ST1-TST are reported in (Baur et al., 2017) but not available in the released data, no speaker information is provided for ST2-TST or ST3-TST.

3.1.4 Language and Meaning Judgements

The released datasets contain the language and meaning judgements, which were hidden until the end of challenges for the test sets. A judgement of "correct" in language means it's a fully correct response to the prompt, "incorrect" in language and "correct" in meaning means that it is semantically correct and linguistically incorrect, "incorrect" in language and meaning means that it is incorrect both in grammar and meaning. A few examples are shown in Table 3.2.

Table 3.2: *Examples for the language and meaning judgements in the annotated data. Prompt "Frag: Zimmer für 6 Nächte" means "Request: room for 6 nights" (Baur et al., 2017).*

Prompt	Transcription	Language	Meaning
Frag: Zimmer für 6 Nächte	I would like a room for six nights	correct	correct
Frag: Zimmer für 6 Nächte	I wants a room for six nights	incorrect	correct
Frag: Zimmer für 6 Nächte	I want a room for five nights	incorrect	incorrect
Frag: Zimmer für 6 Nächte	It's raining outside	incorrect	incorrect

The linguistic and semantic correctness for ST1 were judged by three native English speakers based on the prompts and the transcriptions which were produced by German/Swiss German speakers fluent in English. This was supposed to create high reliable judgements, however there were more noises in the annotations than expected and hence perhaps as many as 3-4% judgements were incorrect. To minimize incorrect judgements in ST1-TST, organizers rechecked the judgements after receiving the submissions. One English native speaker and one German native speaker fluent in English listened to those examples for which there were a large number of submissions disagree with the judgements, implying that the judgements were mostly likely to be incorrect, and discussed each one until they had reached clear agreement.

The annotation of the judgements for ST2-TRAIN involved four best assessment

systems from the first shared Task (M. Qian et al., 2017; Magooda and Litman, 2017; Oh et al., 2017; Evanini et al., 2017) and three human annotators. Data can be divided into three groups according to the first round judgements made by the four machines: unanimous (4–0) agreement for 70% of the utterances, 3-1 agreement for 22% and a 2-2 split for 8% of the utterances. Three English native speakers independently annotated 200 utterances from each group and decided to consider the 4-0 group reliably judged because they had agreed about 98% of the utterances with the machines. After two annotators independently judged the remaining 3-1 and 2-2 portions, the third annotator re-judged those utterances for which the two annotators had conflict judgements and 20% utterances on which the two annotators had agreed. At the end of the annotation process, the released training set was divided into groups A, B and C by descending reliability:

A 5526 utterances. Either the machines are 4–0 and at least one human supports them, or the machines are 3–1 and all three humans support them.

B 873 utterances. All three humans agree, and either one or two machines support them.

C 299 utterances. Remaining cases.

The annotation for ST2-TST was done in a simpler way in order to keep the material secret from the potential competitors. Two native speakers of English independently annotated 1800 items previously not used in the Shared Task. After removing items wherever the two annotators disagreed or at least one had flagged their judgement as 'uncertain', 1000 items were randomly chosen from the remainder to be the test set. One native English speaker with moderate German and one native German speaker with near-native English jointly listened to the items again after the submission, focusing in particular on examples where many entries disagreed with the initial judgement. This final round annotation corrected a small number of judgements.

For ST3-TST, 1785 utterances from 25 speakers who had not appeared in previously released data were transcribed and separately annotated by three native speakers of English.

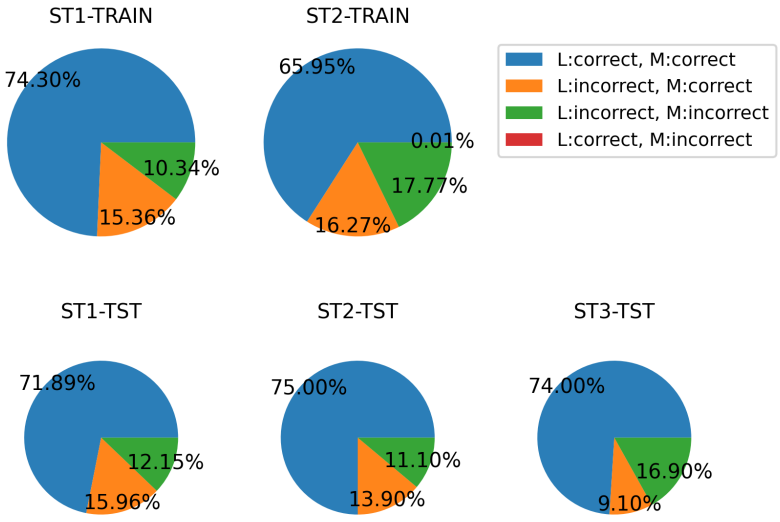


Figure 3.3: *Language and meaning judgements for each dataset in the CALL-ST corpus.*

Items where the three judges were not unanimous on either the language or the meaning judgement were removed. Because incorrect responses were under-represented in the remaining utterances, all of them (260) were kept, then 740 "correct" responses were randomly selected to produce a test set of 1000 items.

The distribution of categories of the judgements in different datasets are shown in the pie charts in Figure 3.3. There is a wrong judgement in ST2-TRAIN with correct language and incorrect meaning – "correct" language means it's a fully correct response so the meaning should be correct as well. For each dataset, the "correct" samples are more than the "incorrect" samples, making up 66% to 75% of the dataset.

3.1.5 Scoring Metric

Comparing the system's judgments with the human language and meaning annotations, the result for each response falls into one of the following categories:

i Correct Accept (CA) – sentence that is labelled as correct both in language and

meaning is accepted by the system;

ii False Reject (FR) – sentence that is correct linguistically and semantically is rejected;

iii Correct Reject (CR) – sentence that is incorrect either in grammar or in meaning is rejected;

iv False Accept (FA) – an incorrect sentence is accepted. The FAs are split into "Plain FAs" (PFAs) and "Gross FAs" (GFAs), corresponding to an FA of a response that is incorrect in language but has correct meaning and that is incorrect in both linguistic and semantic sense, respectively.

In calculating the overall FA, the GFAs are given k times heavier weight than PFAs. The FA is calculated as $FA = PFA + k \times GFA$, with $k = 3$.

The challenge used originally the following metrics: F-measure, scoring accuracy (SA), and differential response score (D). The F-measure is defined as $F = \frac{2PR}{P+R}$, where P and R denote the precision and recall, respectively, being defined as $P = \frac{CA}{CA+FA}$ and $R = \frac{CA}{CA+FR}$. The SA is defined as $SA = \frac{CA+CR}{CA+CR+FA+FR}$. The evaluation of the overall quality of the systems in the Shared Task 1 and 2 is performed using a differential response score, D, which is defined as the ratio of the reject rate on incorrect answers (iRej) to the reject rate on correct utterances (cRej) (Baur et al., 2016), i.e.,

$$D = \frac{iRej}{cRej} = \frac{CR/(CR+FA)}{FR/(FR+CA)} = \frac{CR(FR+CA)}{FR(CR+FA)}. \tag{3.1}$$

After Shared Task 2, D_a and D_{full} metrics were added. The D_a score is defined similarly as D but with concern on acceptance rate, i.e.,

$$D_a = \frac{1-cRej}{1-iRej} = \frac{CA/(CA+FR)}{FA/(FA+CR)} = \frac{CA(FA+CR)}{FA(CA+FR)}. \tag{3.2}$$

The D_{full} is the geometric average of D and D_a, i.e., $D_{full} = \sqrt{DD_a}$, which has been claimed to be more reasonable and more challenge (Baur et al., 2018). This is the metric used to rank the submissions in ST3. Although D_{full} was only used in ST3, the D_{full} scores for the ST1 and ST2 results will also be reported to have a better vision of the improvements between the systems.

3.2 TLT-school corpus

3.2.1 Introduction

The "Trentino Language Testing" in schools (TLT-school) corpus was acquired by Fondazione Bruno Kessler (FBK) in schools for measuring L2 linguistic competence of Italian students taking proficiency tests in both English and German (Gretter et al., 2020a). The collected data contains both written and spoken data from Italian students ranging from 9 to 16 years old, belonging to four different school grade levels and three CEFR (Common European Framework of Reference for Languages) proficiency levels (A1, A2, B1). Age ranges of students at A1, A2 and B1 proficiency level are 9-10, 12-13 and 14-16 years old, respectively. It is worth noting that students were taking tests by answering question prompts and the proficiency level corresponds to the prompts the students were answering, it does not necessarily reflect the actual language performance of each response. Statistics of level, grade, age and number of students participated in the tests are presented in Table 3.3, most of the students did both the English and the German test.

Table 3.3: *Statistics of the TLT-school corpus: proficiency level, grade, age and number of participants (Gretter et al., 2020a).*

CEFR	Grade, School	Age	Number of students		
			2016	2017	2018
A1	5, primary	9-10	1074	320	517
A2	8, secondary	12-13	1521	111	614
B1	10, high school	14-15	378	124	1112
B1	11, high school	15-16	141	0	467
Total	5-11	9-16	3114	555	2710

3.2.2 TLT-school Spoken Data

The spoken data of the "TLT-school" corpus was collected in classrooms. Around 20 students took the test at the same time in the same classrooms. The data was recorded at 16 kHz rate with 16 bit sample resolution, the types of equipment depend on each school. The English part of the corpus was collected in 2016, 2017 and 2018, involving 3000 Italian boys. A subset of the English responses was released for the Shared Task on Automatic Speech Recognition for Non-Native Children's Speech[3], which is a special session in INTERSPEECH 2020. Although the audio quality of the corpus is good, there are a few facts making it difficult for ASR:

- It is speech recognition for both child and non-native speech.
- There exists a large number of spontaneous speech phenomena (hesitations, false starts, fragments of words, giggling, coughs, etc.).
- It is Italian children taking both English and German test, hence English, Italian and German words are frequently uttered in response to a single question.
- As there are 20 students taking the tests in the same recording room, it is common that the speech from classmates or teachers overlaps with the participant's speech.
- There exists a significant amount of background noise and other speech (e.g. comments, jokes with mates, indications from the teachers, etc.), especially at the end of the utterance, due to the fact that the microphone remains open for a fixed duration which depends on the question.

3.2.3 Data Release

The released datasets consist of a 9 hour training set (TLT_Train1P or TLT9h), a 2 hour development set (TLT-Dev), which were both released on 7th February 2020, and an additional

[3]https://sites.google.com/view/wocci/home/interspeech-2020-special-session

40 hour training set (TLT_Train2P, or TLT40h), which was released one week later. The evaluation set (TLT-Eval) for the shared task was released on 17th April 2020. Only these released English subsets of the "TLT-school" corpus were used in the experiments presented in this thesis. Table 3.4 presents details of the datasets.

Table 3.4: *Statistics of TLT-school subsets.*

Dataset	Year	#Speakers	#Prompts	#Utterance	Duration
TLT_Train1P	2017	338	24	2299	8.98h
TLT_Train2P	2016, 2018	3111	109	11700	40.11h
TLT-Dev	2017	84	24	562	2.09h
TLT-Eval	2017	84	24	578	2.34h

3.2.4 Annotation

The subsets TLT_Train1P, TLT-Dev and TLT-Eval were collected in 2017 and were carefully transcribed following transcription rules described below[4]. An initial set of transcription guidelines were made and used by 5 researchers to manually transcribe about 20 minutes of recordings, this led to a discussion and a second set of guidelines were originated. Then, utterances answering the same question were concatenated to blocks of 5 minutes each for transcription, where the question is explicitly reported at the beginning of the block. This is helpful to transcribe some poorly pronounced words or phrases. About 30 students from two Italian linguistic high schools were engaged to perform manual transcriptions. The resulting transcription contains the following special cases:

- non-English word sequences are included in '@it()' for Italian, '@de()' for German and '@unk()' for non-existent words for the three languages considered;

[4]TLT2017ManTransGuidelines.pdf

- hesitations or noises are labelled with initial '@', including '@ae', '@ahm', '@breath', '@cough', '@laugh', etc.;
- truncated and incomprehensible words start or end with '-', e.g. '-vourite' (truncated 'favourite'), 'sub-' (truncated 'subject').
- badly pronounced words are marked with an initial '#'.

The 40 hours training set, TLT_Train2P, was selected from the English recordings collected in 2016 and 2018 and was annotated by the Educational Testing Service. This manual annotation suffers from incomplete knowledge of the Italian language, in particular Italian names and geographical Trentino toponyms. Hence, Italian or German phrases were not marked in TLT_Train2P as they were in TLT_Train1P, '<unk>' is used to mark all types of non-English words or phrases including German, Italian and non-existent words in three languages. Also, fewer types of noises and hesitations were labelled in "TLT_Train2P" – only '@ns', '@uh' and '@um' were used. Truncated words start or end with '-' as in TLT_Train1P, badly pronounced words were not marked.

3.2.5 Prompts

Two prompt files are provided which include the specific questions for each prompt. A few examples of prompt questions are presented in Table 3.5. Questions in A1 level include four introductory questions and a list of questions putting the students in the role of a customer in a pizza place. Tests at A2 level consist of small talk questions related to everyday life situations which expect more open-ended answers. Questions in B1 level are similar to A2 ones, but include a role-play activity in the final part allowing a good amount of freedom and creativity in answering the questions.

The prompt ID contains the year when the prompt was used for data collection, the level of proficiency and three numbers which are associated with the proficiency level. In

Table 3.5: *Examples of prompts.*

PromptId	Question
Ip17_A1_en_6_8_100	You are entering a pizzeria to buy a take-away pizza. Talk to the cashier and order your pizza. - Hello! How are you? (Expected answer: "fine, thanks" or similar)
Ip17_A1_en_6_8_102	Which pizza would you like? A cheese and tomato pizza, a pizza with mushrooms or a salami pizza?
Ip16_A2_en_22_107_100	Did you like your visit to Edinburgh? Why?
Ip16_A2_en_22_107_101	Was it your first time in Edinburgh? When did you go last?
Ip16_A2_en_22_107_103	Would you visit Edinburgh again? Why?
Ip16_B1_en_35_248_100	How do you usually get to school? How long is your journey?
Ip16_B1_en_35_248_105	Which new sport or hobby would you like to learn? Why?
Ip18_B1_en_36_29_100	Your friend doesn't do much sport; convince him that playing sports is useful and fun and is a healthy way to spend your free time. Give some advice on how to get started and get him involved in your activity.

some cases, the test takers were put in a role play situation and a list of break-down questions were used in which the first two numbers of the prompts are the same. Each audio file has a name which is composed by a speaker ID followed by a prompt ID, the speaker ID and prompt ID vary in different datasets. Below are two examples in TLT_Train1P and TLT_Train2P:

- "1010101_en_21_19_100.wav";
- "speakerIp16_A1_001001001001-promptIp16_A1_en_5_53_101.wav".

In above examples, '1010101' and 'speakerIp16_A1_00100100100' are the speaker IDs, 'en_21_19_100' and 'promptIp16_A1_en_5_53_101' are the prompt IDs. The audios in TLT-Dev and TLT-Eval have the same name format as in TLT_Train1P.

3.3 AMI Corpus

3.3.1 Overview

The AMI corpus consists of 100 hours of adults meeting recordings[5] (Carletta et al., 2005; Carletta, 2006). Although the recordings are in English, English is not the first language for the majority of the participants. The AMI corpus was recorded in various conditions with multiple devices. Recordings recorded in three conditions are available: IHM (Independent Headset Microphone), MDM (Multiple Distant Microphone) and SDM (Single Distant Microphone). The IHM data from the AMI corpus are used in this thesis. It has a sample frequency of 16 kHz with 16 bit resolution. The recordings used in the experiments have been down-sampled to 8 kHz.

3.3.2 Recording Setup

The AMI corpus was recorded using a wide range of devices, including close-talking and far-field microphones, individual and room-view video cameras, projections, a whiteboard and individual pens. It was collected from meeting rooms constructed at the University of Edinburgh (UK), Idiap (Switzerland) and the TNO Human Factors Research Institute (Netherlands). The three meeting rooms feature the same types of equipment with only minor differences in the ways they were configured. Among the recorded 171 meetings, 138 of them are remote control scenario meetings, 26 are natural meetings and 7 have their own fictitious scenario (not natural meetings). There are four to five participants in each meeting.

[5]http://groups.inf.ed.ac.uk/ami/corpus/

3.3.3 Annotation

The AMI corpus provides a high quality, manually produced orthographic transcription for each individual speaker. The word-level timings were derived from the a speech recognizer in forced alignment mode, other available annotations are dialogue acts, named entities, topic segmentation, emotional state, hand gesture, extractive and abstractive summaries, etc.

The meeting IDs are formed with 7 letters taking the form [IETB][SNB][1-5][0-9][0-9][0-9][a-z]. The first letter denotes the recording location: I for Idiap, E for Edinburgh, T for TNO and B for Brno (Brno University of Technology, Czech). The second one is the meeting type, S for scenario-based meetings, N for naturally occurring meetings and B for fictitious scenario-based meetings. The following four digits are series assigned to meetings in different locations, the series for meetings in Idiap, Edinburgh, TNO and ISSCO are 1000, 2000, 3000 and 4000 series, respectively. ISSCO stands for fictitious scenario-based meetings recorded in IDIAP. The last letter is an optional postfix.

Table 3.6: *Statistics for each gender/L1 subset of the AMI-IHM corpus. Categories for native language (L1): E – English, D – Dutch, O – other.*

Gender/L1		#Speaker	#Wav	Duration
L1	E	85	301	41.57h
	D	40	160	22.20h
	O	67	221	31.60h
Gender	M	131	466	67.70h
	F	61	216	27.67h

Most of the participant IDs consist of 3 characters in the form of [MF][IET][EDO] followed by 3 numbers which were chosen to make a unique identifier for each speaker. The three characters stand for gender (male or female), recording location (I: Idiap, E: Edinburgh, T: TNO) and speaker's native language (E: English, D: Dutch, O: other). A limited number

of participant IDs for the TNO recordings have 2-3 extra letters at the end indicating the participant's role in meetings. Specifically, PM is short for Project Manager, ID is for Industrial Designer, ME is Marketing Expert and UID is Interface Designer. Table 3.6 presents the statistics for each gender or native language subset of the AMI-IHM corpus.

3.3.4 AMI-IHM

Each wav file in IHM is recorded with an individual headphone. Hence, there will be n recordings for the same meeting if there are n participants in this meeting, n varies between 4 and 5. The corpus also provides a suggested division of the corpus into the training, development and evaluation set. An annotation file is available for each subset, where a wav file is split into a few segments with the start time, end time and word-level alignment. There are 5 wav files of the AMI-IHM corpus not included in the annotation files for train, dev or eval set, so the overall duration is slightly less than 100 hours. The number of speakers, wav files, segments and duration of each subset are presented in Table 3.7. The 78 hours training set from the AMI-IHM corpus was used in this thesis.

Table 3.7: *Statistics for train/dev/eval subset of the AMI-IHM corpus.*

Subset	#Meeting	#Speaker	#Wav	#Segment	Duration
Train	137	155	547	108,221	77.89h
Dev	18	21	72	13,059	8.87h
Eval	16	16	63	12,612	8.61h
Total	171	192	682	133,892	95.37h

3.4 PF-STAR Corpus

The complete PF-STAR corpus contains more than 60 hours of native and non-native speech from children aged between 4 and 15 years, including read and spontaneous native language speech in British English, German and Swedish and non-native read English from German, Italian and Swedish children. The recordings took place in the context of the PF-STAR project (Batliner et al., 2005; M. Russell, 2003; M. Russell, 2004) that involved four institutes including Kungliga Tekniska Högskolan (KTH), Sweden; the University of Erlangen (UERLN), Germany; ITC-irst (ITC)[6], Italy and the University of Birmingham (UoB), UK. The German part of the PF-STAR corpus was collected in Erlangen and consists of native German recordings and non-native English recordings. In this thesis, German children's English recordings (PF-STAR De_En) of the PF-STAR corpus was used and the details will be presented below.

3.4.1 Recording Setup

The PF-STAR De_En corpus contains about 3.4 hours read speech recordings collected from 57 German children from a grammar school (Ohm-Gymnasium) and a general-education secondary school (Montessori-Schule Erlangen) (Hacker, 2009). It was recorded with a head-mounted microphone and sampled at 44.1 kHz, with 16 bit quantisation, via the sound card of the data collection computer. A 16 kHz downsampled copy was also available. In this thesis, the recordings have been downsampled from 16kHz to 8kHz for all the experiments. The data collection took place in a classroom, where only the child and an instructor were present.

[6]ITC-irst is the abbreviation of the Institute for Scientific and Technological Research, set up within the Trentino Institute of Culture (ITC). It is restructured with the name of Bruno Kessler Foundation (FBK) now.

3.4.2 Recording Materials

The ITC-irst English texts were chosen by ITC-irst in consultation with Italian teachers of English. The texts include isolated words (ITC-W), "phonetically rich" sentences (ITC-S) and generic sentences (ITC-G). Originally it was intended that the ITC-irst English texts should be used by all the partners to record non-native read English speech for the PF-STAR project. However, most children participated in recording in Erlangen had only been learning English for half a year and thus had relatively poor English skills. Therefore, only those words from ITC-W which were already known to the children (220 words) were used and supplemented with known texts from the text book used in the respective school. For all of the data, the reference text (which should be read) is available as well as a manual transliteration. The manual transliteration is almost the same as the reference text except the punctuation marks. Each prompt used in the actual recordings consists of a few sub-prompts, which can be a word, a phrase or a short sentence. There are 49 prompts and 1132 sub-prompts in total. The utterances in the PF-STAR De_En corpus were recorded at prompt level and the length of an utterance is between 12 to 426 words, with an average of 81 words. The overall number of tokens is 18572 and the vocabulary size is 950.

3.4.3 Speaker Information

The PF-STAR De_En corpus comprises of three subsets: MONT-subset, OHM-subset and OHMPLUS. The MONT-subset include recordings collected from 25 children (8 male and 17 female) in the Montessori-Schule Erlangen school who were in grade 5-6 and were 10- to 13-years old. The OHM-subset is collected from the Ohm-Gymnasium school, involving 28 pupils (15 male and 13 female) who were in grade 5 with age ranging from 10 to 11. The OHMPLUS is a superset of OHM includes additional data from four older children (1 of them is female) in grade 6-7 aged 12 to 14 years old. The number of speakers, number of recordings and recording duration for each age range is presented in Table 3.8.

Table 3.8: *Statistics of the PF_STAR De_En corpus.*

Age	# Speakers	# Utterance	Duration
10	15 (8 Male, 7 Female)	50	0.68h
11	26 (11 Male, 15 Female)	78	1.30h
12	13 (6 Male, 7 Female)	79	1.11h
13	2 (1 Male, 1 Female)	14	0.19h
14	1 (1 Female)	8	0.10h
Total	57 (26 Male, 21 Female)	229	3.38h

3.4.4 Annotation

Pronunciation of material in the MONT-subset was assessed by a German student (S) of English (graduate level) on the word level, sentence level and speaker level. On the word level, the labeller marked words with a strong deviation, insertions of English and German words, substitutions of words with English or similar German words as well as mispronounced words by adding the phonetic transcription. All sentences were graded with school grades from 1 (best) to 5 (worse pronunciation). On the second run, the labeller measured the overall pronunciation per speaker on a scaling from 1 (best) to 5 (worst). The annotation for the OHM-subset was made by the student expert S, 12 teachers of English $T_1 - T_{12}$ and a native teacher N in a similar way as for the MONT-subset. Five teachers rated the data again half a year later making it possible to measure not only the inter-rater agreement but also the intra-rater agreement. The additional data in OHMPLUS subset was recorded during an English reading test, the annotation was done by 14 experts, S, $T_1 - T_{12}$ and N.

3.5 WSJCAM0

The WSJCAM0 corpus (Robinson et al., 1995) is a British English speech corpus recorded by Cambridge University. The corpus is intended to be the British English equivalent of the relevant parts of the North American English WSJ0 database (Paul and Baker, 1992). Recordings were made in an acoustically isolated room in the Engineering Department at Cambridge University. They were recorded using a head-mounted microphone and a desk microphone which was positioned about 1/2 meter from the speaker's head. The sample rate is 16 kHz with 16 bit resolution.

The corpus consists of 140 speakers and it is partitioned into 92 training speakers, 20 development speakers and two sets of 14 evaluation speakers. The minimum age of the speakers is 18 and the majority of them are between 18 and 28. Table 3.9 details the sex and age range distribution of speakers in the training set, the distribution in the development and evaluation test sets are similar.

Table 3.9: *Age range distribution of training speakers.*

Range	18-23	24-28	29-40	> 40
Female	21	11	3	4
Male	25	19	4	5

All recorded sentences were taken from the Wall Street Journal (WSJ) text corpus which contains more than 16,000 sentences selected from the 1987-89 editions of the US business newspaper. The recording text material consists of adaptation, training and testing components. The same set of 18 adaptation sentences was recorded by each speaker. Each training speaker read 90 training sentences taken from the WSJ0 training subcorpus and each of the 48 test speakers read 80 testing sentences from the development testing subcorpus in WSJ0.

All of the 18.93 hours recordings, including the training, development and evaluation sets, from the close talking microphone were used in the official baseline speech recognition system for the CALL Shared Task.

Chapter 4

ASR for German-speaking Children's English Speech

4.1 Introduction

This chapter describes the ASR systems developed with the CALL-ST corpus to explore the speech recognition accuracy for non-native children's English speech using limited amounts of training data. As described in Section 3.1, the CALL-ST corpus was released for the three Spoken CALL Shared Tasks (Baur et al., 2017; Baur et al., 2018; Baur et al., 2019). Two training sets, ST1-TRAIN and ST2-TRAIN, were released for the 2017 and 2018 edition of the shared tasks (ST1 and ST2), respectively. Three test sets were released for the shared tasks, one for each task, namely ST1-TST, ST2-TST and ST3-TST. The work in this chapter was conducted under the three CALL-ST challenges, with in-domain data and out-of-the-domain data that were available during the time of the CALL-ST challenges. With the release of the TLT-school corpus, which contains English speech from Italian-speaking children as introduced in Section 3.2, the work is extended after the CALL-ST challenges. This will be introduced in Section 7.2, as it involves the TLT-school corpus and more advanced techniques

that will be introduced in Chapter 6.

The CALL Shared Tasks require recognition for spontaneous speech from German-speaking Swiss children. Both of these factors, non-native and child speech, make the tasks challenging. Although the CALL-ST corpus was provided for speech recognition, it seemed unlikely that this amount of task-specific data would be sufficient, especially for ST1. The officially provided baseline ASR system, which will be described in Section 4.2, uses WSJCAM0 as out-of-domain data to address the lack of data problem. Although WSJCAM0, described in Section 3.5, is a widely used corpus for large scale vocabulary speech recognition purpose, it mismatches with our target data in many aspects: a) it contains read speech rather than spontaneous speech, b) it is adult speech not child speech, c) it's native English not non-native speech. Instead of using WSJCAM0, a few corpora have been explored to find out which one is more relevant to the task. It is worth mentioning that the target children in the CALL-ST task are between 12 and 15 years old. They can be considered as teenagers, hence their speech will be different from young kids. There is no such corpus that matches with our target data exactly, various corpora match in part. The AMI corpus, previously described in Section 3.3, is adult speech though, it matches the target as it is spontaneous English speech from mostly non-native speakers. The English recordings from PF-STAR German, covered in Section 3.4, matches the target data in terms that it is German children speaking English, but they are not Swiss German children and it's read speech not spontaneous speech. These two corpora both have matches and mismatches with our target data, they have been used in this thesis and we have explored what percentage of these corpora to be added and what particular combination of these corpora gives the best performance for the speech recognition part of the tasks.

Section 4.4 will present the ASR systems built for the 2017 CALL Shared Task. The intuitions for data selection, model adaptation, feature selection and adaptation are described in Section 4.4.1 to Section 4.4.4. Experimental results are discussed in Section 4.4.5. Section 4.4.6 describes our submission system for ST1 which is also the best model in ST1.

Section 4.5 will describe the ASR systems developed for the 2018 Spoken CALL Shared Task (ST2). An extra 6 hours of CALL-ST training data were released in ST2, but the in-domain data is still relatively small to build a robust ASR system. Data selection and feature selection were not explored again in ST2. Instead, the experimental conclusions from ST1 were utilised and the best model for ST1 was updated with more training data (described in Section 4.5.1). Considering the importance of alignments to a good ASR model, Section 4.5.2 explores training models with alignments obtained in various ways. The structure of deep neural networks has been developed rapidly over the past years, long short-term memory (LSTM) has become a standard model structure for large scale speech recognition. Although our target data is not big, it is worth exploring whether it is beneficial to use a more advanced model structure (e.g. LSTM) on a task having limited in-domain data with compensation of a big out-of-domain data set (described in 4.5.3). The conventional DNN uses cross-entropy training criterion, while various sequence discriminative training criteria, introduced in Section 2.6, have been reported outperforming cross-entropy in many related researches. The use of sequence discriminative training techniques on the task has been explored in Section 4.5.4. Section 4.5.5 describes the models used for ST2 submissions.

No more training data released for the third edition of the Spoken CALL Shared Task, hence we focused on the text processing component of this task and did not develop more ASR systems in ST3.

4.2 Official Baseline Speech Recognition Systems

A hybrid deep neural network – hidden Markov model (DNN-HMM) built using Kaldi (Povey et al., 2011) is provided by the organizers as a baseline ASR system of the first edition of the Spoken CALL Shared Task, referred to as DNN_BASE1. The Shared Task data used to develop this baseline ASR is a super-set of ST1-TRAIN, comprising recordings of 5,500

utterances. This corpus is referred to as ST1-BASE. Thus ST1-BASE includes ST1-TRAIN plus some utterances that were not subsequently released. The baseline Kaldi ASR system is trained on about 18.93 hours of recordings from WSJCAM0 (described in Section 3.5) and 90% of ST1-BASE. The remaining 10% of ST1-BASE is used for testing.

The process for training DNN_BASE1 is presented in Figure 4.1. A speech signal frame is represented using 13-dimensional Mel-Frequency Cepstral Coefficients (MFCCs) with delta and delta-delta coefficients appended. The formed 39-dimensional MFCCs were used to train a GMM-HMM monophone model. A GMM-HMM triphone model was trained with the alignment obtained from the monophone model to produce the state-level time alignments for the neural network. A neural network with 4 hidden layers and 1024 neurons for each layer was trained with WSJCAM0 and 90% of ST1-BASE. The input to the neural network is 13-dimensional MFCCs with a context of 15 frames (i.e., 7 frames before and after). The output layer is a softmax layer, each node of this layer represents the posterior probability of the context-dependent HMM states. After training the DNN model, an adaptation is applied by fine-tuning the network using only the ST data. The language model (LM) is a bigram model trained on the reference transcription of the ST data. In cross-validation evaluations, this system achieved an average WER of 14.03%.

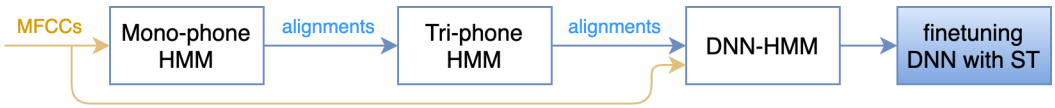


Figure 4.1: *The training process of the baseline acoustic model.*

In Shared Task 2 and 3, our best ASR developed for Shared Task 1, which will be presented in Section 4.4.6, was provided for all the participants as a baseline ASR system.

4.3 DNN Implementation in Kaldi

The DNN models in this chapter are trained using Karel Vesely's version in Kaldi[1]. It consists of an RBM pretraining step and a cross-entropy training step.

4.3.1 RBM Pretraining

The neural network is initialised with stacked restricted Boltzmann machines (RBMs) that are pretrained in a greedy layerwise fashion (Hinton et al., 2006). It is implemented according to the details in (Hinton, 2012). The training algorithm is Contrastive Divergence with 1-step of Markov Chain Monte Carlo sampling (CD-1). The first RBM has Gaussian-Bernoulli units which is trained with an initial learning rate of 0.01, and following RBMs have Bernoulli-Bernoulli units with a learning rate of 0.4.

The weights of RBM are randomly initialised with a Gaussian distribution $N(0, 0.01)$. The hidden biases of Bernoulli units and the visible biases of the Gaussian units are initialised to zero, while the visible biases of the Bernoulli units are initialised as $b_v = log(p/1 - p)$, where p is the mean output of a Bernoulli unit from the previous layer. The L2 regularization, with a penalty factor of 0.0002, is applied to the weights.

A weight decay mechanism is applied to avoid weight explosion when training the RBM Gaussian-Bernoulli units. The variance of training data is compared with the variance of the reconstruction data in a minibatch (the size is 100), the weights are shrunk and the learning rate is temporarily reduced if the variance of reconstruction data is >2x larger.

[1] http://kaldi-asr.org/doc/dnn1.html

4.3.2 Cross-entropy Training

Each RBM is trained with all the training samples and the training is unsupervised. The RBMs are stacked layerwise to compose a Deep Belief Network (DBN). Each hidden layer uses a fully connected affine transform followed by a non-linear Sigmoid layer. The pre-trained DBN-DNN is created by adding a "softmax" output layer in which the output units correspond to the leaves of the phonetic decision tree used in the GMM-HMM system. The output layer is initialised randomly.

The DBN-DNN is then trained discriminatively using the cross-entropy criterion to predict the HMM state corresponding to the central frame of the input window in a forced alignment. This is done by mini-batch Stochastic Gradient Descent (SGD) with a mini-batch size of 256 and learning rate of 0.008. The early stopping scheme is applied to prevent over-fitting. The objective function is measured on a cross-validation (cv) set, if the loss on the cv set is not decreasing in the current iteration compared to the last iteration, the learning rate will be halved. In most of our experiments, 10% of the training set is held out as the cv set and the rest is used for training. When fine-tuning a DNN trained with mixed datasets, 100% ST data were used for training and cross-validation considering the limited size of the ST data.

4.4 Developed ASR for 2017 Spoken CALL Shared Task

In ST1, most of the developed DNN-HMM systems used a similar training pipeline as the official baseline ASR system, DNN_BASE1, except for the following differences. We used 13-dimensional MFCCs with a context of 11 frames (5 on each side of the current frame) in most experiments – the use of a slightly smaller context than the official baseline ASR was accidental and was considered to have little effect on results. In addition, some DNN-HMM experiments (see Section 4.4.3) were also performed using Mel-scaled filter-bank energies with

the same 11 frames context. The neural network with 6 hidden layers and 1024 neurons for each layer was used. Two more components, linear discriminant analysis (LDA) with further decorrelating using maximum likelihood linear transform (MLLT) and speaker adaptation, were added for some DNNs in Section 4.4.2 to generate better alignments and to produce fMLLR features for the DNN.

In all the experiments, tri-gram language models trained on the reference transcriptions of the ST training data using the SRILM toolkit (Stolcke et al., 2011) were used.

4.4.1 Data Selection

The intuition for this section, as discussed in Section 4.1, is to find a combination of out-of-domain speech corpora that works best to address the lack of in-domain data problem in this task as there are only 4.8 hours of in-domain training data. Two out-of-domain corpora were selected to add to the training set, the AMI corpus (described in Section 3.3) and the German English recordings of the PF-STAR corpus (PF-STAR De_En, described in Section 3.4), because they both match multiple recording aspects with our target data.

We opted for the AMI corpus as it contains conversational speech of native and non-native speakers. The 77.3 hours training set from the IHM recordings of the AMI corpus were used in our experiments. As it is an adult speech corpus, which differs from our target data, the trained model may not learn enough from child speech if it sees too much adult speech during training, so 20%, 50% or 100% IHM have been added into the training set to find the optimum amount of IHM to improve the model. The English recordings from the PF_STAR German corpus are also included in some experiments because they are recordings of L1 German children speaking English the same as the CALL-ST corpus.

4.4.2 Model Adaptation

Linear discriminant analysis (LDA) and feature space maximum likelihood linear regression (fMLLR) were included to obtain better alignments and features for DNN training. The developed DNN training pipeline was depicted in Figure 4.2.

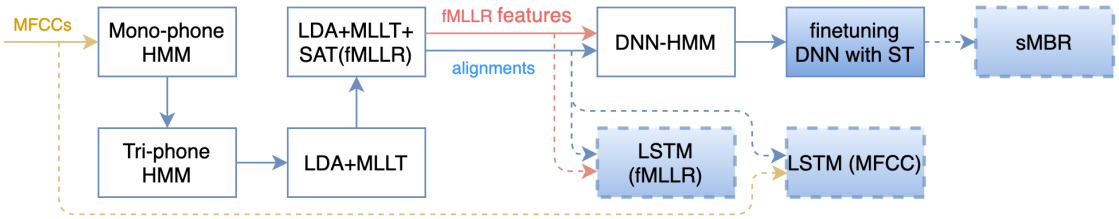


Figure 4.2: *The structure of the acoustic models for shared tasks. The filled blue components are models used in submissions. The solid components are developed in Shared Task 1 and the dashed components are added in Shared Task 2. The blue lines are alignments, the orange lines are MFCC features and the red lines are fMLLR features.*

A GMM-HMM has been trained by applying LDA on the 91-dimensional vector of MFCCs in context to decorrelate and reduce its dimension to 40-dimensional features and then further decorrelating using maximum likelihood linear transform (MLLT). The MFCCs are obtained by splicing together 7 frames (i.e., ±3 frames) and they have been normalised to have zero mean per speaker. Then speaker adaptation with fMLLR has been applied to obtain the fMLLR features and alignments. A DNN was trained with these fMLLR features and alignments on both in-domain and out-of-domain datasets and then fine-tuned with only the in-domain CALL-ST data.

4.4.3 Feature Selection

The MFCC features were obtained by applying Discrete Cosine Transform (DCT) to mel-scaled filter-bank energies (fbank) to decorrelate filter-bank coefficients. MFCCs and GMM-HMMs together used to be the standard way of doing automatic speech recognition for a

long time. With the advent of deep neural network in speech recognition, it is hard to say if MFCCs are still the right choice given that deep neural networks are less susceptible to highly correlated input and DCT is a linear transform therefore undesirable as it discards some information in speech signals. Hence, DNN-HMM experiments were also performed using 15-dimensional filter-bank features with the same 11 frames context in the same model training pipeline (see Figure 4.1).

4.4.4 Feature Adaptation

Although the DNN training does not apply normalisation on the features, the GMM-HMM used to produce the decision tree and the alignments for the DNN involves feature normalisation and adaptation. Specifically, cepstral Mean Normalisation (CMN) is applied in each GMM-HMM model training and fMLLR is applied in the developed training process. In Kaldi, each utterance is associated with a speaker label and consequently CMN and fMLLR are performed per-speaker. However, the speaker label information is not available in ST1-TRAIN. As such, we initially used a single speaker-id for all utterances in ST1-TRAIN, which resulted in using globally calculated statistics for CMN and fMLLR. In later experiments, we explored the application of CMN and fMLLR per-utterance basis, i.e., each utterance in ST1-TRAIN was considered to be from a different speaker. This was implemented in Kaldi by making the speaker-ids identical to the utterance-ids.

4.4.5 Results and Discussion

This section presents the results obtained from various systems developed for ST1. In the development experiments, the released ST training set (ST1-TRAIN) was split into two subsets at a ratio of 9:1, namely ST1_train and ST1_dev. Only the ST1_train set is involved in training, the ST1_dev set is used for evaluating the system performance during

development. The results reported in this section are evaluated on ST1_dev.

The results in Table 4.1 and Table 4.2 are slightly different from the results reported in (M. Qian et al., 2017) because the experiments were re-run after the challenge. As mentioned in Section 4.3, a subset of the training set is held-out as a cross-validation set to control the learning rate for DNN training. In the development experiments for the ST1 challenge, the last 10% of the sorted training set utterance list is used as the held-out set. Since there are much more AMI data than the CALL-ST data and PF-STAR data, and the utterance list is sorted, the resulted held-out set is undesirable: (a) the held-out set contains no CALL-ST data, (b) when we include the PF-STAR data into the training set, all of the utterances go to the held-out set, none PF-STAR utterances are actually involved in training. To avoid these, the training set of each corpus was split in advance at a ratio of 9:1. When the DNN was being trained, 90% of each corpus were combined to form the training set and 10% of each corpus were combined as the cross-validation set for training. By doing this, both training and held-out sets have a balanced mixture of datasets. The difference in Kaldi version used in the old and new experiments could also contribute to some differences in results. The conclusions do not change in the new experiments.

Feature Adaptation and Selection

Section 4.4.4 discussed the difference in using global statistics or utterance-based statistics for feature adaptation. Table 4.1 presents the experimental results on ST1_dev. According to the results, using utterance-based statistics is around 30% to 40% relatively better than using globally calculated statistics.

Three features are compared as the input to the DNN, including 13-dimensional MFCCs, 15-dimensional FBANKs or 40-dimensional fMLLR features, all spliced in the context of 7 frames. The alignments for the DNNs trained with MFCCs or FBANKs are obtained from a tri-phone GMM-HMM model, and the alignments for the DNN trained with fMLLRs

Table 4.1: *Results obtained from DNN models trained with various features using globally trained statistics or utterance-based statistics, evaluated on ST1_ dev.*

WER (%)	global stats	utt-based stats
DNN(fbank)	24.56	14.58
DNN(mfcc)	19.82	13.79
DNN(fmllr)	16.73	10.45

are from the GMM-HMM model after LDA+MLLT+SAT. The results in Table 4.1 show that fMLLR features result in the best performance among the three features. When using utterance-based statistics, the DNN model trained with fMLLR features outperform the one trained with MFCCs, which are the features used in the official baseline system, by about 4% absolute in WER. This is mainly due to the speaker adaptation that fMLLR involves, the better alignment also contributes to the improvements.

Data Selection

As discussed, three corpora were used and different combinations of the datasets were explored. Results of models trained with different training data are presented in Table 4.2. $D_1 - D_8$ are different combinations of training datasets, e.g. D_1 contains only the ST1_train set, D_8 contains the ST1_train set, the PF-STAR German corpus and 100% of IHM. The DNN models are trained with fMLLR features and the setup is using utterance-based statistics for CMN and fMLLR. 90% of D_i is used to train the DNN models and the rest of D_i is used to control the learning rate. We randomly selected 90% from each corpus if there are multiple datasets in D_i. The output layer of the DNN models trained with $D_i(i = 2, 3, ...8)$ is removed and a randomly initialised layer with the same senone set is added, the whole network is then fine-tuned with only ST1_train. The fine-tuned model is DNN.reTr. The performance is evaluated on ST1_dev.

Table 4.2: *Performance of models trained with different training data on ST1_dev using utterance-based statistics for CMN and fMLLR.*

		D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
Data	ST1_train	✓	✓	✓	✓	✓	✓	✓	✓
	+ PF_STAR	-	-	-	-	✓	✓	✓	✓
	+ IHM (%)	0	20	50	100	0	20	50	100
%WER	DNN(fmllr)	10.45	11.13	12.93	14.29	10.59	10.45	12.53	14.29
	DNN.reTr (fmllr)	-	9.41	9.80	9.98	-	8.98	9.69	9.84

It can be seen from the results that adding data to the training data can improve the performance of the DNN model. Adding 20% of IHM in the training set results in a greater advantage than including all of the IHM data. The fine-tuned DNN with D_2 obtained a WER of 9.41%, which is around 10% relatively better than the DNN model trained with only ST1_train. Including PF-STAR German can further improve the model, the DNN trained with D_6 and fine-tuned with ST_train is 8.98% in WER, which is around 14% relatively better than DNN trained with D_1. These conclusions also extend to the models trained with MFCC features or FBANK features.

4.4.6 Models for Submissions

The final model used for the submissions for the 2017 Spoken CALL challenge was chosen based on the cross-validation experiments. It was built following the procedure in Figure 4.2 and was trained with fMLLR features of 20% of AMI-IHM, PF-STAR German plus all of the ST1-TRAIN data (instead of using only 90% as in cross-validation experiments presented in previous sections) and then fine-tuned with only ST1-TRAIN data, referred to as ST1-DNN. Each utterance of ST1-TRAIN was assigned with an individual speaker-id, so that CMN and fMLLR were performed per utterance. The language model was a tri-gram language model

trained with the transcriptions of ST1-TRAIN. The values for language model weight (lmwt), acoustic model weight (acwt) and insertion penalty (p) parameters were chosen based on the best performance in the cross-validation experiments. This model has obtained a WER of 15.59% on ST1-TST.

4.5 Developed ASR for 2018 Spoken CALL Shared Task

4.5.1 Baseline System

Data

Our best ASR system from ST1 challenge, ST1-DNN (presented in Section 4.4.6), was provided for all the participants in the 2018 Spoken CALL Shared Task (ST2) as the baseline ASR system. A training set of 6 hours recordings (ST2-TRAIN) and a test set (ST2-TST) were distributed for ST2.

Table 4.3: *Statistics of ST1, ST2 data and the subsets used in ST2 experiments.*

Dataset	#Utterance	Duration
ST1-TRAIN	5222	4.80
ST1-TST	996	0.89
ST2-TRAIN	6698	5.99
ST2-TST	1000	0.91
ST12_train	10521	9.50
ST12_dev	1948	1.86

In development, the ST2-TRAIN set, after excluding sentences containing only silence, was split into 10 sub-sets and these were used to evaluate the ST1-DNN model. The two sets

with the best and the worst WER together with ST1-TST were used as the development set (ST12_dev). The remaining sub-sets of ST2-TRAIN and all of ST1-TRAIN formed the training set (ST12_train). In the end, there were 10521 utterances in ST12_train and 1948 utterances in ST12_dev as shown in Table 4.3. Both ST12_train and ST12_dev, referred to as ST12all, were included in the training set when developing the models for submissions.

Acoustic Model

We updated the ST1-DNN model with more CALL-ST data, the same training process as depicted in Figure 4.2 was used. The input features to the DNN are fMLLRs in a context of 7 frames. Section 4.4.5 discussed which setup is better in feature adaptation, global statistics or utterance-based statistics. The speaker-ids are available for ST2-TRAIN, so we used the actual speaker-ids for ST2-TRAIN and used the utterance-id as speaker-id for ST1-TRAIN in ST2 development experiments. The PF-STAR German corpus was excluded from the training set as it is relatively small compared to the bigger in-domain ST12_train set in ST2. In ST1-DNN training, we included 20% of AMI-IHM training set (IHM20) because it was more helpful than including IHM50 or IHM100 as the results shown in Table 4.2. However, it is hard to say whether including IHM20 or IHM50 will be more beneficial when we have more ST data in the training set.

Language Model

The language models used in our experiments were trigram models trained on the transcriptions of the CALL-ST data. The LM1 denotes the model obtained based on the reference transcriptions of ST12_train and used during the ASR development. The LM2 was trained on both ST12_train and ST12_dev, and was used for the final evaluation on ST2-TST.

Results

Results of the baseline DNN-HMM models on ST12_dev and ST2-TST are presented in Table 4.4. The DEV models are trained with ST12_train and the FINAL models are trained with both ST12_train and ST12_dev (ST12all) developed for final submissions. Varying the amount of AMI-IHM data to augment the ST data has a small effect on the recognition performance.

Table 4.4: *Recognition results (%WER) obtained by DNN-HMM system on the development and final test set, when using different amounts of AMI-IHM data and language model (M. Qian et al., 2018b).*

ASR model: DNN-HMM			Test data	
Data Aug	Train	LM	ST12_dev	ST2-TST
IHM20	DEV	LM1	12.68	9.62
IHM50	DEV	LM1	12.64	9.78
IHM20	FINAL	LM2	-	9.84
IHM50	FINAL	LM2	-	10.01

4.5.2 Converting Alignments

Alignment is important in training a deep neural network, it provides labels for each input feature. A better alignment usually can leverage a more accurate model. In Section 4.5.1, the baseline DNN model is trained with mixed datasets (DNN-mix) using the alignments from a GMM model which is also trained with mixed datasets (GMM-mix). It might be possible to obtain better alignments for each individual dataset from separate models rather than from a single model. Therefore, a DNN model trained with only AMI data was used to generate alignments for AMI (ali-AMI) and a DNN model trained with only ST12_train data was used to generate alignments for ST12_train (ali-ST). In order to have the same number of

senones and the same ids for each senone, both ali-AMI and ali-ST were converted with the same decision tree which was from GMM-mix. The converted alignments of AMI and ST were used to train a DNN (DNN-new) and then a new alignment generated by re-aligning ST12_train with DNN-new was used to fine-tune DNN-new. This new fine-tuned DNN obtained a negligible improvement compared to the baseline DNN, hence this approach was not applied in the final submission for ST2.

4.5.3 Long Short-Term Memory

Long short-term memory (LSTM) has often been shown to perform better than DNNs in large vocabulary speech recognition (Sak et al., 2014; X. Li and X. Wu, 2015) when there are sufficient data for training. In the CALL STs, the amount of training data is limited, but it is interesting to explore whether LSTM will outperform conventional DNN with augmenting out-of-domain AMI-IHM data into the training set. Our LSTM networks were trained based on the alignments obtained from our best DNN-HMM system. 13-dimensional MFCCs and 40-dimensional fMLLR features, both with the context of 5 frames (i.e. ± 2 frames), were compared, see Figure 4.2. To make use of the information from the future frame, the output HMM state labels were delayed by 5 frames. The LSTM network has 1024 memory cells, two fully connected hidden layer each with 1024 units, a recurrent projection layer with 256 units and a non-recurrent projection layer with 256 units.

Results, presented in the first two sections of Table 4.5, show that LSTMs trained on IHM20 perform better than IHM50 for the development model on ST12_dev and also the final model on ST2-TST. MFCCs and fMLLR features were used as the front-end features for LSTM. Although fMLLRs outperform MFCCs in DNN-HMM system, inconclusive results were observed in LSTM models – fMLLRs were slightly better on ST12_dev but then slightly worse on ST2-TST. The results obtained with LSTMs are close to that obtained with DNNs, showing that it's difficult to take advantage of LSTMs when there is limited in-domain data

even when augmented with out-of-domain data.

Table 4.5: *Recognition results (%WER) obtained by various systems on the development and final test set, when using different amounts of AMI-IHM data and language model.*

ASR model				Test data	
AM	Data Aug	Train	LM	ST12_dev	ST2-TST
LSTM (MFCC)	IHM20	DEV	LM1	12.79	9.99
	IHM50	DEV	LM1	12.82	8.65
	IHM20	FINAL	LM2	-	8.82
	IHM50	FINAL	LM2	-	9.71
LSTM (fMLLR)	IHM20	DEV	LM1	12.11	10.21
	IHM50	DEV	LM1	13.11	9.76
	IHM20	FINAL	LM2	-	9.60
	IHM50	FINAL	LM2	-	10.15
sMBR	IHM20	DEV	LM1	12.28	9.08
	IHM50	DEV	LM1	12.00	9.50
	IHM20	FINAL	LM2	-	10.56
	IHM50	FINAL	LM2	-	9.28

4.5.4 Sequence Discriminative Training

Sequence discriminative training criteria, including maximum mutual information (MMI), boosted MMI (BMMI), minimum phone error (MPE) and state-level minimum Bayes risk (sMBR), are popularly used in speech recognition. Section 2.6 has provided the details of these training criteria and the difference between cross-entropy training and sequence discriminative training. The state-level minimum Bayes risk (sMBR) was used in our experiments. The baseline DNN-HMM system was used as the base for sequence training, which

used 3 iterations and a learning rate of 0.00001. For each iteration, the alignments and word lattices were generated by decoding the ST12_train data using the corresponding cross-entropy trained DNN. The results in the bottom part of Table 4.5 shows that, during the development stage, the sMBR trained DNNs slightly outperformed the LSTMs and DNNs. The best WER in development experiments is 12.00% by the sequence training model using 50% of IHM.

4.5.5 Models for Submissions

In the development experiments, the best model was an sMBR model trained with ST12_train and 50% of AMI-IHM. Hence, the setup of this model was used to train a final model to create submissions for the challenge. The final model, referred to as ST2-sMBR, has the same setup except it was trained on ST12_train, ST12_dev and 50% of AMI-IHM. This model had a WER of 9.28% on ST2-TST, shown in Table 4.5. There was no more training data distributed for ST3, so this model was also used to generate the recognition result for ST3-TST. The WER of 9.28% on ST2-TST was obtained based on the decoding parameters estimated on ST12_dev. For decoding ST3-TST, the parameters were estimated on ST2-TST, which resulted in a WER of 8.98% on ST2-TST and 10.94% on ST3-TST.

The Spoken CALL Shared Task consists of an ASR component followed by a text processing component. Each participant was allowed to submit three entries. Although the baseline DNN model did not outperform the sMBR model in the development, it was also used to create a submission because we were interested in how ASR affects text processing in the task. The final DNN model was trained on ST12_train, ST12_dev and 50% of AMI-IHM using fMLLR features following the training processing in Figure 4.2. It obtained a WER of 10.01% on ST2-TST. The discussion of the influence of ASR on text processing in the Spoken CALL Shared Task will be presented in Section 5.7.

4.6 Summary and Conclusions

This chapter presents the ASR systems developed for German-speaking children's English speech. The work was based on the Spoken CALL Shared Tasks, where 13.6 hours of English recordings were distributed over the three years when the shared tasks were held. For ST1, AMI and PF-STAR German corpus were augmented to the training set to address the lack of data problem. The best model was obtained from a DNN-HMM model trained with both in-domain and out-of-domain data and fine-tuned with only in-domain data. It achieved a WER of 9.16% on ST1_dev and 15.59% on ST1-TST. For ST2, effort has been put to find a better alignment for neural network training and to find a better neural network structure for a limited amount of in-domain data. LSTMs do not show consistent advantages over conventional DNN-HMM models, sequence discriminative training using sMBR is capable of improving the performance over cross-entropy training. The best submitted model, ST2-sMBR, achieved a WER of 9.28% on ST2-TST. The ST2-sMBR model was also used in ST3, and it obtained a WER of 10.94% on ST3-TST.

Chapter 5

Text Processing for German-speaking Children's English Speech

5.1 Introduction

The aim of the Spoken CALL Shared Task is to label prompt-response pairs as "accept" or "reject" based on the correctness of the meaning and grammar of the spoken response. These judgements will be provided to the students to help them improve their spoken language. The language and the meaning gold standard judgements have been provided along with the data. The system consists of an ASR system which is used to transcribe the spoken responses to texts and a text processing (TP) system to make decisions based on the ASR transcriptions. Chapter 4 has covered the ASR systems developed for the shared tasks and this chapter will present the text processing systems.

The baseline text processing system will be introduced in Section 5.2. The prompts and responses in the baseline grammar and their corresponding lesson labels are analysed in Section 5.3. The developed rule-based TP system and a machine learning-based TP system

will be presented in Section 5.4 and Section 5.5. Results and discussions are provided in Section 5.6 and Section 5.7.

5.2 Baseline System

A baseline text processing system was provided by the ST challenge. It is a rule-based system using a template-based grammar (Rayner et al., 2015). The baseline reference grammar includes 565 prompt-units, each prompt-unit consists of a German text prompt and a set of possible responses to it. An example of the prompt-unit is shown in Figure 5.1. Each item in the CALL-ST corpus is provided with a German text prompt, the prompt is compared with the prompt-units in the reference grammar to obtain a list of possible valid responses. If the ASR transcription of a given utterance was in the response list, this utterance would be labelled as "accept", otherwise, it would be labelled as "reject".

```
<prompt_unit>
  <prompt>Frag: Wie viel kostet es?</prompt>
  <translated_prompt>Ask for: how much does it cost?</translated_prompt>
  <response>how much does it cost</response>
  <response>how much does this cost</response>
  <response>how much is it</response>
  <response>how much is this</response>
</prompt_unit>
```

Figure 5.1: *An example of the prompt-units in the reference grammar.*

5.3 Analysis of Lessons, Prompts and Responses

The CALL-ST corpus is comprised of 8 lessons as mentioned in Section 3.1.1. Each lesson has multiple prompts and each prompt has a list of possible responses in the reference grammar. However, there are some prompts shared in different lessons and prompts in different lessons could be similar to each other. Prompts that occur in multiple lessons are listed below:

- prompts "Hallo (hello)" and "Ja (yes)" occur in all the lessons;
- prompt "Danke (thank you)" occur in all the lessons but not in 'contact';
- "Frag: Wie viel kostet es? (Ask: how much does it cost?)" has both the label of 'airport' and 'shopping';
- "Sag: Dies ist okay", "Sag: Ich bin aus XXX (Say: I am from XXX)" and "Sag: Ich heisse XXX (Say: my name is XXX)" with different places and names have two lesson labels – 'airport' and 'contact'.

There are also a few prompts from different lessons quite similar to each other. This could be confusing for the TP system and makes the classification difficult. For example, prompt "Frag: X Tickets für XXX (Say: X tickets for XXX)" is applied in lesson 'airport', 'tourist office' and 'tube' with different places; "Frag: XXX (Say: XXX)" is used when the student is asking for something both during shopping or in a restaurant.

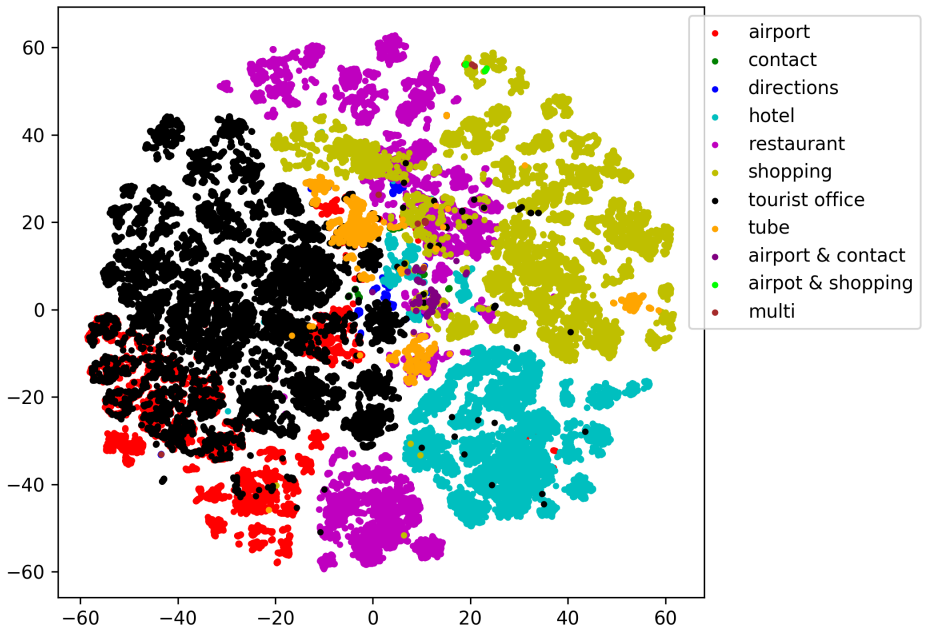


Figure 5.2: *2-dimensional t-SNE visualisations of responses in the reference grammar using a Doc2Vec model trained with ST data.*

How different are the prompts and corresponding responses in different lessons? Are

these factors discussed above affecting responses in the vector space? To understand these questions, a visualisation of the responses has been created in the embedding space (Figure 5.2). A Doc2Vec model was used to extract the sentence embeddings for all the responses of each prompt in the reference grammar, the details of the Doc2Vec model will be introduced in Section 5.5.5. The embeddings have been clustered and projected using t-SNE, the responses in different lessons have been plotted in different colours. The x-axis and y-axis are the 1^{st} and 2^{nd} of the projected embeddings. Figure 5.2 shows clear clusters of responses from different lessons. The overlaps between different lessons might result from the fact there are similar prompts between lessons.

5.4 Developed Rule-based Text Processing System

The developed rule-based text processing system is based on the baseline rule-based system, with an extra post-processing front-end and a fusion back-end. Different versions of the rule-based system were used in the 2017 Spoken CALL Shared Task (ST1) and the 2018 Shared Task (ST2), referred to as TP1 and TP2. They both have post-processing and reference grammar but differ in details.

5.4.1 Post-processing

As the CALL-SLT tool was designed for children to practice English conversation, it seems reasonable to disregard some hesitations, repetitions and modifications of the spoken responses, which are difficult to be handled by the grammar. As such, the ASR output was post-processed as described below.

Formulaic expressions: Words like "yes", "hello", "hi", "sorry" often occur at the beginning of the sentences. These words were removed because they are not useful to make

judgments neither on grammar nor meaning.

Interjections: Some hesitation words (such as "um", "ah", "hah") may appear anywhere within the sentences. A list of hesitation words was created according to the transcriptions of the ST1-train set. Any words in this list were removed from the ASR output.

Repetitions: There are different kinds of repetitions appearing in the sentences which are caused by hesitations of person speaking. Those repetitions could be words or phrases, for example, "I have three three tickets", "No I don't have a don't have a reservation". Similarly, rephrases may occur, for example, "No I don't have a do not have a reservation". Repetitions of words and phrases can be detected and removed. A mapping list of commonly occurred rephrases were created, and were used to remove the rephrases from the ASR output.

Half-words: Another case that has been processed is false-start. Persons may be uncertain about their answer, so they may not give the correct response for the first time, but it should also be accepted if they make it correct for the second time. Half words, like "gal" in "I want tickets for the gal gallery", "a" in "I want a an orange juice", "post" in "I would like to pay by post postcard" were removed from the sentences.

Formulaic expressions, interjections and repetitions were excluded in post-processing stage of TP1. After ST1 challenge, it was noticed that more errors could be handled by post-processing. Thus, the post-processing was updated in TP2 with a more complete set of repetitions and half-words.

5.4.2 Expanded Reference Grammar

The key to the rule-based text processing system is the reference grammar, an example of which is shown in Figure 5.1. However, the baseline grammar provided by the challenge organizers is not complete. It was expanded in this thesis to be as complete as possible.

In many cases, the prompt-units in the grammar are closely related. For this reason, the baseline grammar was created based on a few prompt-templates and template-applications by Rayner et al. (2015). A prompt-template allows parameterisation of prompt units and template-applications combine a prompt-template and a list of arguments. Figure 5.3 presents examples of a prompt-template and template-applications. The prompt-template contains a list of valid responses, the "ENGLISH" in each response is the argument that will be replaced in template-applications. The list of prompt-templates, the responses in the prompt-templates and the template-applications have been expanded according to the training materials.

```
PromptTemplate where_is_london_place FRENCH GERMAN ENGLISH
Lesson         directions
Group          ask_directions_to
Text/french    Demande où se trouve FRENCH
Text/german    Frag: Wo ist GERMAN?
Text/english   Say: where is ENGLISH
Response       ( can | could ) you give me directions to ENGLISH ?please
Response       i am looking for ENGLISH ?please
Response       where is ENGLISH ?please
EndPromptTemplate
```

(a)

```
ApplyTemplate where_is_london_place "le musée britannique" "das British Museum" "the british museum"
ApplyTemplate where_is_london_place "le zoo" "der Zoo" "the zoo"
```

(b)

Figure 5.3: *(a) Example of a prompt-template, the responses in the example is not complete. (b) Examples of template-applications.*

Responses in the true transcriptions should have all been accepted if they are labelled as "correct" according to the gold standard. However, a considerable number of false rejections were found and this is due to transcriptions not being covered by the grammar. We went through all the false rejections and gradually added the correct transcriptions that have correct human gold standard judgments to the grammar. The modifications were made to the regarding prompt-template or template application. We then applied text processing to the true transcriptions with the updated grammar and went through all the false rejections and false acceptances again and updated the grammar accordingly. This procedure

was repeated a few times, at each step taking care that the responses added to the grammar did not cause a large increase of false acceptances. Afterwards, this grammar updating procedure was applied to the actual ASR output – expanding the prompt-templates and template-applications according to the false rejections of the ASR output.

Table 5.1: *Number of units and responses in different versions of reference grammars.*

Version	Num. of units	Num. of responses
Baseline	565	43,862
Grammar_ST1	561	56,425
Grammar_ST2	557	63,469

The above procedure was applied on ST1 and ST2 training set, resulting in grammars referred to as "Grammar_ST1" and "Grammar_ST2", respectively. The total number of prompt units and responses of the baseline and modified grammars are shown in Table 5.1. Note that although more prompt units were added to the baseline grammar, the "Grammar_ST1" and "Grammar_ST2" have fewer prompts than the baseline grammar due to the baseline grammar containing duplicated prompt units, which were then excluded. The total number of responses from the baseline increased by 28.6% in "Grammar_ST1" and by 44.7% in "Grammar_ST2".

5.4.3 Fusion

In developing the ASR systems, multiple acoustic models have been trained and it has been observed that the performance of models or the best parameters for a specific model varies on different test sets. As such, the weighted summation fusion approach was explored with parameters trained on the development set to take advantage of multiple systems.

The final output of the system is "accept" or "reject", it can be considered as a 2-class

classification problem. The judgments were converted into 2-class scores for fusion. Let class c_1 and c_2 represent "accept" and "reject", respectively. If the judgment for item x is "accept", then the score should be $score_{c_1}(x) = 1, score_{c_2}(x) = 0$, and if it is "reject", then the score should be $score_{c_1}(x) = 0, score_{c_2}(x) = 1$. In our experiments, the log score has been used:

$$score_c(x) \leftarrow log(score_c(x) + \epsilon). \tag{5.1}$$

where ϵ is set to 0.0001. Let there be K input systems where the ith system outputs the log score vector $score_{c,i}(X)$. Then the fused score $score_c(X)$ is given by:

$$score_c(X) = \sum_{i=1}^{K} w_{c,i} \cdot score_{c,i}(X). \tag{5.2}$$

The weight, $w_{c,i}$ can be trained on the training data. After obtaining the fused score, we could assign the class for item x by:

$$class(x) = arg \max_c score_c(x).$$

Fusion was achieved using the linear logistic regression based fusion module in the FoCal toolkit (Brümmer, 2007).

5.5 Machine Learning-based Text Processing System

5.5.1 System Structure

The baseline TP system is making decisions based on a 1-best match, the quality of the judgements highly depends on the coverage of the reference grammar. To make use of multiple responses in the reference grammar and make decisions based on more information, a machine learning-based text processing system was developed. The structure of the system is depicted in Figure 5.4.

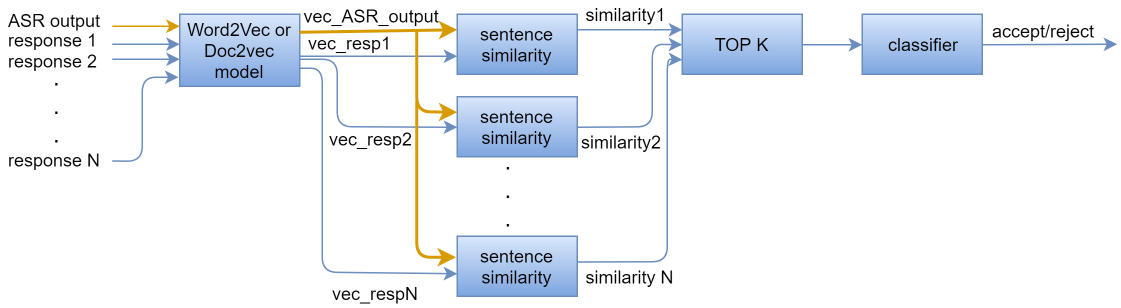


Figure 5.4: *The structure of the machine learning-based text processing system for ST.*

The ASR transcriptions and the reference responses have firstly been converted into vector representations with a Word2Vec or Doc2Vec model (Mikolov et al., 2013a; Mikolov et al., 2013b). Then these vectors are used to calculate the sentence similarities between the ASR transcription and each reference response. The similarities are taken as feature representations for the utterances given their specific prompts. As the number of reference responses is different for each prompt, the number of similarities varies among the prompts. It would be ideal to build prompt-dependent systems, each has a different dimension of input features. However, it is not feasible in practice due to the limited amount of training samples for each prompt. The top K similarities (lowest K distances) are selected as the input features to the neural network. A fixed dimension of the input features enables the training on data corresponding to different prompts. A 2-class classification can be applied to these features with any conventional classification approach, e.g., Logistic Regression, Support Vector Machine, Nearest Neighbor and Neural Networks.

5.5.2 Sentence Similarity

Sentence similarity is to determine the closeness of two pieces of texts both in lexical level and semantic level. There are a lot of popular approaches calculating sentence similarity based on different word embeddings, e.g. k-means, cosine similarity, Word Mover's Distance (WMD), Variational Auto Encoding. In this thesis, WMD is compared with dynamic programming

(DP) distance.

Word Mover's Distance measures the minimum travelling distance from the embedded words of one sentence to another one without considering the word orders (Pele and Werman, 2008; Pele and Werman, 2009; Kusner et al., 2015). While a dynamic programming distance (Bellman, 1966) takes into consideration of the word orders when calculating the sentence similarity. Each word in the ASR and reference transcriptions is represented as a word vector and the distance between each ASR and reference word vector is calculated with the Euclidean distance. DP is used to find the alignment between the ASR and reference transcriptions that minimizes the accumulated distance. In Section 5.5.3 and Section 5.5.4, sentence similarity is calculated using WMD algorithm based on the pre-trained Google-News model. A comparison between different embeddings is presented in Section 5.5.5, the comparison between WMD algorithm and various versions of DP distance is provided in Section 5.5.5.

5.5.3 Comparing between Classifiers

The classifier is the last component in the ML-based text processing system, but it has been investigated first in the experiments considering its importance to the entire system. Four classifiers have been compared on ST2-TST, namely Linear Discriminative Analysis (LDA), Logistic Regression (logReg), Support Vector Machine (SVM) and neural network (NN). The dimension of input features has been reduced to 2 using Principal Component Analysis (PCA) for logistic regression and SVM. The ASR output was from the best model obtained during the development stage in ST2, which was trained with sMBR criteria and achieved a WER of 12.00% on ST12_dev, described in Section 4.5.4. The neural network has two hidden layers and each has 16 neurons, it uses *tanh* as the activation function and the limited-memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm as the optimization function with a maximum iteration of 400. The output layer of the network has two units,

representing the probabilities of the data belong to each class.

Table 5.2: *Results obtained by machine-learning text processing systems employing different classifiers (K was set to 10) (M. Qian et al., 2018b).*

Classifier	F-measure	D	D_{full}
LDA	0.88	9.767	4.136
logReg (PCA)	0.884	10.263	4.281
SVM (PCA)	0.891	10.939	4.616
NN	0.928	12.716	7.101

Results obtained using different classifiers are presented in Table 5.2. The presented results are with K set to 10 (but they did not vary largely for different values of K). It can be seen that the NN-based system performed better than other classifiers in all evaluation measures (F-measure, D and D_{full} score).

5.5.4 Comparing Thresholds in Neural Network

Varying the threshold is a typical trick to improve the performance of a neural network classifier. The results from the neural network classifier presented in Table 5.2 were obtained using a threshold of 0.5. Various thresholds were explored in the experiments to make better judgement decisions.

Results, as a function of the threshold value, obtained using the NN-based TP system are depicted in Figure 5.5. The left y-axis represents the value of the D score as well as D_a and D_{full} scores, the y-axis on the right stands for the value for $F-$measure. The thinner lines are scores for ST12_dev and the thicker dashed lines are scores for ST2-TST.

A considerably higher D-score can be achieved on ST12_dev by varying the threshold. The best D-score is 26.087 when the threshold is set to 0.032. This however is more due to

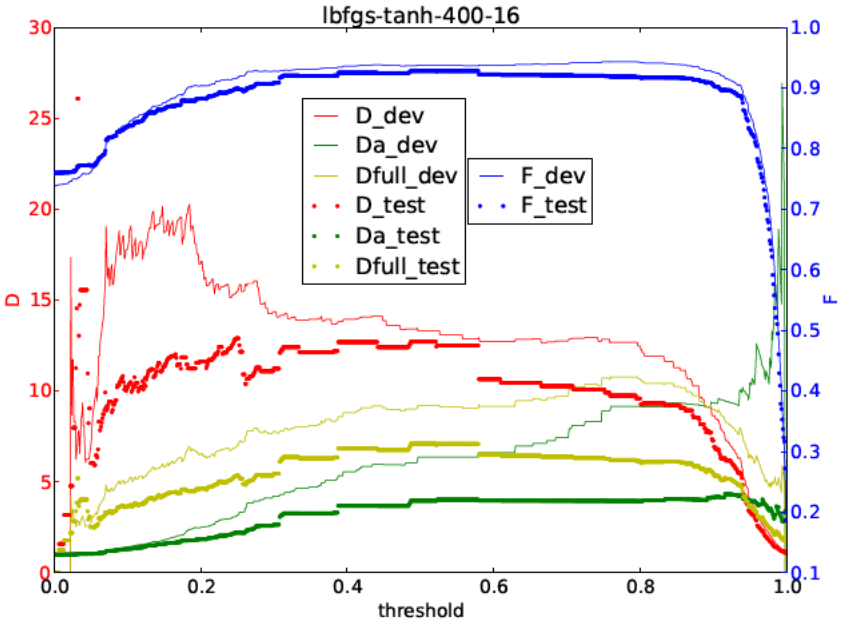


Figure 5.5: *Results in terms of various evaluation measures for the NN-based TP system when varying the decision threshold (M. Qian et al., 2018b).*

the fact that the D-score does not fully reflect the system's performance. Indeed, a closer examination shows that the system prefers to accept most of the utterances. This unhealthy state of the system is indicated by the F-measure which reduced to only 0.771. This problem is also handled by the use of the new proposed D_{full} measure. A reasonably higher D-score (12.900) can be achieved when the threshold is set to 0.250, while the D_{full} measure (5.340) is lower than 7.101. Besides the system is not stable when the threshold is around 0.25. This figure proves that the D_{full} score, which was proposed after ST2, is a better measure for the system compared to the D score. A better D_{full} score may be achieved by tuning the threshold, but it is better to limit the threshold with a range, e.g. 0.4-0.6, to obtain a healthy system.

5.5.5 Comparing Embeddings

Word embeddings (Word2Vec) and document embeddings (Doc2Vec) have been introduced in Section 2.10. The pre-trained GoogleNews Word2Vec model[1] was used in the experiments presented in the previous sections. It contains 300-dimensional word vectors for a vocabulary of 3 million words and phrases which are trained on approximately 100 billion words from the Google News dataset.

In this section, the GoogleNews model was compared with a Word2Vec model trained on the ST data. This ST Word2Vec model includes a vocabulary of 1366 words, each has a vector of 100 dimensions. It is trained using the CBOW algorithm with the responses in the reference grammar, transcriptions of ST1-TRAIN and ST2-TRAIN, and recognition transcription of ST2-TST. Furthermore, the above Word2Vec word embeddings are compared with the word embeddings obtained from a Doc2Vec model (Mikolov et al., 2013b; Mikolov et al., 2013a), which aims to create a vector representation of a document. While training a document vector for each document, a word vector is generated for each word. The word embeddings from a Doc2Vec model were explored as they might contain more information about the document as the model has seen the document vector during training. Our Doc2Vec model was trained with ST1 and ST2 training set using DBOW algorithm with 100-dimensional word vectors.

To explore the influence of different word embeddings on the system, the similarity algorithm and the number of features are fixed (Word Mover's Distance, K=10). For each vector model, neural networks with different numbers of layers and numbers of neurons per layer were trained. Apart from using 0.5 as the threshold, the threshold has also been optimized carefully (bad thresholds may result in unhealthy systems as discussed in Section 5.5.4). The threshold and the neural network structure were tuned based on ST2-TST. The D_{full} scores for ST2-TST and ST3-TST with the best threshold or with 0.5 as the

[1] https://code.google.com/archive/p/word2vec/

threshold are provided in Table 5.3.

Table 5.3: *D_{full} score for ST2-TST and ST3-TST with different vector models and different threshold.*

Model	optimize th		th=0.5	
	ST2-TST	ST3-TST	ST2-TST	ST3-TST
GoogleNews	6.073	4.925	5.562	5.236
ST_word2vec	6.894	5.221	5.697	5.461
ST_doc2vec	5.705	5.740	5.639	5.526

It seems that vector models trained with ST data outperform the GoogleNews model. This might be related to the fact that the ST data has a very small vocabulary size and its context scenarios are limited. The word embeddings for the ST vocabulary have been trained well in a small Word2Vec or Doc2Vec model, while a huge generic model trained on general external data may not represent the ST vocabulary very well. The word embeddings from the ST Doc2Vec model did not outperform that from the ST Word2Vec model on ST2-TST, when the models are optimised on ST2-TST. However, they achieved higher D_{full} score (5.740) on ST3-TST than embeddings from the ST Word2Vec model (5.221), which is not expected. Although the word embeddings from a Doc2Vec model might contain more information about the document, they may not be trained well as the model is aiming to train the document vectors.

5.5.6 Word Order in Sentence Similarities

Word Mover's Distance was used in previous experiments. Although WMD works pretty well in the experiments, it does not consider the word order when calculating the similarity. It is interesting to find out whether the word order is important in word embedding based sentence similarity calculation, especially for short sentences like the ST data. To answer this

question, WMD has been compared with a dynamic programming (DP) distance which takes into consideration of the word orders in sentence similarity calculation. The D scores for ST2-TST and ST3-TST in terms of different distance algorithms are presented in Table 5.4.

Table 5.4: *ST2 test results for ST2-TST and ST3-TST obtained by the machine learning-based system with different word embeddings and distance algorithms.*

Model	Distance	ST2-TST			ST3-TST		
		D	D_a	D_{full}	D	D_a	D_{full}
GoogleNews	wmd	8.974	4.110	6.073	7.655	3.168	4.925
	dp0	12.199	3.727	6.743	9.114	3.098	5.314
	dp1	10.367	4.096	6.516	7.768	3.173	4.965
	dp2	11.979	4.011	6.932	8.909	3.177	5.320
	dp3	11.004	3.926	6.573	8.293	3.231	5.176
ST_word2vec	wmd	10.478	4.536	6.894	8.244	3.307	5.221
	dp0	11.571	4.706	**7.379**	9.239	3.341	5.556
	dp1	11.485	4.569	7.244	9.239	3.341	5.556
	dp2	10.957	4.396	6.940	8.643	3.321	5.358
	dp3	11.691	4.231	7.033	9.082	3.336	5.504
ST_doc2vec	wmd	10.349	3.146	5.705	9.764	3.375	**5.740**

Apart from the accumulated DP distances (dp0), three other DP distances (dp1 $\sim$ dp3) obtained using different normalisation were also used. The DP distance for a long sentence may be bigger on average than that for a short sentence, as the DP distance is heavily influenced by the length of the path chosen by the algorithm, and a longer sentence usually needs a longer path to move to the endpoint on the distance lattice than a shorter one. Hence, we divided the DP distance by the length of the chosen path. This is the dp1 distance in Table 5.4. For each utterance, comparing the ASR transcription of the utterance with the reference responses results in a set of DP distances. The number of the distances

and the variance of the distances depend on the number and the variance of the responses for this prompt. The variance normalised distance (dp2) equals to dp0 divided by the standard deviation of the set of the distances for the given prompt. The dp3 distance in Table 5.4 is the DP distance with both length normalisation and variance normalisation.

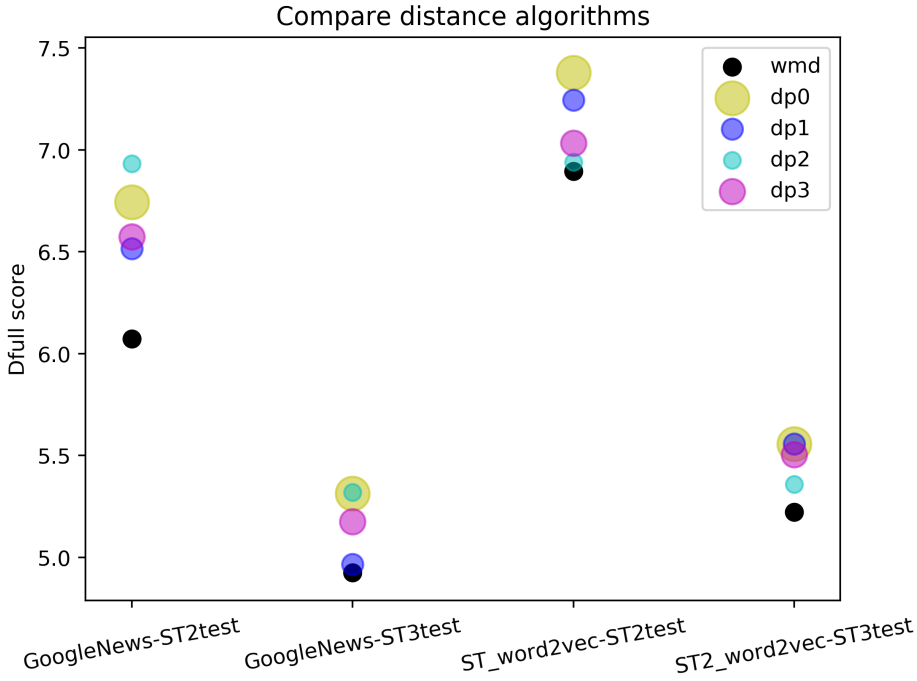


Figure 5.6: *Comparison of different distance algorithms.*

A 10-dimensional distance feature from each vector model has been input to a neural network, the hidden layers of the neural network and the threshold for the output layer have been tuned based on ST2-TST. The results in Table 5.4 are from the best network for each setup. The best D_{full} score is 7.379 for ST2-TST and 5.740 for ST3-TST, both are obtained with the DP distances extracted from the ST Word2Vec model. The comparison between the distance algorithms with different datasets is shown in Figure 5.6[2]. For different data, the best distance algorithm is always the DP distance, although it's not always the same DP distance. All different DP distances outperform WMD for both ST2-TST and ST3-TST.

[2]ST2test is the same as ST2-TST and ST3test is ST3-TST.

5.5.7 Comparing ASR transcriptions

It has been shown to be beneficial to use multiple ASR hypotheses as an n-best list or lattice in many spoken language processing tasks. In the 2018 Shared Task, the system developed by Nguyen et al. (2018) shows that using 2-best ASR hypotheses outperforms 1-best ASR hypothesis. They compute the edit-distance between each hypothesis and each sample response from the reference grammar, then use the ASR hypothesis with the smallest edit-distance as the input of the text classifier. In this thesis, the 2-best ASR transcriptions are leveraged in a different way. In the experiments discussed above, the input of the text classifier is a 10-dimensional distance feature calculated between the best ASR hypothesis and the reference responses. In this section, a 5-dimensional feature is obtained from each of the 2-best ASR hypotheses, then 10-dimensional concatenated features have been input to the text classifier.

Table 5.5: *The best D_{full} score for ST2-TST and ST3-TST obtained by using the best and two best ASR transcriptions.*

ASR	ST2-TST	ST3-TST
1-best	7.379	5.740
2-best	7.089	5.727

Similar systems have been developed with these 2-best hypotheses features as it has been done for the 1-best hypothesis features. All the results with different word embedding models and different sentence similarity algorithms show that the 2-best system is worse than 1-best system. The best D_{full} scores for ST2-TST and ST3-TST from 1-best and 2-best systems are shown in Table 5.5. This suggests that the advantage of using the n-best hypotheses, in terms of mitigating the effects of ASR errors, can also be achieved using vector representations of words and similarity calculations.

5.6 D scores for TP

5.6.1 Comparison between Different Systems

This section will compare the three text processing systems on ST1-TST, ST2-TST and ST3-TST. TP1 and TP2 are both rule-based systems with different versions of post-processing front-end and reference grammar, they have been used in ST1 and ST2, individually. TP3 and TP3* are ML-based systems and have been used in ST3. The results obtained from these TP systems are presented in Table 5.6.

Table 5.6: *$D/D_a/D_{full}$ scores for test sets from ST1, ST2 and ST3 with different text processing systems.*

	ST1-TST			ST2-TST			ST3-TST		
	D	D_a	D_{full}	D	D_a	D_{full}	D	D_a	D_{full}
TP1	4.710	2.995	3.756	9.328	3.058	5.341	9.102	3.355	5.526
TP2	-	-	-	10.714	3.116	5.778	9.259	3.360	5.577
TP3	-	-	-	-	-	-	9.239	3.341	5.556
TP3*	-	-	-	-	-	-	9.764	3.375	5.740

The post-processing of TP2 tackles a few more half-words and repetitions. TP2 also has a more complete grammar than TP1, statistics of Grammar_ST1 for TP1 and Grammar_ST2 for TP2 are provided in Table 5.1. The post-processing and reference grammar of TP2 was updated according to the observations of ST1-TST and ST2-TRAIN, so the results of TP2 on ST1-TST is not presented in Table 5.6. A more complete grammar will leverage a better D score. Results on ST2-TST and ST3-TST show that the TP2 outperformed TP1 in all system scores (D, D_a, and D_{full}). The ST1-TST and ST2-TST are involved in training and evaluation of the machine learning-based text processing system, so results of TP3 are only reported on ST3-TST. In developing the TP3 system, we made decisions about the type of embeddings and similarity measure that we used based on ST2-TST data – this led to

the use of Word2Vec embeddings trained on ST data and DP-based similarity measure. The developed TP3 then achieved a D_{full} of 7.379 on ST2-TST, which is a large improvement over TP2. However, the TP3 then did not outperform TP2 on ST3-TST data. Consequently, we found that the best setup for the TP3 system for ST3-TST is to use Doc2Vec embeddings trained on ST data and WMD similarity measure. Results of this (TP3*) system are shown in the last line in Table 5.6 – it achieved D_{full} of 5.740 on ST3-TST and outperformed the rule-based TP systems.

5.6.2 Influence of Fusion

The technique of fusion, details are described in Section 5.4.3, is applied in Shared Task 1 and Shared Task 2. The results from a fused system and the best single system in ST1 and ST2 are presented in Table 5.7.

Table 5.7: *Performance of the best single system and a fused system in ST1 and ST2.*

	System	Evaluation Measure			
		F-measure	D	D_a	D_{full}
ST1	Single (JJJ)	0.859	4.710	2.995	3.756
	Fused (KKK)	0.862	4.766	3.236	3.927
ST2	Single (DDD)	0.915	10.714	3.116	5.778
	Fused (FFF)	0.914	10.764	3.009	5.691

In ST1, the outputs from six separated ASR systems were fused, four of them were from the best ASR model (presented in Section 4.4.6) with different decoding parameters, the other two were from the provided Kaldi baseline ASR and the Nuance ASR. This fused result, submission KKK in the challenge[3], achieved a D score of 4.766, which was the highest D score in ST1 challenge. It outperformed the second best submission (JJJ) in the challenge,

[3]https://regulus.unige.ch/staging_spokencallsharedtask/

which was our entry applying the same text processing to our single best ASR output without back-end fusion, by 1.2% relatively in D score.

In ST2, there were also outputs from 6 variants of ASR systems being fused together, including DNN-HMM, sequence training model and LSTM model each with 20% or 50% of IHM data. The fused result, submission FFF in the challenge[4], outperformed our submission with single best ASR output by 0.5% relatively in D score. However, "FFF" did not outperform "DDD" in D_{full} score. As the fusion only resulted in minor improvements in D score and no improvement in D_{full} score in ST2, it was not applied in ST3.

5.7 Analysis of the Effect of ASR Errors on the System Performance

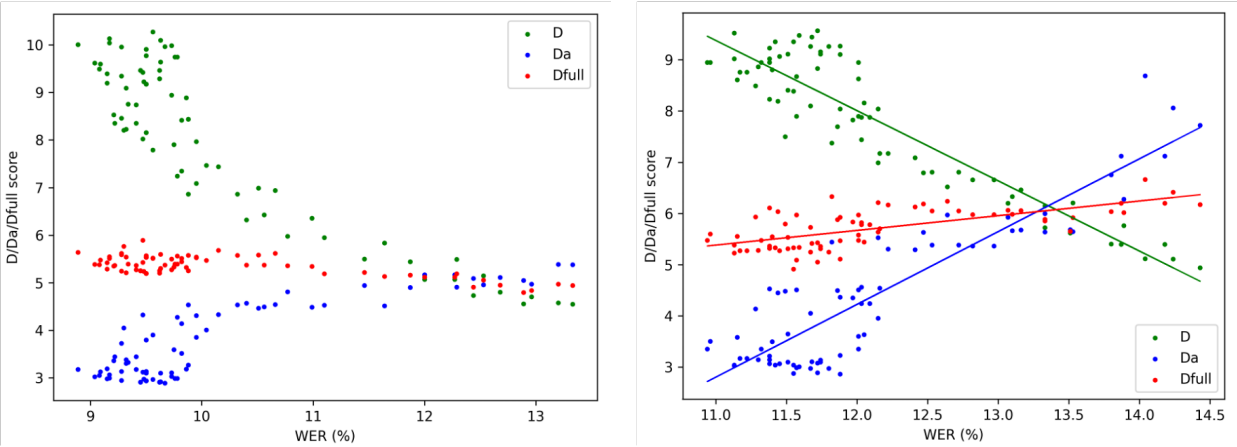
The target of the system is to obtain a better D/D_{full} score, while the criteria of the first component of the system (ASR) is the word-error-rate (WER). In general, if the recognizer is as precise as possible, it could produce recognition results that are close to the true transcription which has a very high D/D_{full} score. However, does an ASR output with a lower WER always result in a higher D/D_{full} score?

5.7.1 D scores for various outputs from the same ASR system

To explore the above question, a list of recognized results were firstly obtained from model ST2-sMBR for the test sets by varying the acoustic scaling factor and the word insertion penalty in decoding. Although these results were from the same model, the word-error-rate varied a lot because of the difference in decoding parameters. Then, the same text processing, TP2 without fusion (presented in Section 5.4 and 5.6), was applied to these different ASR

[4]https://regulus.unige.ch/spokencallsharedtask_2ndedition/

outputs to get the corresponding D scores. The $D/D_a/D_{full}$ scores and word-error-rate for ST2-TST and ST3-TST are plotted in Figure 5.7.



(a) *ST2-TST* (b) *ST3-TST*

Figure 5.7: *D-WER for ST2-TST and ST3-TST with the ST2-sMBR model.*

It is obvious that a better WER corresponds to a better D score from a general point of view, but it is not always the case within each small range of WER, e.g. the relationship between D and WER is more complicated when it comes to $[best\_WER, best\_WER+0.5]$. For this reason, the decoding parameters for the test set were selected based on the D scores of the development set rather than the word-error-rate at ST2 development stage.

However, D_a seems to correlate positively with WER for ST3-TST in Figure 5.7b - an increase in WER can produce an improvement in D_a rather than a degradation. In theory, an ASR with a higher WER tends to produce more rejects – mostly FRs but also CRs, this means more FRs and fewer PFAs/GFAs. As the D_a score is more sensitive to FAs than FRs, an ASR with a higher WER potentially can have a higher D_a score. In real systems, it's also possible that more ASR errors lead to fewer PFAs and GFAs. Table 5.8 shows some examples from two versions of ASR outputs, one with a WER of 10.94% and the other 14.43%. All the examples should be rejected according to the true transcription. The outputs from ASR1 contain fewer errors, but the incorrect grammar or meaning have been corrected by the ASR, resulting in false accepts. While ASR2 makes more errors, producing an incorrect response

that are correctly rejected by the system. Therefore, the D_a may correlate positively with WER, although it's not always the case.

The newly proposed D_{full} metric still positively correlates with WER in Figure 5.7b, but the slope is smaller compared to the D_a score. D_{full} is more stable when the word-error-rate varies according to the decoding parameters.

Table 5.8: *Examples of more ASR errors leading to fewer decision errors. I: insertion error, S: substitution error, PFA: plain false accept, GFA: gross false accept, CR: correct reject.*

Example	System	Text	Error	Decision
1	True transcription	can i have room for three nights	-	
	ASR1 (WER=10.94%)	can i have a room for three nights	I	PFA
	ASR2 (WER=14.43%)	can i have a room for three night	I+S	CR
2	True transcription	i have a older brother	-	
	ASR1 (WER=10.94%)	i have one older brother	S	PFA
	ASR2 (WER=14.43%)	i have a older brother blue	I	CR
3	True transcription	where is the nationality museum	-	
	ASR1 (WER=10.94%)	where is the natural history museum	S	GFA
	ASR2 (WER=14.43%)	where is the natural a ti museum	S + 2 I	CR

5.7.2 D scores for outputs from different ASR systems

Apart from the outputs from the same recognizer, the relationship between $D/D_a/D_{full}$ scores and the word-error-rate for outputs from different speech recognisers was also explored. Results of applying TP2 (without fusion) to six ASR outputs, each from a different recognizer, are plotted in Figure 5.8. The entry with the highest D_{full} score ("■" in Figure 5.8) is the true transcription. The second best entry (scattered using "♦") is applying our rule-based text processing (TP2 without fusion) to the output from the ASR developed by the group of Fondazione Bruno Kessler (Gretter et al., 2019) which has obtained the best

speech recognition result for ST3 (referred to as FBK_ASR). The other four entries ("•") are from our four ASRs (DNN-HMM and sMBR trained with ST12_train, ST12_dev and 20% or 50% of IHM data).

The FBK_ASR has a WER of 8.96% on ST3-TST, while our best ASR – ST2-sMBR (sequence discriminative training with sMBR using 50% IHM) has a WER of 10.94% on ST3-TST. When the same text processing applied to the outputs from these two systems, the D_{full} scores were 6.825 for FBK_ASR and 5.476 for our ASR, an 18.1% relative reduction in WER resulted in an 26.9% relative improvement in D_{full} score.

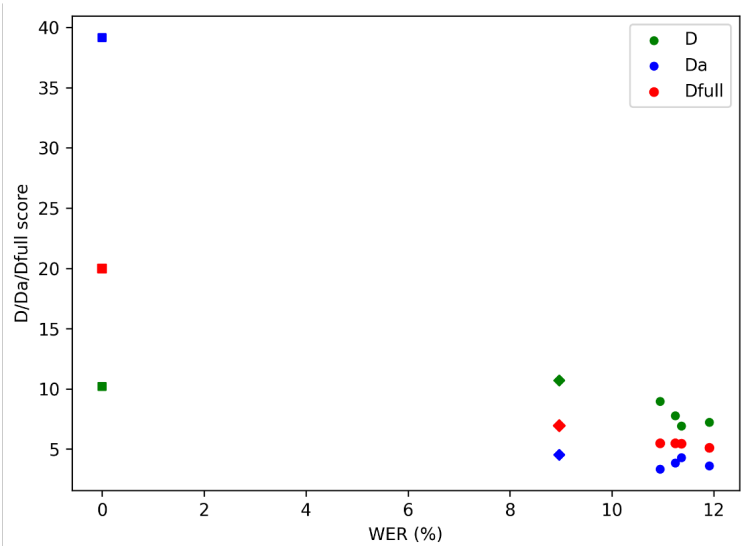


Figure 5.8: *D-WER for outputs from multiple ASRs.*

5.7.3 D scores for synthesised ASR outputs

To further explore the potential of improving ASRs in improving D/D_{full} scores, three different approaches were applied to synthesis ASR outputs.

Synthesis method 1

The first idea is to make use of the n-best ASR hypotheses of our best ASR system. For each set of decoding parameters (lmwt and p), a 1-best hypothesis can be obtained. Varying the

parameters, giving a n-best list where the hypotheses are sorted by WER. The 1-best ASR output has the best overall performance for all the utterances in the test set, but it may not outperform other hypotheses for some particular utterances. Hence, for each utterance in ST3-TST, all the ASR hypotheses in the n-best list from model ST2-sMBR have been compared with the true transcription and the one with the maximum number of correctly recognised words has been chosen as the hypothesis for this utterance, with n varying from 1 to 78. This can be seen as replacing the transcriptions in the 1-best ASR output with better transcriptions from other ASR outputs for each utterance if they exist. Increasing the value of n leads to a lower (or equal) WER.

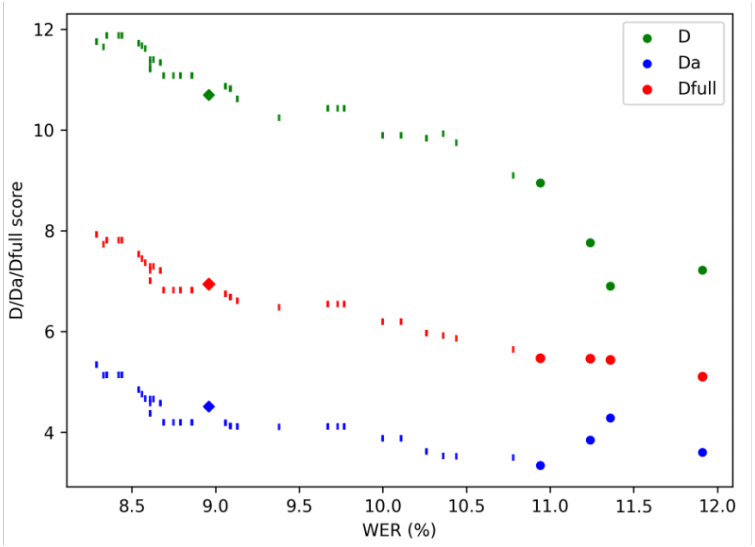


Figure 5.9: *D-WER for outputs from multiple ASRs and synthesised ASR outputs. ♦: FBK output, •: our four ASR outputs, |: fake ASR outputs generated with n-best ASR outputs from ST2-sMBR.*

With this approach, the WER of the simulated ASR outputs improves from 10.94% (n=1) to 8.29% (n=78), the corresponding D_{full} score improves from 5.476 to 7.928. The $D/D_a/D_{full}$ scores of these synthesised ASR hypotheses with respect to WER are plotted in Figure 5.9 (scattered with "|"). The results for FBK output and our outputs from multiple ASRs are also scattered in Figure 5.9. To have a better scale for the synthesised hypotheses, the true transcription which has a D_{full} score of 19.972 is not presented in this figure. It can be seen that it's possible to improve the ASR performance and the overall system

performance (D_{full} score) with n-best ASR hypotheses. The improvement can be significant compared to the 1-best ASR hypothesis. However, there is still a huge gap between the synthesised results and the true transcription.

Synthesis method 2

The second approach to synthesis ASR hypotheses is to make use of the true transcription. For the 1-best ASR output, a randomly selected subset (p%) of incorrect utterances was replaced with the correct ones from the true transcription, p is between 1 and 99. The system results of these data are plotted in Figure 5.10 using "+".

According to Figure 5.10, the relationship between WER and D_{full} score of this type of synthesised data can be seen as a piece-wise linear transform, of which the turning point is at around 3% WER. When the WER is above 3%, the system performance slowly improves when WER decreases. When WER is below 3%, the D_{full} score increases rapidly when WER decreases. In practical, it's very hard to improve the ASR to a WER close or below 3%, especially for non-native children's speech without sufficient in-domain data. Hence, it might be possible to improve the system performance to some extent but it will be extremely difficult to minimize the gap between the system performance of a real ASR output and the true transcription.

Synthesis method 3

The second type of synthesised data showed us the potential D_{full} score that can be obtained with a precise ASR system. To simulate low-error-rate ASR outputs in a different way, the test data (ST3-TST) was added into the training set to train the DNN model, which is not allowed in practical system development.

In the previous experiments, a DNN was pretrained with ST12_train, ST12_dev and 50% of IHM , and it was fine-tuned with ST12_train and ST12_dev. If the pretrained DNN was fine-tuned with all the shared task data including ST3-TST (ST12_train, ST12_dev and ST3-TST), a WER of 4.22% can be obtained for ST3-TST and the corresponding D_{full}

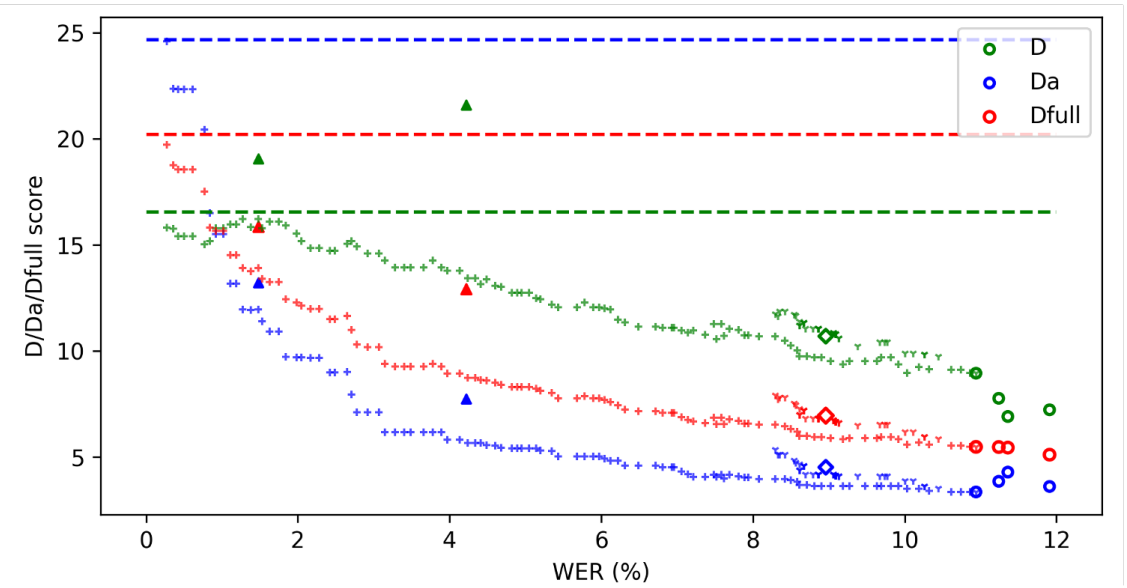


Figure 5.10: *System score (D, D_a, D_{full}) as a function of the WER for four real ASR systems (○), true transcription (- -) and hypothetical outputs from ASRs obtained by Method 1 (ʏ), Method 2 (+) and Method 3 (▲) – see text for description of the methods.*

score is 12.927 with TP2 system (no fusion). If the DNN was fine-tuned with only ST3-TST, the obtained WER is 1.48% and the corresponding D_{full} score is 15.880. These are scattered in Figure 5.10 with "▲". Compared to the second type of synthesised data with similar WERs, these results are closer to real ASR outputs and they have better D_{full} scores. This shows the potential of a good ASR in improving the overall system performance.

5.8 Interaction with the Spoken CALL Shared Task Community

The CALL-ST challenges helped to build a good connection among the community who are interested in speech and language processing for non-native children's speech. UoB winning system in ST1 has been released to the public, so that all the participants in ST2 and ST3 and anyone else that is interested in the task can develop their systems starting from a good

baseline. Support was also provided for others, to help them use UoB system and help them build their own baseline systems. On the other hand, we also benefited a lot from other groups. We were inspired by the use of different types of features to capture the language and meaning correctness as suggested by groups (Magooda and Litman, 2017; Oh et al., 2017). We then proposed to use sentence similarity based on word embeddings to capture the overall correctness of the sentences (Section 5.5). The feature ablation method used in Evanini et al. (2017), to determine the relative contribution of different features, is inspiring. It was used in our other experiments later.

5.9 Summary and Conclusions

This chapter describes the text processing systems developed for the three Spoken CALL Shared Tasks. The baseline rule-based TP system was extended with a post-processing front-end, expanded grammar and a fusion back-end. A machine learning-based TP system was proposed, various aspects of the system were investigated, including the word embedding model to convert ASR outputs to vectors, the algorithm to calculate sentence similarity and the classifier to make the final decision. The performance of these systems was compared, the machine learning-based TP system outperformed the rule-based TP system. However, it is worth noting that in the context of limited scenarios and short sentences as the Spoken CALL Shared Tasks, a big word embedding model may not be necessary and the order of the words in sentence similarity calculation may have some importance. Furthermore, since the TP system is following an ASR system, the influence of the ASR system on the TP system has been explored. We simulated ASR output of a lower WER using three approaches. Results showed that the relationship between WER and the D_{full} score is approximately piece-wise linear with a smaller slope from 11% WER down to 3% WER and a bigger slope as the WER decreases further.

Chapter 6

ASR for Italian Children's English Speech

6.1 Introduction

This chapter describes the ASR systems we developed for the Interspeech 2020 Shared Task on ASR for non-native children's speech. The task uses a corpus of recordings of Italian students taking English language proficiency test. The students are 9 to 16 years old and their ability in spoken English is ranging from minimal to limited but effective. The details of the dataset have been covered in Section 3.2. The 49 hours training set, TLT_Train1P and TLT_Train2P, and the 2 hours development set, TLT-Dev, are used for developing the systems, the evaluation was performed on a 2.3 hours evaluation set (TLT-Eval). These datasets are referred to as "Train1P", "Train2P", "Dev" and "Eval", respectively, in the following sections in this chapter.

The motivation for the shared task is to advance the state-of-art in ASR for children's non-native speech. The shared task includes closed and open tracks. In the closed track,

participants can only use the distributed training data to train the models. In the open track, any data can be used. After the release of the test data, each group was allowed to submit one entry per day and maximum seven entries in total. Submissions were processed immediately after being uploaded. The challenge website was continuously being updated with the best performance achieved by each group.

Various aspects of a ASR system have been explored in this chapter. Lexicon was modified to accommodate mispronunciations and phonological interference. The transcription of the training set was modified, poorly recognised utterances were excluded from training, and training data were augmented with its noise-corrupted versions. Transfer learning was employed in acoustic modelling. Prompt/scenario -based language modelling, language model interpolation and RNNLM rescoring were explored. Fusion of outputs from several ASR systems was performed. Section 6.2 introduces the baseline ASR system built in Kaldi, Section 6.3 to Section 6.8 discuss these different aspects, one section for each aspect. Section 6.9 presents the evaluation on the best systems and Section 6.10 concludes.

6.2 Baseline ASR System

A baseline ASR system was distributed with the shared task by the challenge organisers, which in their experiments achieved a WER of 37.60% on Dev set. The baseline acoustic model was trained on Train1P using Kaldi standard "chain" model and Time-Delay Neural Network (TDNN) (Peddinti et al., 2015; Povey et al., 2018). The initial alignment was produced by a triphone GMM-HMM using standard MFCC features, applying linear discriminative analysis (LDA), maximum likelihood linear transformation (MLLT), feature space maximum likelihood linear regression (fMLLR) and speaker adaptive training (SAT). There were 13 TDNN layers with a size of 1024, one of them is a normal TDNN layer and 12 were TDNN-F (factorized TDNN) layers with a bottleneck size of 128. The features

for TDNN-F training were 40-dimensional MFCCs stacked with 100-dimensional online extracted i-vectors. A 3-way speed perturbation was applied to augment data for AM training, the warping factors were 0.9, 1.0 and 1.1 (Ko et al., 2015). The model was trained with lattice-free maximum mutual information (LF-MMI). Descriptions of TDNN and LF-MMI are provided in Section 2.8 and Section 2.6. The baseline model used the CMU English lexicon (Weide, 1998) plus an extra lexicon generated using Phonetisaurus (Novak et al., 2016) for out-of-vocabulary (OOV) items in the training set. The language model was a 4-gram (Goodman, 2001) trained on "TLT16W17train.trn.txt", which combines written sentences collected in 2016 ("TLT2016Wtrain.trn.txt") with manual transcriptions of Train1P ("TLT2017train.trn.txt").

Our baseline TDNN model (BASE_1P) scored 36.34% WER on Dev set – this differs from the organisers' baseline result due to the use of different hardware and Kaldi versions. Using the same process but combining the 9 and 40 hour training sets (Train1P2P) gave a system (BASE_1P2P) with a WER of 23.6% on Dev set. The breakdown results are given in Table 6.1. Note that, as introduced in Section 3.2, the reference transcription contains the following special cases: i) non-English word sequences are included in '@it()', '@de()' and '@unk()'; ii) hesitations or noises are labelled with initial '@'; iii) truncated and incomprehensible words start or end with '-'; iv) badly pronounced words are marked with an initial '#'. The silence and the first three special cases as given above are removed from the reference and hypothesis for scoring. The symbol '#' in the fourth case is removed but the words are kept.

Table 6.1: *WER (%) of baseline systems on Dev set.*

System	WER (%)			
	A1	A2	B1	Total
BASE_1P	29.75	30.22	42.26	36.34
BASE_1P2P	23.56	19.03	25.33	23.60

From Table 6.1, it's interesting to see that B1 has the highest WER in both BASE_1P and BASE_1P2P systems. Although B1 is the highest proficiency level among the three levels, the questions are designed to be the most difficult and the resulting vocabulary size in the responses is the biggest compared to A1 and A2. This explains the high WER for B1.

6.3 System Development - Lexicon

6.3.1 Italian Pronunciation

Non-native speakers articulate sounds very differently from native speakers, due to the influence of the phonology of their mother language, giving rise to two types of errors: mispronunciation, when they aim to pronounce a wrong target, and phonological interference where they try to use their original phone set (Browning, 2007). Standard British English (SBE) is usually regarded as having 44 phonemes including 20 vowels and 24 consonants, while Italian has 7 vowels and 43 consonants, of which only 22 are shared with English (Browning, 2007). Table 6.2 provides a summary of the English and Italian phoneme system structures. The TLT-school corpus includes instances of mispronunciation and phonological interference, hence modifications were made to the lexicon to accommodate these issues.

Since many Italian words end with a vowel, the lexicon was expanded by adding vowels to the end of some English words to address possible pronunciation errors. For example, the vowel /er/ was added to words ending with 'k/b/p/t/g/d' or /ah/ to words ending with 'a'. Students may use the geminate form when pronouncing English words spelt with double letters because Italian has geminate plosives for these cases. Pronunciation variants with geminate plosives were added to the lexicon for such words. Some of these modifications gave a slight improvement over the baseline lexicon, but only when using a smaller subset of the training set. Hence this approach was discarded.

Table 6.2: *Summary of the English and Italian phoneme system structures. Number of phonemes in English, unique to English, in Italian, unique to Italian, in either English or Italian, shared by English and Italian.*

		Eng	Unique to Eng	Ita	Unique to Ita	Eng +Ita	Shared	Shared (%)
Consonants	Plosive	6	0	12	6	12	6	50.00
	Fricatives	9	4	9	4	13	5	38.46
	Affricates	2	0	8	6	8	2	25.00
	Nasals	3	1	6	4	7	2	28.57
	Liquids	2	0	6	4	6	2	33.33
	Semivowels	2	0	2	0	2	2	100.00
	Total	24	5	43	24	48	19	39.58
Vowels	Monophtong	14	11	7	4	18	3	16.67
	Diphthong	8	8	0	0	8	0	0.00
	Total	22	19	7	4	26	3	11.54

6.3.2 Modelling Non-English Words

Although the students were doing English language test, they were also asked to do German language test, they occasionally fall back to their native language or speak German when they could not find the right word or phrase in English. Knill et al. (2020) analysed the code-switching for each proficiency level of the corpus, presented in Table 6.3. Note, the analysis was based on Train1P and Dev sets because code-switching details were not marked in Train2P reference transcriptions. On average, over 5% of words are non-English words and over 2% utterances are entirely not in English.

Especially, the recordings contain some Italian and German words or phrases which are likely to be confused with English words (e.g. German 'ist' and 'und' are close to English

Table 6.3: *% Italian and German words and utterances only containing non-English in Train1P and Dev sets (Knill et al., 2020).*

Grade	% Words			% Utt
	Italian	German	All	
A1	7.67	2.15	9.82	2.48
A2	4.01	0.86	4.87	2.84
B1	3.88	0.18	4.06	1.54
All	4.87	0.87	5.74	2.38

'is' and 'and'). Although non-English words in the reference are removed for scoring, there will be insertion errors if they are failed to be detected and removed from the hypothesis. The baseline lexicon models all code-switched words as '*<unk-it>*'/'*<unk-de>*'.

In our experiments, the lexicon has been modified to model some foreign words so that the recogniser can learn to distinguish these words from similar English words. Specifically, the non-English words: 'ciao', 'hallo', 'ist', 'ich' and 'und' were modelled as special phones using five states. Adding any of these to the lexicon improved results over the baseline. Including all five words in the lexicon and training using Train1P and Train1P2P with the baseline recipe resulted in WERs of 33.73% and 22.73% on Dev set, respectively. This has reduced the WER by about 7.2% relatively compared to BASE_1P, while a smaller decrease on WER – 3.7% relatively – was observed when trained with Train1P2P. The smaller improvement compared to BASE_1P2P might due to the fact that the code-switched words have not been labelled in Train2P, hence the target code-switched words accounts for a smaller proportion in Train1P2P and have been trained less well.

6.3.3 Modelling Non-Speech Sounds

The transcription of Train1P includes hesitations and noises, e.g. laugh, cough, ah, eh, mm, etc. These non-speech sounds account for 17.1% of the words in Train1P reference transcription which has 27593 words in total. Table 6.4 lists a few non-speech sounds that are most often occurred in Train1P. The sound '@e', '@voice' and '@breath' take up around 9.3% of the total words.

Four 'phone's (hes, laughs, noise and sil) are used to model all of these phenomena in the baseline lexicon. Considering the high occurrence of non-speech sounds in the transcription, the same strategy has been applied to model these sounds as we did for non-English words aiming to reduce the confusion between non-speech sounds and English words. We experimented with modelling one or more types of noises. The best model obtained 33.31% WER on Dev set when "breath" was modelled. However, the benefits of using such modified lexicons went away when Train1P2P was used for training models. The reason for this might be similar as in Section 6.3.2 – Train2P transcription has a small amount of non-speech annotations.

Table 6.4: *Examples of non-speech sounds and statistics of their occurrence in Train1P transcription, which has 27593 words in total including English words, non-English words and non-speech sounds.*

	@e	@voice	@breath	@noise	@em	@bkg	@laugh	@m	@sil	Total
#occur	985	804	778	517	506	341	228	167	93	4711
%occur	3.6	2.9	2.8	1.9	1.8	1.2	0.8	0.6	0.3	17.1

6.4 System Development - Data Augmentation

The coverage of data variability in the training set has a large influence on the quality of the model. Augmentation with speed perturbed data is already included in the baseline model (Section 6.2). Corrupting clean training data with noises can improve the noise-robustness of ASR systems (Gales et al., 2009; Hannun et al., 2014; M. Qian et al., 2016). Recordings in the TLT-school corpus were made in various conditions (microphone, room, background noise). Thus, training data augmentation was explored to address convolutional and additive noise.

For convolutional noise, room impulse responses and point-source noises from (Ko et al., 2017) were used. For additive noise, white Gaussian noise and 37 of the 100 real-world noise types[1] (Hu and D. Wang, 2010) were used. The selection was based on noise frequency characteristics considered likely to appear in the TLT-school corpus. Signal-to-noise ratios (SNRs) of 20 dB and 15 dB were used. In both cases, the 9 hour Train1P set was noise-corrupted and then added to the training set. The baseline recipe (BASE_1P) was used to build the ASR system. Results without and with the augmentation, using only Train1P, and both Train1P and Train2P, are presented in Table 6.5.

Table 6.5: *WER (%) obtained by using convolutional (CONV) and additive (ADD) noise augmentation of the training data.*

Training data used:	Train1P	Train1P2P
Baseline ASR	36.34	23.60
+Train1P_CONV	35.79	23.26
+Train1P_ADD(SNR20dB)	36.38	23.48
+Train1P_ADD(SNR15dB)	34.08	24.19

[1]http://web.cse.ohio-state.edu/pnl/corpus/HuNonspeech/HuCorpus.html.

Augmentation with convolutional noisy data resulted in a small improvement in both training data cases. Results of augmentation with additive noisy data are inconclusive. While the 15 dB SNR data gave better results when the model was trained with the smaller training set (Train1P), raising performance from 36.34% to 34.08%, the 20 dB SNR data gave better results with the larger training set (Train1P2P), but with negligible gain over the baseline.

6.5 System Development - Feature Representation

6.5.1 Setup for MFCCs

The default acoustic features in the Kaldi TDNN model are 40-dimensional MFCCs obtained from 40 mel-frequency bins. These features may not be optimal for children's speech. Children have higher pitch, resulting in further apart spaced harmonics, and formant structure occurs at higher frequencies due to children's shorter vocal tracts. Thus, different MFCC feature extraction regimes were explored: 23 or 40 mel bins, and 13/23 or 20/40 cepstral coefficients used, respectively. Results are presented in Table 6.6. Of these, 20-dimensional MFCCs from 40 mel bins gave the best result (35.10% WER) when trained on Train1P data. However, this benefit did not extend to the model trained on Train1P2P.

6.5.2 Setup for I-vectors

The input features to the TDNN model were 40-dimensional high-resolution MFCCs stacked with 100-dimensional i-vectors. The i-vectors are extracted every 10 frames, using an online decoder. We varied the period for i-vector extraction to 20 and 100 frames. We also explored the use of lower (50) or higher (200) dimensional i-vectors. Using Train1P for training, varying the i-vector period had negligible influence but using 50-dimensional i-vectors

Table 6.6: *Influence of MFCCs with different numbers of cepstrals and bins. The models are trained with Train1P and tested on Dev set.*

#ceps	#mel_bin	Dev (% WER)
13	23	36.32
23	23	35.89
20	40	35.10
40	40	36.45

raised performance to 35.71% WER. However, none of the above methods outperformed BASE_1P2P model when trained on Train1P2P.

6.6 System Development - Acoustic Model

6.6.1 Proficiency-dependent Modelling

The corpus contains data for three English proficiency levels (A1, A2, and B1). As introduced in Section 3.2.5, the prompts have been designed at different difficulty levels for the three levels of tests. Variation in the length and complexity of responses are expected. Hence proficiency-dependent modelling was explored. As each proficiency level involves students with a specific age range, this modelling does to some extent also reflect age-dependent modelling. Age-dependent modelling has been shown to be beneficial for children's speech recognition (Potamianos et al., 1997) but this could be attempted in this Shared Task only using an unsupervised approach as the speaker's age was not available in the corpus.

In an initial experiment, separate proficiency-dependent models for A1, A2 and B1 were trained. This resulted in poorer performance than the BASE_1P model for all three levels. We conclude that any benefit of level-dependent modelling is offset by the reduced

amount of training data per model.

Transfer learning (Ghahremani et al., 2017) was then employed to adapt the baseline model using a subset of each proficiency data, to create three level-dependent models. Only the weights of the output layer were updated. Interestingly, the model transferred with the B1 proficiency data obtained best results for all proficiency subsets of the Dev set, with an overall WER on the dev set of 35.56%. This may be due to the better quality of the spoken language in the B1 part of the corpus. However, results were worse than the baseline when the same transfer learning strategy was applied to the model trained with Train1P2P. The breakdown results of each model on each level are given in Table 6.7.

Table 6.7: *Performance (%WER) of proficiency-dependent models trained with Train1P and Train1P2P.*

	Dev_A1	Dev_A2	Dev_B1	Dev
BASE_1P	29.75	30.22	42.26	36.34
+Train1P_A1	30.87	36.14	47.13	40.28
+Train1P_A2	**29.68**	31.22	44.23	37.50
+Train1P_B1	**29.68**	**29.88**	**40.61**	**35.56**
BASE_1P2P	23.56	19.03	25.33	23.60
+Train1P2P_A1	**23.19**	24.19	29.89	27.08
+Train1P2P_A2	24.53	21.45	27.22	25.41
+Train1P2P_B1	**23.35**	20.95	**24.98**	24.05

6.6.2 Data Selection from Train2P and Transcription

The training sets Train1P and Train2P were annotated differently, and we were initially concerned about the quality of Train2P transcriptions. Thus, decoding of Train2P was run using the model trained on Train1P and the results were analysed. First, an experiment was performed where the model was trained with Train1P2P with these decoded transcriptions

rather than the original Train2P transcription. This resulted in worse performance. Experiments were then performed using only a subset of Train2P data – utterances with the worst WER in the previous experiment were discarded. The best performance was obtained when 95% of the Train2P data was used. It was observed that the abandoned 5% utterances were shorter, containing fewer words and having more cases of '@' and 'unks'. The abandoned 5% utterances have 3.8 words (including @ phenomena and unks) per utterance on average compared to 13.1 words for the whole set of Train2P. The average percentage of actual words in an utterance is 46.9% for these 5% data, while there are averagely 84.5% actual words in the utterance for the whole Train2P set. These figures indicate the poor quality of these discarded 5% utterances.

The following changes have also been applied to Train2P transcriptions: i) numbers were transcribed as text; ii) changed "bye-bye" to "bye bye", "coca-cola" to "coca cola", spelling of "favorite" and "color" to British English spelling "favourite" and "colour". A TDNN model was obtained with the baseline recipe after applying these changes, referred to as "TDNN_1P2Pe95".

6.6.3 Transfer Learning between Training Sets

Since the transcriptions of Train1P appear to be more precise than those of Train2P, we fine-tuned the output layer of the TDNN_1P2Pe95 model with Train1P data for one epoch. Performance rose from 23.60% to 22.65% WER with the baseline language model (baseLM), but decreased from 19.51% to 20.65% WER when a better language model (LM012e) was used. The transcription of the Dev set has the same quality as Train1P, hence TDNN_1P2Pe9 was also fine-tuned with both Train1P and Dev data for one epoch. The corresponding results of this model with both language models, baseLM and LM012e, were in Table 6.8. The lower WERs of the second transferred model does not mean it outperformed the other two models because the evaluation was performed on Dev set which was involved in the training set for

fine-tuning.

Table 6.8: *Results (%WER) for model transferred with Train1P and Train1P+Dev.*

Model	baseLM	LM012e
TDNN_1P2Pe95	23.60	19.51
+trans9hEP1	22.65	20.65
+trans9hDevEP1	14.97	12.94

6.7 System Development - Language Models

6.7.1 N-gram

The baseline system employs a 4-gram language model (LM). As the amount of text for training the LM is small, the use of 3-gram and 2-gram models were explored. These however did not outperform the 4-gram LM.

6.7.2 Using Variations of Training Texts

The baseline 4-gram LM was trained using "TLT16W17train.trn.txt", which includes manual transcriptions of Train1P and written sentences collected in 2016. Different variations of text were used for training the 4-gram LM. A summary of results is given in Table 6.9 and discussed below.

Text "TLT16W17train.trn.txt" contains normalised transcriptions, with non-English sequences replaced by '*<unk>*', '*<unk-it>*' or '*<unk-de>*', and words starting with '@' and truncated words removed. Although foreign phrases, hesitation and noises will not be considered during scoring, it may be useful to keep these details for LM training. The

supervision file "TLT2017train.sup" was used to train a language model LM0, in which non-speech words and truncated words were kept. This is also used as the transcription of Train1P for acoustic model training. Despite this transcription being an order smaller than "TLT16W17train.trn.txt" (27.6k tokens vs 208k tokens) it considerably improved the performance.

Next, LM2 is obtained by training using Train2P transcriptions. Although this provides a larger training text than used in LM0, it did not give as good results as LM0. We then trained LMs with a combination of texts, LM01 with "TLT2017train.sup" (Train1P transcription) and "TLT16W17train.trn.txt", LM02 with Train1P and Train2P transcriptions, LM12 with "TLT16W17train.trn.txt" and Train2P transcription, LM012 with Train1P, Train2P transcription and "TLT16W17train.trn.txt". The LM012, containing the largest amount of text, gave the best performance in nearly all experiments. Finally, the best results were obtained using the LM012e, which is the same as LM012 except using a slightly modified Train2P transcription (as described in Section 6.6.2). This model achieved a WER of 19.51% on Dev set using our best acoustic model.

Table 6.9: *Results (%WER) with different language models.*

Language model	Acoustic model	
	BASE_1P	TDNN_1P2Pe95
baseLM	36.45	23.60
LM0	33.98	22.77
LM2	35.91	23.15
LM012	33.43	20.65
LM012e	32.33	19.51

6.7.3 Interpolation of LMs

Instead of training LM with combined texts, language model interpolation was also performed to combine LMs trained using individual texts. Multiple LMs can be combined with a weight assigned to each language model component. It was observed that such combined LM with appropriately tuned weights can provide slightly improved results but the improvement is negligible when an LM is trained with a larger text (such as LM012).

6.7.4 Prompt- and Scenario-based LMs

In "TLT-school" corpus, students are responding to a set of pre-defined prompts, covering several scenarios. Prompt-based N-gram LMs (with N set to 2, 3, and 4) were trained based on the subset of the LM012e training material corresponding to each prompt. These prompt-based LMs performed worse than LM012e, which is likely due to the lack of training text. Interpolating 3-gram prompt-based LMs with LM012e helped to improve the quality of prompt-based LMs.

Scenario-based LMs were also trained, the scenario can be identified based on the first two digits of the prompt, the name format of the prompts has been explained in Section 3.2.5. This helps model the context of the data while maintaining the training text larger than using prompt-based modelling. The interpolated 3-gram scenario-based LM improved over LM012e and achieved a WER of 18.98% on Dev set. The breakdown by scenario results in Table 6.10 shows that the scenario-based LM outperformed LM0123e on almost every scenario in Dev set. However, it did not improve results on Eval set.

Table 6.10: *The breakdown by scenario results on Dev set with the 3-gram LM012e and the scenario-based 3-gram LM.*

LM	Total	A1		A2		B1	
		en_5_7	en_5_8	en_21_19	en_22_20	en_35_28	en_36_29
LM012e	19.90	21.19	18.92	18.29	14.42	20.30	20.90
scenario-based	18.98	20.59	18.32	17.42	13.94	18.69	21.47

6.7.5 RNNLM Rescoring

Recurrent neural network language models (RNNLM) (Bengio et al., 2003; Mikolov et al., 2011) have surpassed back-off N-gram models in various language-related tasks. In ASR tasks, the common method to use RNNLMs is a 2-pass method. A set of possible hypotheses is produced from decoding on a graph which is generated from a back-off N-gram language model, then a neural-based model is used to rescore the hypotheses (X. Liu et al., 2016; Xu et al., 2018). Many researches have shown that it is useful to use lattice-rescoring in ASR and related task (X. Liu et al., 2016; Sundermeyer et al., 2014; Chen et al., 2017; Khayrallah et al., 2017). In our experiments, we followed the pruned algorithm for performing lattice-rescoring with RNNLMs as described in (Xu et al., 2018). Experiments were performed with different RNNLM setups: varying the dimension of word embeddings, number of training epochs, and weight for RNNLM. By performing lattice rescoring, the recognition on Dev set improved from 19.51% to 18.72% in WER.

6.8 System Fusion

It was observed that although many of the developed ASR systems achieved similar WERs, the actual decoded transcription often differed considerably across the systems. In order to combine the output of different acoustic models, these models have to share the same

decision tree, i.e., the set of senones. As the number of training epochs affects how much a system is fine-tuned to the training data and as such how much robust it will be to unseen testing data, fusion of variants of our best system trained using a different number of epochs was explored.

Firstly, the fusion was performed with two systems, one being the best system using 10 epochs and other varying from 4 to 14 epochs. Then, taking the best two fused systems, we explored adding more systems into the fusion. The best results were obtained using fusion of 2 systems, specifically, epoch 10 and 12. Finally, these 2 systems have been fused with another system which was the 10 epoch system further fine-tuned with the training set Train1P in 1 epoch. This system, with the language model LM012e, achieved a WER of 18.7% on Dev set.

6.9 Evaluation and Submission

During the period of evaluation, 7 entries were submitted to the closed track of the task. All entries involved RNNLM rescoring. The first 3 submissions were derived from the best single acoustic model, a TDNN model with edited transcription and filtered training data (Section 6.6.2), various RNNLMs were employed for language model rescoring.

Submission 4 and 5 utilised the Dev set for acoustic model training in different ways. Submission 4 added Dev set into the training set and trained the TDNN model from scratch, while Submission 5 fine-tuned TDNN_1P2Pe95 with the 9 hour training set and the Dev set. The obtained acoustic models both have lower WER than TDNN_1P2Pe95 on Dev set, this is expected as the model has already seen these data. However, neither of them outperformed TDNN_1P2Pe95 on Eval set.

Submission 6 used three different types of language models instead of the single

LM012e. LM012e was used to decode scenario 'en_36_29' as it outperformed scenario-based LM as presented in Section 6.7.4. Prompt-based LMs were used to decode scenario 'en_5_7' and scenario-based LMs were used to decode the rest scenarios. With RNNLM rescoring, it has a minor improvement on Dev set compared to Submission 1. However, it did not achieve any improvement on Eval set.

Submission 6 and 7 employed fusion of 2 systems and 3 systems, as presented in Section 6.8, and further applied RNNLM rescoring. Results are presented in Table 6.11. The best result on Eval set is 18.33% in WER from the fusion of 3 systems.

Table 6.11: *WER (%) of our best systems on dev set and eval set.*

Submission	ASR system	dev set		eval set
		LM012e	+RNNLM	+RNNLM
Submission 1	TDNN_1P2Pe95	19.51	18.72	19.41
Submission 2	TDNN_1P2Pe95	19.51	18.78	19.56
Submission 3	TDNN_1P2Pe95	19.51	18.86	19.61
Submission 5	+trans9hDevEP1	12.94	12.11	19.59
Submission 4	TDNN_1P2Pe95Dev	13.88	13.10	19.59
Submission 6	TDNN_1P2Pe95	18.84[1]	18.56	19.43
Submission 7	fuse2	19.06	18.17	18.80
-	fuse3	18.70	18.01	18.33

[1] This result was obtained by using a mixture of prompt-based and scenario-based language models, choosing based on the performance on Dev set.

6.10 Conclusion

In this chapter, the systems developed for the Shared Task on Automatic Speech Recognition for Non-Native Children's Speech was described. Expanding the lexicon by non-English words and non-speech sounds were explored. Convolutional and additive noises were used to increase data variability for the training set. Transfer learning was applied to train proficiency-dependent models and to transfer models between two released sets of training data. To obtain a strong language model, different texts and the concatenation of the available texts were used to train the language models. Language model interpolation and prompt/scenario -based language models were also investigated. RNNLM rescoring was applied to further improve the performance and decoding fusion was used to combine the output of different acoustic models. In the experiments, many techniques improved the performance over the baseline model when using the smaller Train1P training set. However, they then in most cases did not outperform BASE_1P2P that was trained on the big Train1P2P training set. The best single model system was trained on Train1P and selected 95% of Train2P with edited transcription and achieved a WER of 19.41% on Eval set with a 4-gram LM and RNNLM rescoring. Fusing this acoustic model with a model trained with two more epochs and a model fine-tuned with the smaller Train1P training set resulted in 18.33% WER on Eval set.

Chapter 7

Comparison between German- and Italian-speaking Children's English Speech

7.1 Introduction

In previous chapters, we presented the ASR performance for German-speaking children's English speech based on the CALL-ST corpus (Chapter 4) and Italian-speaking children's English speech based on the TLT-school corpus (Chapter 6), and also the language assessment systems developed for the CALL-ST corpus. This chapter discusses the similarities and differences between the two corpora, the CALL-ST corpus and the TLT-school corpus, from various aspects, to better understand the properties of the English speech spoken by German-speaking and Italian-speaking Children.

Section 7.2 presents ASR performance for CALL-ST with TDNN-F models which are more advanced and were not explored at the time when the ASR systems were developed

for CALL-ST. This section also explores the benefits of combining two corpora for acoustic model training. Section 7.3 interprets the i-vectors with LDA projection to visualise the datasets of the two corpora in acoustic space and Section 7.4 analyses the two corpora at the text level. Finally, Section 7.5 concludes the chapter.

7.2 ASR Performance

7.2.1 TDNNs for CALL-ST

The technology for speech recognition developed rapidly during recent years. The state-of-the-art acoustic models at the time we developed systems for CALL-ST were no longer the best models by the time this thesis was written. Instead, the time-delay neural network (TDNN) has become a standard acoustic model structure in many speech recognition tasks. It was used as the baseline system in the Italian children's English speech recognition (presented in Chapter 6). To compare the ASR performance for two children's corpora, the CALL-ST corpus and TLT-school corpus, TDNN models were trained for CALL-ST data using the same model structure as described in Section 6.2.

The model consists of a normal TDNN layer followed by 12 TDNN-F layers with a bottleneck size of 128. Each hidden layer has a size of 1024. The input features were 40-dimensional high resolution MFCCs and the alignments were obtained from a triphone GMM model with LDA, MLLT and SAT applied. A 3-way speed perturbation was applied to augment AM training data. The model was trained with lattice-free maximum mutual information. To evaluate the improvement of a TDNN model on ASR performance, the same language model as used in Section 4.5 – LM2, was used. It is a tri-gram model trained with all the training material from ST1 and ST2.

In this Section, a TDNN model was firstly trained with all the ST training recordings

(ST12_train and ST12_dev, referred to as ST12all). The effect of stacking 100-dimensional online extracted i-vectors to 40-dimensional MFCCs was evaluated. The benefits of adding different amounts of AMI-IHM data into the training set for acoustic model training have been discussed in Chapter 4 with results presented in Section 4.4.5. Is data augmentation using an out-of-domain dataset still useful for a more advanced acoustic model structure, e.g. a time-delay neural network? To answer this question, two TDNNs were trained with ST12all plus 20% or 50% of AMI-IHM. The corresponding results are presented in Table 7.1.

Table 7.1: *Recognition results (%WER) obtained from TDNN models on the test sets of 2018 and 2019 Spoken CALL Shared Task.*

Train	i-Vector	ST2-TST	ST3-TST
ST12all	no	7.77	8.91
+IHM20	no	6.60	7.54
+IHM50	no	6.60	7.35
ST12all	yes	7.46	8.66
+IHM20	yes	6.42	**7.31**
+IHM50	yes	**6.40**	7.43

Recall that the best results presented in Chapter 4 were 8.89% WER on ST2-TST and 10.94% WER on ST3-TST obtained from ST2-sMBR, which is a DNN trained with state-level Minimum Bayes Rule (sMBR) discriminative training criterion with ST12all and 50% of IHM. Compared to that, the results obtained from TDNN models are significantly better. The basic TDNN model, trained with only ST12all without i-vectors, achieved a WER of 7.77% on ST2-TST and 8.91% on ST3-TST, which outperform ST2-sMBR by 1.1% and 2.0% absolute on ST2-TST and ST3-TST, respectively. The performance can be further improved when adding IHM data into training. For both ST2-TST and ST3-TST, the improvement is more than 1.1% absolute in WER when 20% IHM is added into the training set. A smaller improvement can be achieved on top of this when augmenting the training set with

50% of IHM. On average, adding i-vectors to the input features can obtain a 3% relative improvement in WER, except testing the TDNN model trained with ST12all and IHM50 on ST3-TST.

All these results show that data augmentation with out-of-domain data is beneficial even on an advanced acoustic model when the in-domain data is limited. The speaker information that i-vectors contain is useful for acoustic modelling, although the benefit is small compared to that obtained from training data augmentation.

7.2.2 TDNNs Trained with Mixed Children's Corpora

The AMI-IHM has been chosen for training data augmentation because it matches with the CALL-ST data in that it is also spontaneous English speech from (mostly) non-native speakers. Since there is no perfect match for CALL-ST, AMI is chosen even though the recordings of AMI are from adults rather than children. The release of TLT-school corpus provided another choice for augmenting the CALL-ST dataset. The TLT-school corpus is also spontaneous English speech from non-native children. It is not the exact match with CALL-ST, the difference includes (a) the age range, the children are aged between 12 and 15 years old in CALL-ST while they are between 9 and 16 in TLT-school, (b) their native language, German and Italian for children in the CALL-ST and TLT-school recordings, respectively, (c) the recording scenarios. Despite these differences, it is worth exploring whether TLT-school is a better match for CALL-ST compared to AMI. Therefore, TDNNs with the same model structure as introduced in Section 7.2.1 were trained with CALL-ST and TLT-school, the input features are 40-dimensional MFCCs stacked with 100-dimensional i-vectors. Note that the CALL-ST corpus is sampled at 8kHz, while the original TLT-school recordings are 16kHz. The TLT-school corpus has been downsampled to 8kHz for training and testing in this section.

Two training sets, TLT_Train1P (TLT9h) and TLT_Train2P (TLT40h), were distributed and they have different quality of transcriptions. To evaluate the TDNNs trained with CALL-ST and TLT-school on the TLT-Dev set, baseline TDNNs models for TLT were trained with downsampled TLT-school training sets. The results were not compared with that obtained from models trained with 16kHz TLT-school recordings, because the reduction of bandwidth itself might lead to a reduction in ASR performance, especially for children's speech (M. Russell et al., 2007). This has been confirmed on TLT-Dev in experiments for models trained with TLT9h or TLT49h. As presented in Table 7.2, the ASR performance is better when the TDNNs are trained and tested on 16kHz data rather than on 8kHz data.

Table 7.2: *Recognition results (%WER) on TLT-Dev obtained from TDNNs trained with 8kHz or 16kHz TLT-school.*

Train	8kHz	16kHz
TLT9h	33.25	32.33
TLT49h	20.63	20.45

The results obtained from models trained with only 8kHz TLT-school data and with both ST12all and 8kHz TLT-school are presented in Table 7.3. The language model for testing ST test sets is LM2 and the LM used for TLT-Dev is LM012e (described in Section 4.5 and Section 6.7, respectively). The use of two corpora results in better models for both ST and TLT-school test sets. For ST2-TST, the best performance was achieved when 40 hours of TLT-school data were included in the training, the WER is 6.49% which is around 1% better than the model without TLT-school data included. For ST3-TST, the model trained with ST12all and 49 hours of TLT-school recordings obtained a WER of 7.12% – outperformed the model trained with only ST12all by 17.8% relatively. For TLT-Dev set, it is beneficial to add ST12all into the training set, especially when the in-domain training data is insufficient. When there are 49 hours of in-domain TLT-school data, adding ST12all into the training does not show any advantage.

Table 7.3: *Recognition results (%WER) obtained from TDNN models trained with CALL-ST and 8kHz TLT-school.*

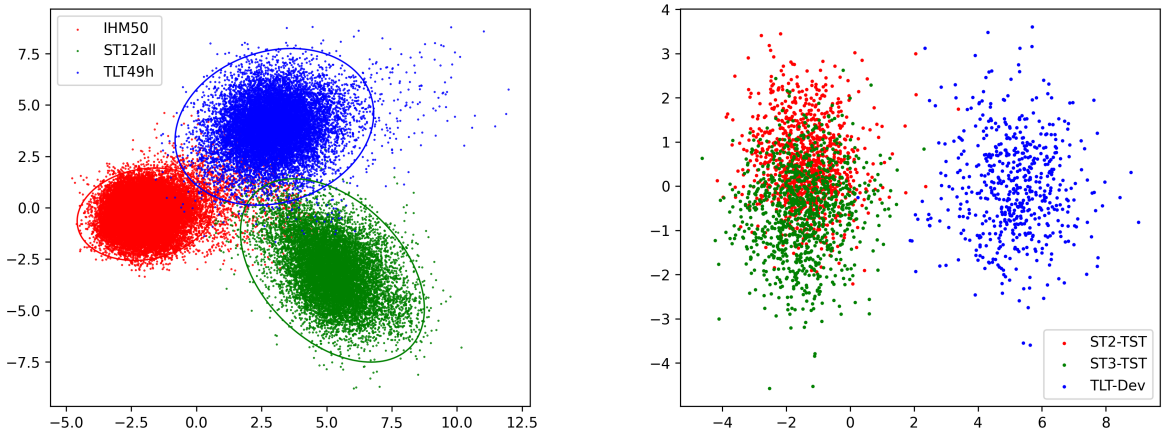
Train	ST2-TST	ST3-TST	TLT-Dev
TLT9h	34.44	37.68	33.25
TLT49h	21.61	22.92	20.63
ST12all	7.46	8.66	65.13
+TLT9h	7.05	7.85	27.45
+TLT40h	6.49	7.41	22.38
+TLT49h	6.68	7.12	20.69

However, it is not clear whether TLT-school or AMI is a better match for CALL-ST. The 40 hours TLT-Train2P has a similar amount of recordings as IHM50, the TDNN models trained with augmenting these two datasets into ST12all did not show a significant difference in ASR performance for ST test sets. When adding TLT40h, the model is 0.11% absolute worse in WER for ST2-TST and 0.02% absolute better for ST3-TST compared to the model with IHM50 added.

7.3 Acoustic Space Analysis

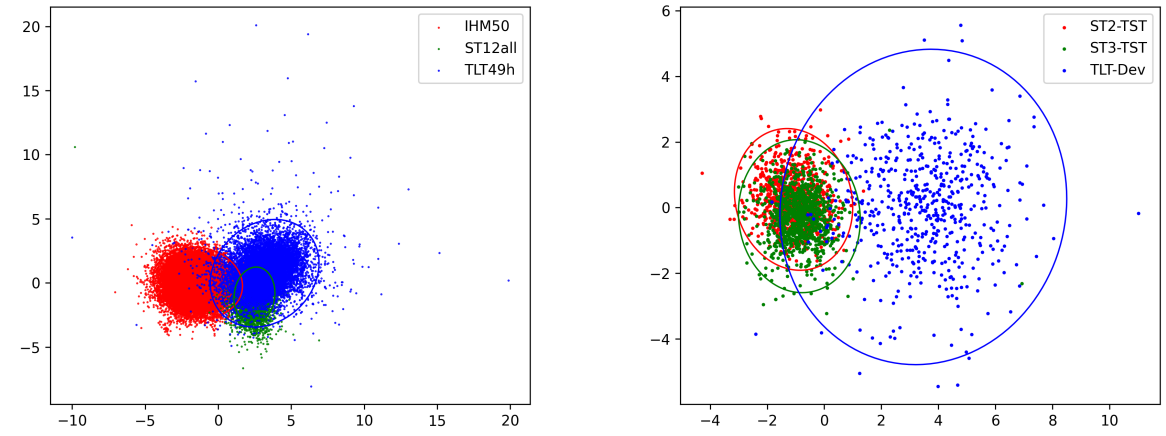
In this section, i-vectors were visualised in 2D space and interpreted to understand the relationship between different datasets. In the previous section, 100-dimensional online-extracted i-vectors were used as part of the input features to train the TDNN models. The i-vectors were trained with 40-dimensional high resolution MFCCs and were extracted every 10 frames – the input to the computation is all frames of the same speaker that are prior to the current frame. To interpret the i-vectors, the extracted i-vectors for each utterance were averaged to get an utterance-level i-vector, then Linear Discriminant Analysis (LDA),

was applied to utterance-level i-vectors to obtain 2-dimensional projections. Each dataset is a different class.



(a) *Training sets* (b) *Test sets*

Figure 7.1: *Visualisations of LDA projections of i-vectors for training sets (ST12all, TLT49h and IHM50) and test sets (ST2-TST, ST3-TST and TLT-Dev). Horizontal axis: the* 1^{st} *dimension of LDA projections, vertical axis: the* 2^{nd} *dimension of LDA projections.*

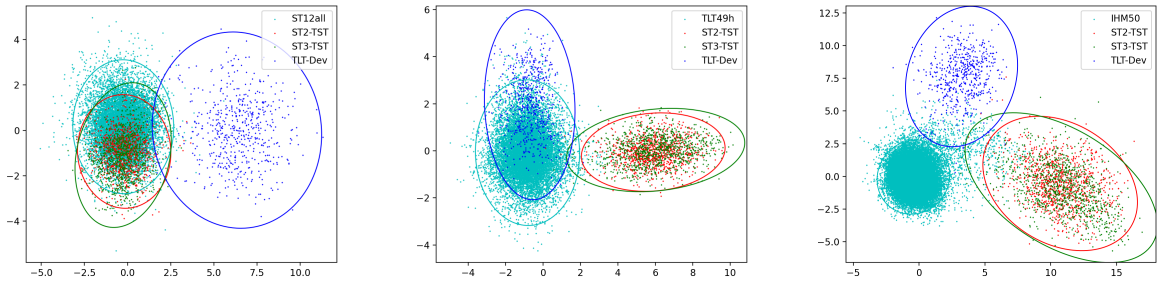


(a) *Training sets* (b) *Test sets*

Figure 7.2: *Visualisations of LDA projections of i-vectors for training sets (ST12all, TLT49h and IHM50) and test sets (ST2-TST, ST3-TST and TLT-Dev). I-vectors are trained with* Δ*MFCCs. Horizontal axis: the* 1^{st} *dimension of LDA projections, vertical axis: the* 2^{nd} *dimension of LDA projections.*

The corpora involved in acoustic model training in the previous section are CALL-ST, TLT-school and AMI, the corresponding training sets of these corpora are ST12all, TLT49h and IHM50. Figure 7.1a presented the LDA projections of i-vectors of these training sets. The confidence ellipse with three standard deviations for each set is also plotted. It's interesting to see that the clusters of the three datasets form a symmetric triangular shape. The sets are all very distinct and each pair appears to be equally distinct. This explains why there is no big difference when adding IHM or TLT-school into CALL-ST for acoustic model training. Figure 7.1b shows that the two test sets of CALL-ST, ST2-TST and ST3-TST, are mixed on the 1^{st} dimension of LDA projection, while TLT-Dev is separated from CALL-ST test sets. This indicates the difference between the two corpora.

Recording environment could be one of the major difference between different corpora. To eliminate the influence of recording conditions, i-vectors were trained with ΔMFCCs and the same approach was applied to visualise utterance-level i-vectors. As presented in Figure 7.2a, there are more overlaps between datasets from different corpora, but the three ellipses are still pointing to three directions. This implies that after eliminating the influence of recording environment, there are still significant differences between the corpora. The difference between test sets from different corpora are also clear as shown in Figure 7.2b.

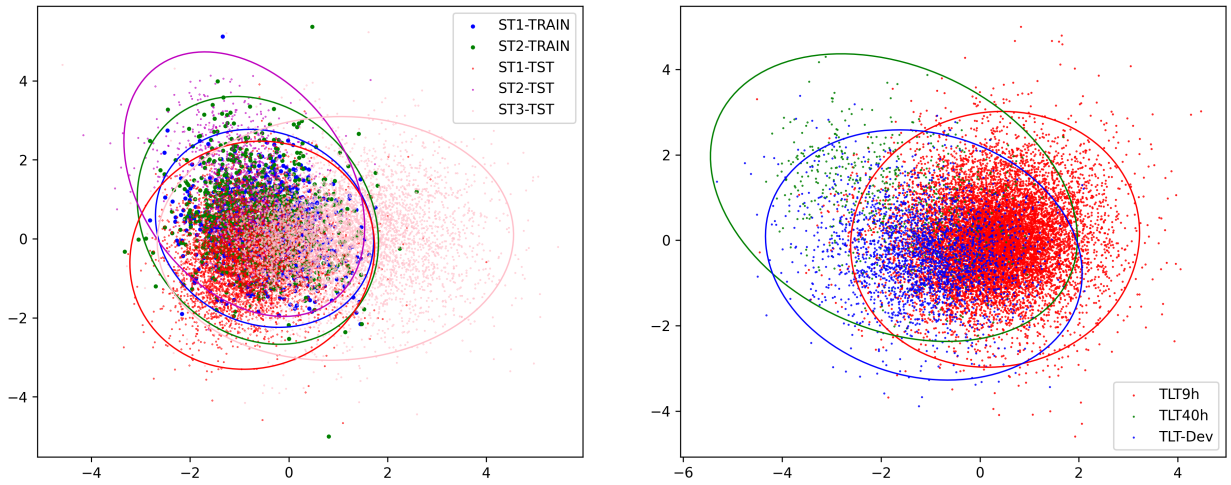


(a) *Test sets and ST12all* (b) *Test sets and TLT49h* (c) *Test sets and IHM50*

Figure 7.3: *Visualisations of LDA projections of i-vectors for test sets (ST2-TST, ST3-TST and TLT-Dev) together with different training sets.*

Each figure in Figure 7.3 presents the LDA projections of i-vectors for test sets (red,

green and blue scatters) and one of the training sets. The light blue scatters in each figure represent a different training set. It can be seen that the training set of CALL-ST corpus (ST12all) is hugely overlapped with in-domain test sets (ST2TST and ST3TST) and separated from out-of-domain test set (TLT-Dev). The same conclusion applies to TLT-school training set. Figure 7.3c shows that IHM50 has similar distance from CALL-ST and TLT-school test sets. This is consistent with the visualisation of the training set i-vectors shown in Figure 7.1a.



(a) *CALL-ST subsets* (b) *TLT-school subsets*

Figure 7.4: *Visualisation of LDA projections of i-vectors for subsets of CALL-ST and TLT-school corpus.*

Since both CALL-ST and TLT-school have multiple subsets, it would be interesting to see how the subsets in each corpus are located in i-vector space. Figure 7.4a and Figure 7.4b are the visualisations of i-vectors of the subsets in CALL-ST and TLT-school corpus. For both corpora, LDA did not manage to separate the subsets. This makes sense as the training and test sets of the same corpus should have similar recording setup and acoustic characteristics. Although there are no clear separations, ST3-TST seems to have a bigger variance in the horizontal axis than other subsets in CALL-ST corpora, this is probably the reason that it is more difficult for the ASR to recognise ST3-TST than other test sets in CALL-ST.

7.4 Text Analysis

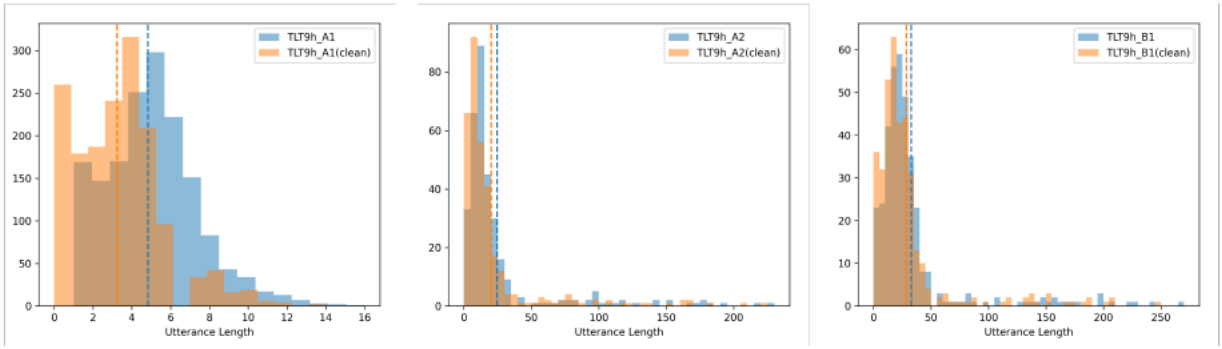
The difference between the two children's speech corpora, the CALL-ST and the TLT-school, is not only in the acoustic level but also in text level. Some statistics about the transcriptions of the training set of these two corpora are reported in Table 7.4.

Table 7.4: *Some statistics about the transcriptions of ST12all and TLT49h datasets: number of utterances, vocabulary size, number of running words, the average number of words in the utterances, the average number of words in the utterances when '@' phenomena and unks are removed, average duration.*

	#utt	#vocab	#run words	avg. words	avg. words (cleaned)	avg. dur (s)
ST12all	12469	1126	66753	5.35	-	3.28
TLT49h	13999	4526	180209	12.87	11.06	12.64
TLT9h	2299	1772	27593	12.00	9.66	14.08
TLT9h_A1	1611	473	7769	4.82	3.23	8.49
TLT9h_A2	326	842	7996	24.53	20.20	25.81
TLT9h_B1	362	1063	11828	32.67	28.75	28.41

It is obvious that TLT-school has a bigger vocabulary size than the CALL-ST corpus and its average utterance length is longer than CALL-ST as well. Multiple types of noises have been labelled with special symbols beginning with '@', code-switching and unknown speech have been marked as unks in TLT-school corpus. After these cases have been removed, the average number of words in TLT49h transcription is 11.06, it's still more than two times longer than utterances in CALL-ST.

TLT-schools recordings are taken at three CEFR levels: A1, A2 and B1. All levels have the same introductory questions, but A2 also includes more open-ended questions and B1 has role-play questions. The differences in questions are reflected in responses. The average number of words of A1 responses in TLT9h is around 3, while A2 and B1 are much



(a) *TLT9h_A1* (b) *TLT9h_A2* (c) *TLT9h_B1*

Figure 7.5: *Utterance length (number of words in the utterance) distribution for 3 subsets in TLT9h, the original transcription (blue) and the cleaned transcription (orange).*

longer – around 20 and 29 words for A2 and B1, respectively. Figure 7.5 displays the number of words distribution in A1, A2 and B1 subsets of TLT9h, the blue histograms are the original transcriptions and the orange ones are the clean transcriptions with '@' phenomena and unks removed. A2 and B1 have longer utterances, the longest one in the B1 subset has more than 250 words, while no utterances in A1 is longer than 16 words. The majority of sentences in A2 and B1 have $0 \sim 50$ words.

Two 4-gram language models, LM2 and LM012e, have been used for testing CALL-ST and TLT-school test sets. LM2 was trained with ST12all transcription and hence has been used for testing CALL-ST test sets. LM012e is used for TLT-school and it was trained with multiple materials from TLT-school, details were described in Section 6.7. The quality of

Table 7.5: *Perplexity of various texts with probabilities from two language models. Texts include ST12all transcription, the original and cleaned transcription for TLT9h and TLT49h. Two LMs are involved: LM2 - the LM trained with ST12all transcription, LM012e - the LM trained with multiple edited TLT-school texts.*

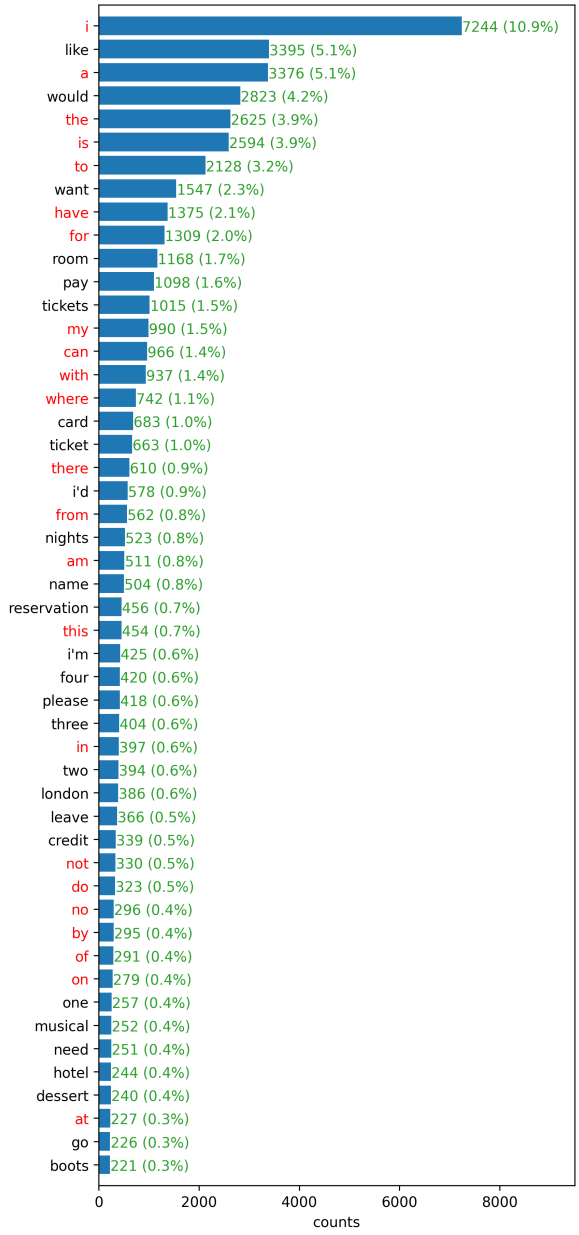
	ST12all	TLT9h	TLT49h	TLT9h (clean)	TLT49 (clean)
LM2	4.26	9485.10	9061.52	3559.71	2733.79
LM012e	113.68	22.21	26.06	19.17	26.69

the language model is hugely influenced by the training material and will affect the ASR performance. Therefore, these two language models were evaluated on multiple texts using the perplexity metric, results are presented in Table 7.5. It is expected that the perplexity of ST12all transcription is low with LM2 and the perplexities of TLT-school transcriptions are low with LM012e. However, it is surprising that the perplexity of LM2 on TLT-school texts is over 9000. It becomes lower when the texts are cleaned, but it is still over 2000. This is mostly due to the limited vocabulary of CALL-ST corpus and it also indicates the variety of LM2 training text is restricted. This explains why the acoustic model trained with only CALL-ST data performs much poorer on TLT-Dev than on CALL-ST test sets, with a WER of 65.13% in Table 7.3.

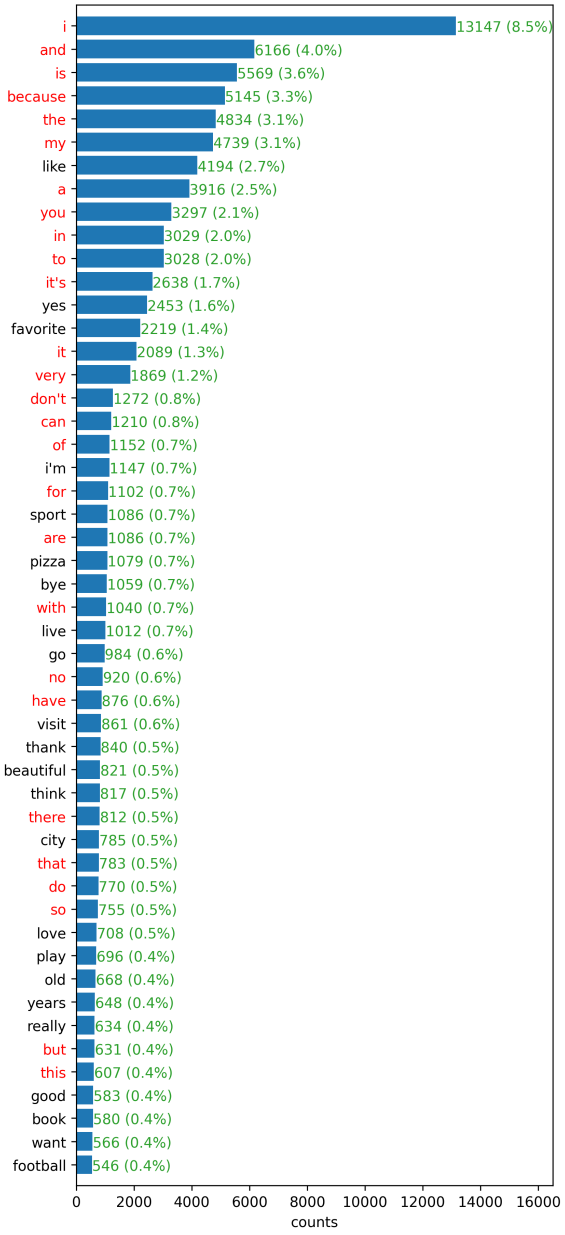
To have a better understanding of the vocabulary in CALL-ST and TLT-school corpus, Figure 7.6 lists 50 most frequent occurred words in the training transcriptions of the two corpora. The words in red are stop words, the green numbers on the right of each bar are the number of occurrences and the frequency of corresponding words. In both corpora, stop words account for around half of the 50 words, and 'i' is the most frequent word. The higher frequency of the most common words in CALL-ST implies the density of the vocabulary and lack of changes in sentences compared to TLT-school corpus.

7.5 Summary and Conclusion

This chapter discusses the differences between a German-speaking children's speech corpus (CALL-ST) and an Italian-speaking children's speech corpus (TLT-school) from multiple perspectives. For ASR performance, it's beneficial to use an advanced acoustic model structure for both corpora. Combining the training sets of two children's corpora for acoustic model training is helpful for both corpora. Applying LDA projection to visualise i-vectors helps understand the relationship between the datasets in acoustic space. It also explains



(a) *ST12all transcription*



(b) *TLT49h transcription*

Figure 7.6: *50 most frequent words in ST12all and TLT49h transcription.*

why training data augmentation using TLT-school corpus which is also a children's corpus for CALL-ST did not outperform that using AMI which is an adults' speech corpus. Finally, the differences between the two corpora have also been analysed at text level. The TLT-school has a bigger vocabulary size, more running words and more varieties in texts, making

it more difficult for speech recognition but also more potential to have a robust recogniser for children's speech.

Chapter 8

Conclusion

8.1 Contributions

The first aim of the work presented in this thesis is to improve ASR performance for children's non-native English speech with a limited in-domain resource. In Chapter 4, we conducted experiments on the CALL-ST corpus which consists of 13.6 hours of recordings collected from German-Swiss children. Specifically, data augmentation was explored with the AMI corpus which contains adults' English speech from mostly non-native speakers and the PF-STAR German corpus which has English recordings from German-speaking children. A hybrid DNN-HMM model is trained with both in-domain and out-of-domain data then fine-tuned with only in-domain data, experiments show that the ASR performance can be improved by around 14% in WER when the combination and amount of out-of-domain data are appropriate. The fMLLR features are more capable of representing the characteristics of children's speech than the MFCC and FBANK features. It's worth noting that utterance-based statistics greatly outperform globally calculated statistics for CMN and fMLLR. Despite the advantage of LSTMs over conventional DNN-HMM models in LVCSR tasks, the benefit is not stable in the experiments when the in-domain resource is limited. However, sequence

discriminative training outperforms cross-entropy training even when the data is limited.

Providing accept/reject feedback to the learners is the first stage in language assessment and needs to be the most accurate. In Chapter 5, we explored different approaches to evaluating the acceptance of the prompt-response pairs from learners where the spoken responses have been converted to text using an ASR system. A rule-based system using a well-defined reference grammar is capable of making judgements for most of the responses. However, it is not able to accept correct responses that are not included in the grammar, which might result in many false rejects (FRs). Hence, a machine learning-based (ML-based) system is proposed which uses a Word2Vec or Doc2Vec model to convert responses into embeddings and calculates the similarities between the responses and prompts. A classifier is trained to make final judgements based on the similarity features. This ML-based system is more flexible and outperforms the rule-based system. Considering that the target sentences are short, using a small Word2Vec or Doc2ec model trained on target resources is better than using a big generally trained model, e.g. the GoogleNews Word2Vec model. Besides, choosing a similarity algorithm that takes the word order into consideration is necessary when aiming at short sentences. In addition, score-level fusion to make use of multiple ASR hypotheses can improve the overall score although the improvement is not big. The text processing systems are based on the output from ASR systems, the relationship between the ASR performance in terms of WER and the TP performance in terms of D/D_{full} is explored. Although the D_{full} score may not have a negative correlation with WER when it comes to outputs from the same ASR with different decoding parameters, improving the ASR can generally improve the system score.

The importance of sufficient training data to an ASR system has been proved by many researchers. We explore how to improve ASR performance for children's speech with limited training data in Chapter 4. With the release of the TLT-school corpus, which contains 51 hours of children's speech and might be adequate for ASR development, we investigate other techniques to improve ASR performance in Chapter 6. Speech technologies have been devel-

oped rapidly over the last ten years, the time-delay neural network has become a standard structure for acoustic modelling and has been used as the baseline system in this chapter. The TLT-school corpus consists of two training set, a 9 hours subset with noises and code-switched words well labelled (TLT9h) and a 40 hours subset with only a few noises labelled (TLT40h). Pronunciation modelling for non-speech sounds and non-English words is first explored, the results show that this is useful when the model is trained on TLT9h, however the benefits have been taken over by the increase of the training data. Similarly, transfer learning to build proficiency-dependent models improves the overall ASR performance when we use the 9 hours training set, but does not show improvement when all of the 49 hours of training data are used. The improvement obtained from the language model is more consistent. Although noises and code-switched words will be ignored while scoring, they need to be detected and removed from the hypothesis to avoid insertion errors. Including these special cases into language modelling contributes to a better ASR. Lattice-rescoring with RNNLMs is performed and shows constant improvements for most of the experiments. System fusion at score level has been conducted for ASR models which share the same decision tree and the best single system can be further improved with this.

Speech recognition for children has been explored on two corpora from different perspectives. In Chapter 7, we compare and contrast the two corpora and explore how to improve the ASR performance with both of the corpora. The TDNN model has been used as the baseline system in Chapter 6, and experiments show that it is also beneficial to use TDNNs for a smaller corpus, i.e. the CALL-ST corpus. Combining the training sets from two corpora for acoustic model training using a TDNN improves the performance for both corpora. The gain for CALL-ST test sets from a combined model over a model trained with in-domain data is higher than that obtained for TLT-Dev, because CALL-ST has insufficient in-domain training data.

Another way to compare the different corpora is to visualise them by projecting them into two-dimensional space. Clusters can be observed from visualisations of i-vectors for the

CALL-ST, TLT-school and AMI corpora in the projected 2-dimensional space, this suggests that there are significant differences between the corpora. The visualisation of i-vectors trained with ΔMFCCs which have eliminated the influence of recording conditions, shows more overlaps between the different corpora suggesting that one of the difference between the corpora is the recording condition. The analysis at text level shows that the TLT-school corpus has more varieties in texts, making it more difficult for speech recognition compared to the CALL-ST corpus. However, these factors also contribute to a more robust speech recogniser.

8.2 Future Work

Pronunciation modelling explored in this thesis does not show advantages for non-native children's speech recognition. A graphemic lexicon, which replaces phones with graphemes together with some predefined attributes, was proposed by Gales et al. (2015) for low-resource languages, where the lexicon required by standard phonetically based systems are often not available or may be inconsistent. It is able to handle spontaneous effects such as hesitations, and has been shown to outperform phonetic lexicons in speech recognition tasks for non-native learners, particularly for lower proficiency speakers (Knill et al., 2017; Knill et al., 2020). In future work, we would like to investigate graphemic lexicons for non-native children's speech.

As previously discussed in this thesis, non-native children's speech often contains several phenomena that can greatly reduce ASR performance. The work covered in this thesis has tried to address the lack of data problem and attempted to model code-switching and non-speech effects. It's also of importance to explore other aspects of children's non-native speech, e.g. the cognitive differences between children and adults. There are known systematic differences between children's speech and adult speech, but current ASR relies

exclusively on ML and doesn't explicitly use this knowledge. It would be interesting to explore whether speech development from infant through child to adult can be modelled and to see if this can be used to aid ASR for child speech. Besides, there is evidence that a lot of the problem with ASR for child speech is down to variability (Fringi, 2019). Can we characterise this variability better to help improve ASR for child speech?

Fusing the recognition output from multiple ASR systems becomes very popular in recent years and has been shown to produce a hypothesis that is more accurate than those from a single system. System fusion at score level is explored in this work, the limitation of this approach is that it can only combine hypotheses from models that share the same decision tree. Confusion network combination (CNC) (Evermann and Woodland, 2000) and ROVER (Evermann and Woodland, 2000) can be applied to combine systems using different trees and have been shown to gain improvements in many tasks (Knill et al., 2020). It would be interesting to compare these different fusion techniques.

The benefits of BPC-dependent DNN structure, inspired by the notion of manifold, for frame-level phone classification and phone recognition have been confirmed (Bai et al., 2017; M. Qian et al., 2018a). Experiments can be conducted to explore its advantages on word-level speech recognition tasks, and its potential to apply to children' non-native speech.

Providing feedback to learners is important to improve their language skills. This thesis focuses on generating accept/reject feedback for learners' spoken responses which is the first stage of the feedback. In future work, we can explore how to generate more detailed feedback, e.g. highlighted errors, more accurate responses. In addition, the feedback relies on interpreting ASR and TP output in terms that are interpretable by humans. This is easy if the models are based on human knowledge. However it becomes increasingly difficult as ASR and TP rely more and more on ML. Hence, there also needs to be an increased emphasis on interpreting ML models/algorithms.

The work in this thesis employed many machine learning models and algorithms,

different aspects of which have been explored and decisions have been made based on experimental results. An interesting question is that whether a machine can be trained to make these choices for human. Meta learning, also known as "learning about learning" or "learn to learn", offers a potential solution to this problem (Vanschoren, 2018; Huisman et al., 2021). Machine learning is to learn a function, f, to model the data given a particular task, while meta learning is to learn a function F that is used to find the function f for a task. Machine learning often requires a large amount of labelled data, which is often not available. The requirement of annotated data is reduced for meta learning, as it can find learning algorithms through previous experience. The selection of network architecture (Zoph et al., 2018; Pham et al., 2018; Real et al., 2019; H. Liu et al., 2018), initial parameters (Finn et al., 2017) and optimizer (Ravi and Larochelle, 2017; Andrychowicz et al., 2016) are all learnable in meta learning. It has also been shown advantages in speech related tasks. Hsu et al. (2020) meta-learned the initialization parameters from many pretraining languages to achieve fast adaptation on unseen target language for low-resource speech recognition. Hou et al. (2021) combined transfer learning with meta learning, to improve the parameter-efficiency when adapting from a multilingual model to a low-resource target language. Lux and Vu (2021) used meta learning to improve rare word recognition in end-to-end ASR. It's promising that meta learning is beneficial in speech and language processing for children's non-native speech.

Appendix A

Phone Recognition using a Non-Linear Manifold with Broad Phone Class Dependent DNNs

Phone Recognition using a Non-Linear Manifold with Broad Phone Class Dependent DNNs

Mengjie Qian, Linxue Bai, Peter Jančovič and Martin Russell

Department of Electronic, Electrical & Systems Engineering, The University of Birmingham, UK

Abstract

Although it is generally accepted that different broad phone classes (BPCs) have different production mechanisms and are better described by different types of features, most automatic speech recognition (ASR) systems use the same features and decision criteria for all phones. Motivated by this observation, this paper proposes a two-level DNN structure, referred to as a BPC-DNN, inspired by the notion of a topological manifold. In the first level, several small separate BPC-dependent DNNs are applied to different broad phonetic classes, and in the second level the outputs of these DNNs are fused to obtain senone-dependent posterior probabilities, which can be used for frame level classification or integrated into Viterbi decoding for phone recognition. In a previous paper using this approach we reported improved frame classification accuracy on the TIMIT corpus compared with a conventional DNN. The contribution of the present paper is to demonstrate that this advantage extends to full phone recognition. Our most recent results show that the BPC-DNN achieves reductions in error rate relative to a conventional DNN of 16% and 8% for frame classification and phone recognition, respectively.

Index Terms: manifold learning, phone classification, speech recognition, neural network, broad phone classes

1. Introduction

State-of-the-art automatic speech recognition (ASR) systems use a single deep neural network (DNN) to define a mapping f from the "acoustic space" A to the space P of vectors of phone (or senone) posterior probabilities, which are integrated into the Viterbi decoder for speech recognition. Although this is a single continuous mapping, in practice the DNN is trained to approximate a discontinuous function f whose outputs typically jump between 0 and 1 across phone state boundaries. Therefore, it may be advantageous to think of f as a *set* of continuous functions $\{f_1, \cdots, f_J\}$, with each function f_j defined on a subset $A_j \subset A$ of the acoustic space and $\bigcup_{j=1}^{J} A_j = A$. In this case the appropriate mathematical structure is a non-linear topological manifold. There have been few studies that have shown the benefits of coding the acoustic speech signal in a non-linear manifold space [1, 2, 3].

In context of speech analysis, the manifold structure provides a tool to exploit the fact that different phonetic classes employ different production mechanisms and are best described by different types of features. Intuitively, one might hope that the subsets A_j correspond to broad phonetic categories. The idea of phone-dependent feature extraction is well-established. For example, while vocal tract resonance frequencies provide a natural description of vowels, unvoiced consonants are better described in terms of duration and mean energies in key frequency bands [4, 5, 6, 7, 8, 9, 10]. There are also a number of studies that use BPC-dependent classifiers to focus on subtle differences between phones within a BPC [11].

A two-level *linear* computational model motivated by these considerations is presented in [12]. The first level comprises a set of discriminative linear transforms, one for each of a set of overlapping broad phone classes (BPCs), that are used for feature extraction. The transforms are obtained using variants of linear discriminant analysis (LDA). Each transform is applied to an acoustic feature vector and k-nearest neighbour methods are used to estimate probabilities of BPCs and phones, which are then combined in the second level to estimate posterior probabilities and hence to classify the acoustic vector. This two-level linear classifier obtained slightly better results compared to a single transform on frame-level phone classification experiments on TIMIT [13].

Inspired by these observations, in our previous study [14] we introduced a two-level *non-linear* model, referred to as a BPC-DNN. Our premise was that it would be advantageous to replace a single 'global' DNN with several BPC-dependent DNNs. In the first level of the BPC-DNN, several small, separate DNNs were applied to different BPCs. For each BPC, a DNN was trained to map acoustic features onto a vector of posterior probabilities of the phones or senones within the BPC, plus an "outside-BPC" class. In the second level the outputs of these DNNs were passed as input to another DNN, the fusion network, which transformed them into a single phone or senone posterior probability vector, which was used for frame level classification. The BPC-DNN is related to Wu and Gales' multi-basis adaptive neural network (MBANN) [15], in which parallel component DNNs correspond to different speaker types.

An obstacle to the application of a topological manifold model to acoustic speech analysis is the need to cover the acoustic space A with phonetically meaningful subsets A_i on which the "feature extraction" transforms f_i are defined. In the approach described here this problem is avoided by applying the BPC-dependent DNN for a particular BPC to the whole of A but only mapping frames corresponding to phones in the BPC to the correct phone class. All other frames are mapped to the "outside-BPC" category.

It was shown in [14] that the BPC-DNN model gives statistically significant improvements in phone-classification of feature vectors, compared with a single global DNN. The contribution of the present paper is to extend this work to full phone recognition by passing the output of the fusion network to a Viterbi decoder.

2. Broad Phone Classes

Broad phonetic classes are defined in terms of a common articulatory strategy that is used in their production. In a BPC-DNN the component DNNs correspond to BPCs or combinations of BPCs. The elements of the TIMIT 49 phone set are partitioned

into 8 non-overlapping BPCs, referred to as $\{G_1, \cdots, G_8\}$ in Table 1. Consonants are divided into "plosives", "strong fricatives", "weak fricatives" and "nasals/flaps" $(G_1, \cdots, G_4)$, liquids and glides are considered as "semi-vowels" (G_5), the vowels are grouped into "short vowels" and "long vowels" (G_6, G_7).We also define 6 'super' phone classes $\{G_9, \cdots G_{14}\}$, which are the union of two or more BPCs from $\{G_1, \cdots, G_8\}$, to combine broad classes that are frequently confused. These are the BPCs from [12].

Table 1: *Broad phone classes and super classes.*

Group	Phonetic class	Phone labels
G_1	Plosive	/b/, /d/, /g/, /k/, /p/, /t/
G_2	Strong fricative	/ch/, /jh/, /s/, /sh/, /z/, /zh/
G_3	Weak fricative	/dh/, /f/, /hh/, /th/, /v/
G_4	Nasal/Flap	/dx/, /en/, /m/, /n/, /ng/
G_5	Semi-vowel	/el/, /l/, /r/, /w/, /y/
G_6	Short vowel	/aa/, /ae/, /ah/, /ax/, /eh/, /ih/, /ix/, /uh/
G_7	Long vowel	/ao/, /aw/, /ay/, /er/, /ey/, /iy/, /ow/, /oy/, /uw/
G_8	Silence	/cl/, /epi/, /q/, /sil/, /vcl/
G_9	$G_5 \cup G_6 \cup G_7$	Semi-vowel, Short vowel, Long vowel
G_{10}	$G_1 \cup G_3$	Plosive, Weak fricative
G_{11}	$G_5 \cup G_6$	Semi-vowel, Short vowel
G_{12}	$G_5 \cup G_7$	Semi-vowel, Long vowel
G_{13}	$G_6 \cup G_7$	Short vowel, Long vowel
G_{14}	$G_1 \cup G_2 \cup ... \cup G_8$	All phones

3. A Two-Level Broad Phone Class DNN (BPC-DNN)

3.1. Upper level: BPC-dependent DNNs

The input to the i^{th} DNN in the upper-level of the BPC-DNN is a filter-bank feature vector in context, and the output is a set of $n_i + 1$ posterior probabilites, one for each of the n_i phone or senone classes in the i^{th} BPC plus an additional "not in the BPC" probability. In this way all of the training data is used to train each upper-level DNN and the need to identify a subset A_i of the acoustic feature space A corresponding the i^{th} BPC is avoided. We explored the use of different combination of BPCs.

3.2. Lower level: single fusion Network

The outputs of all of the upper-level BPC-dependent DNNs are concatenated to form the input to the lower-level fusion network. The output nodes of the fusion DNN correspond to the posterior probabilities of all of the phones or senones in the complete phone set. In the present implementation the fusion network has a single hidden layer. The structure of a two-level BPC-DNN is shown in Figure 1.

4. Experiments with phone-level alignments

This section compares phone-level frame classification and phone recognition results obtained with a conventional single global DNN and a two-layer BPC-DNN system described in Section (3).

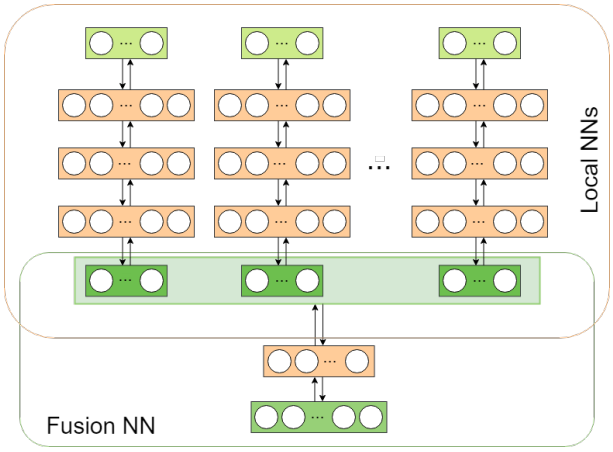


Figure 1: *The structure of a BPC-dependent neural network.*

4.1. Data and features

Experiments were performed on the 16kHz TIMIT speech corpus [13] which has 6300 sentences recorded from 630 speakers. The SA recordings were excluded, hence there are 3696 utterances and 192 utterances in the training and test set, respectively. The 61 phone set used in TIMIT labels is mapped to the 49 phone set [16] for training and testing then be further reduced to 40 [16] for evaluating the results.

4.2. Baseline systems

The baseline ASR model (BASE_MONO1) is a hybrid deep neural network - hidden Markov model (DNN-HMM) trained using the Kaldi toolkit [17]. The speech was encoded as MFCC vectors plus delta and delta-delta coefficients (39 parameters) and used to train a *single state* monophone HMM system (hence there is no distinction between phone and senone labels). The alignments from this monophone model were subsequently used to train a baseline DNN with three hidden layers, each with 1024 nodes.

The inputs to the DNN were 26 dimensional filter-bank features with a context of 11 frames (i.e. ± 5 frames). The output layer is a softmax layer with 49 nodes corresponding to the posterior probabilities of each of the 49 phones.

We evaluated this model in terms of both frame accuracy and percentage phone recognition error. A bi-gram language model trained on the transcriptions in the training set was used for phone recognition. The results are shown in Table 4 (first row).

4.3. BPC-DNN systems (one state per phone)

Each BPC-DNN corresponds to a set of BPCs, which determine the number of BPC-dependent DNNs in the upper layer. We considered five different sets, referred to as $D_1, \cdots, D_5$ in Table 2. D_1 consists of the 8 non-overlapping BPCs from Table (1), while D_2 to D_5 also includes some of the "super groups".

The input features for each local network are the same 26 dimensional filter-bank features with a context of ± 5 frames. Each of the local BPC-dependent DNNs in the upper layer has 3 hidden layers each with 256 nodes. The phone alignment from the baseline model is modified to train the BPC-dependent DNNs in the upper layer. For BPC G_i, the labels in the alignment which correspond to phones in G_i are kept unchanged

Table 2: *Sets $D_1, \cdots, D_5$ of BPCs used to train BPC-DNNs.*

Broad phone class	Experimental setup				
	D_1	D_2	D_3	D_4	D_5
G_1 - G_8	X	X	X	X	
G_9		X	X		
G_{10}			X	X	X
G_{11}				X	X
G_{12}				X	X
G_{13}				X	X
G_{14}					X
# of local DNNs	8	9	10	12	13
total out-nodes in mono1	57	80	92	116	165
total out-nodes in mono3	155	222	256	324	471

while the labels which are not in G_i are all mapped to the same "out-of-group" phone label.

For example, D_1 consists of the 8 BPCs $G_1, \cdots, G_8$ containing $6, 6, 5, 5, 5, 8, 9$ and 5 phones, respectively. Since each BPC-dependent DNN in the upper layer also contains a "not in this class" output, the total number of outputs from the upper layer (inputs to the lower layer) is $49 + 8 = 57$. The number of output nodes from all of the BPC-dependent DNNs in the upper layer for each D_i are shown in Table 2. The original monophone alignment was used to train the fusion network and thus there are 49 output nodes in this network. We only use one hidden layer in the fusion network, but we explored the influence of different number of hidden nodes. We also explored the use of context in the posterior probability vectors that are input to the fusion network.

The results of experiments using 32 and 64 hidden nodes in the fusion DNN, without context (32_0 and 64_0) and with a context of ± 5 phone posterior probability vectors in the input layer (32_5 and 64_5) are shown in Figure 2. The horizontal red dashed line shows the results obtained using the baseline global DNN. All of the results are with respect to the standard 40 phone TIMIT set. The figures show that the two-layer DNNs, with BPC-dependent neural networks in the upper layer and a fusion DNN in the lower layer, outperform the baseline global DNN both in terms of frame accuracy and phone error rate.

Table 3: *Number of parameters (millions) in the 1-state and 3-state per phone-HMM BPC-HMM systems. The corresponding single DNN baseline systems have 2.44 and 2.54 parameters, respectively.*

# of states per phone	Fusion NN	D1	D2	D3	D4	D5
1 state	32_0	1.66	1.87	2.08	2.50	2.72
	32_5	1.68	1.90	2.11	2.53	2.76
	64_0	1.66	1.87	2.08	2.50	2.72
	64_5	1.70	1.93	2.14	2.58	2.83
3 states	32_0	1.69	1.91	2.13	2.56	2.81
	32_5	1.74	1.99	2.21	2.66	2.96
	64_0	1.70	1.93	2.14	2.57	2.83
	64_5	1.80	2.07	2.31	2.78	3.13

For frame classification (Figure 2, top figure), the two-layer BPC-DNN systems corresponding to D_4 and D_5 with a context of ± 5 frames in the input to the fusion DNN achieve a reduction in frame classification error rate of approximately 13% relative to the baseline system. The only cases where the performance of the two-layer system is poorer than the single global baseline network are the experiments for D_1 without context in the input to the fusion layer (these are the blue and green columns on the left of Figure 2). However, these two-layer BPC-DNNs have fewer parameters than the baseline DNN.

For phone recognition (Figure 2, bottom figure), the phone error rates for the two-layer BPC-DNNs are again lower than for the baseline system except for those systems corresponding to D_1 without context in the input to the fusion network. The best system corresponds to D_5, with 32 units in the hidden layer of the fusion DNN and a context of ± 5 frames for the input to the fusion DNN. This system achieves a reduction in error rate of approximately 4% relative to the baseline.

The number of parameters for each system are shown in the top part of Table 3.

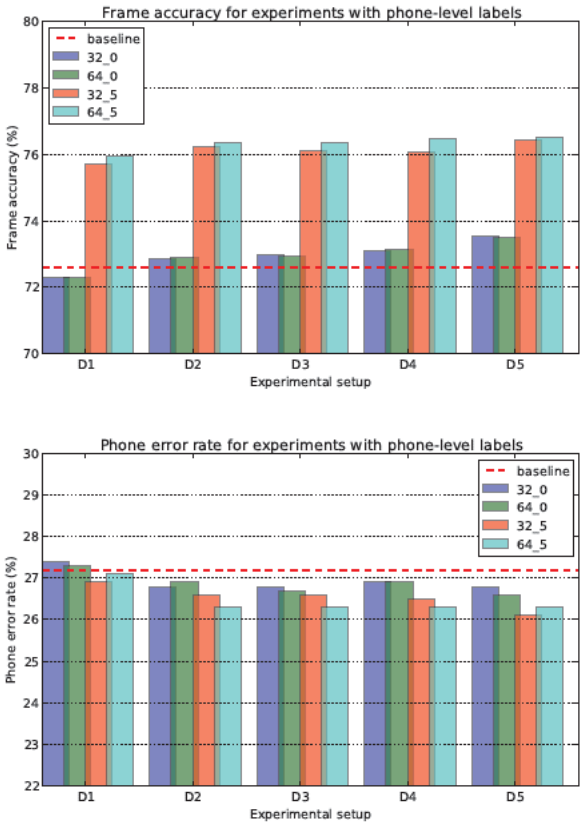


Figure 2: *Percentage Frame accuracy (top) and percentage phone error rate (bottom) for experiments with 32 and 64 hidden nodes and a context of ± 0 and ± 5 frames in the input to the fusion DNN, using the phone-level labels.*

5. Experiments with state-level alignments

For phone recognition, in contrast with the systems described in Section (4), it is normal to use 3 states per phone-level HMM.

This section presents results for frame classification and phone recognition using senone-level alignments obtained with 3 state phone-level HMMs.

5.1. Baseline

A baseline DNN (BASE_MONO3) was trained using the same features and hidden layer structures as in section 4.2, but with different alignments. We trained a monophone GMM-HMM system with 3 states per phone-HMM. In this system there are 147 phone states and the output layer of the BASE_MONO3 DNN has 147 nodes representing the posterior probabilities of these 147 HMM states. The results are shown in Table 4 (second row).

5.2. BPC-DNN systems (three states per phone)

The alignments from sub-section (5.1) that were used to train the baseline were also used in training the BPC-DNNs. When training a local BPC-specfic DNN, the HMM states in the alignment corresponding to phones that are not in this group were again mapped to a "out-of-group" node. We explored the use of combinations BPC-dependent DNNs for the different combinations of BPCs ($D_1, \cdots, D_5$) from Table 2. For each D_i, the total number of output nodes of the local networks are shown in the last row of Table 2. Again we only used one hidden layer in the fusion network, but with different numbers of hidden nodes (32 or 64), and contexts of 0 or ± 5 frames on the input layer of the fusion network.

The frame accuracies and the phone error rate are shown in Figure 3. The number of parameters in each system are shown in the bottom part of Table 3.

Table 4: *Percentage frame accuracies (%FAC) for 147, 49 and 40 targets, and percentage phone error rates (%PER) for the baseline DNN models (base_1 and base_3) and the best performing BPC-DNNs (BPC_1 and BPC_3) with 1 and 3 states per phone.*

	%FAC-147	%FAC-49	%FAC-40	%PER
base_1	-	69.69	72.59	27.2
base_3	61.69	69.64	72.49	26.7
BPC_1	-	73.98	76.52	26.1
BPC_3	66.35	74.46	77.00	25.1

For frame classification (Figure 3, top figure), all of the two-layer BPC-DNN systems outperform the baseline. The best performance, corresponding to D_5 with a context of ± 5 frames in the input to the fusion DNN, achieves a reduction in frame classification error rate of approximately 16% relative to the baseline system. However this BPC-DNN system also has approximately 23% more parameters than the baseline. For BPC-DNN $D4$ with no context the number of parameters is similar to the baseline and the reduction in frame classification error is approximately 4%.

For phone recognition (Figure 3, bottom figure), all of the two-layer BPC-DNN systems again outperform the baseline. With a context of ± 5 the BPC-HMM systems corresponding to D_2, D_3, D_4 and D_5 all achieve a reduction in phone error rate of approximately 6% relative to the baseline. In the cases of D_2 and D_3 this is achieved with fewer parameters than the baseline.

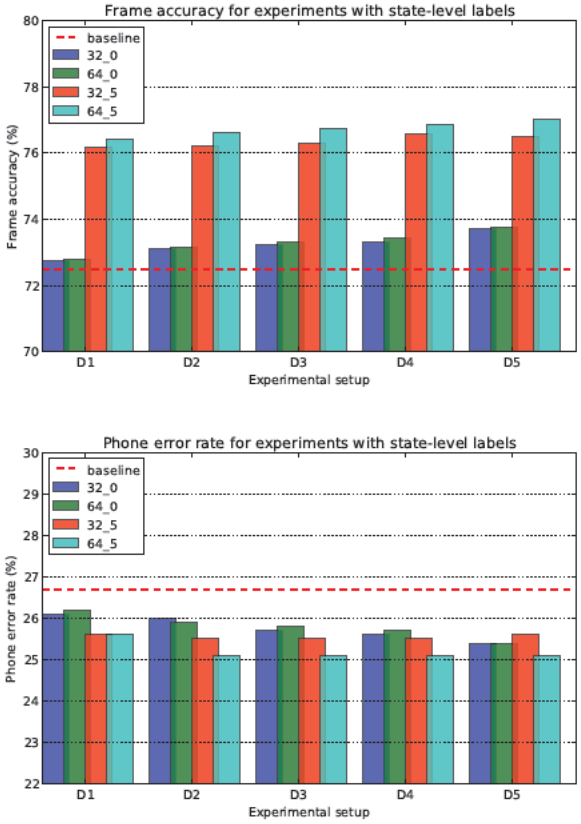


Figure 3: *Percentage Frame accuracy (top) and percentage phone error rate (bottom) for experiments with 32 and 64 hidden nodes and a context of ± 0 and ± 5 frames in the input to the fusion DNN, using the state-level labels.*

6. Conclusions and discussion

This paper describes ongoing research into the application of DNN-based models inspired by the notion of topologial manifold to speech analysis and recognition (BPC-DNNs). Our premise is that such a model is of interest because it reflects the fact that different types of speech sound, corresponding to different modes of production, lend themselves naturally to different types of acoustic analysis. The main conclusion from this work is that the improvement in frame phone classification accuracy previously reported using BPC-DNNs can be extended to phone recognition. Specifically, we obtain a reduction in phone error rate of 6% relative to a conventional DNN using a BPC-DNN with fewer parameters.

The BPC-DNN only approximates a topological manifold structure because the "local" mappings f_i are implemented by DNNs defined on the whole acoustic space A. This raises the question of whether better performance, and more insight, could be obtained with a more faithful manifold structure in which A is covered by proper subsets A_i. For example, if $A_i \cap A_j \neq \emptyset$ and $v \in A_i \cap A_j$ then $f_i(v)$ and $f_j(v)$ could be interpreted as alternative analyses of v from the perspective of different BPCs, and therefore potentially different production mechanisms.

7. References

[1] A. Jansen and P. Niyogi, "Intrinsic fourier analysis on the manifold of speech sounds," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[2] A. Subramanya and J. A. Bilmes, "Entropic graph regularization in non-parametric semi-supervised classification," in *Advances in Neural Information Processing Systems*, 2009, pp. 1803–1811.

[3] Y. Liu and K. Kirchhoff, "Graph-based semi-supervised learning for phone and segment classification." in *INTERSPEECH*, 2013, pp. 1840–1843.

[4] F. Li, A. Menon, and J. Allen, "A psychoacoustic method to find the perceptual cues of stop consonants in natural speech," *The Journal of the Acoustical Society of America*, vol. 127, no. 4, pp. 2599–2610, 2010.

[5] F. Li, A. Trevino, A. Menon, and J. B. Allen, "A psychoacoustic method for studying the necessary and sufficient perceptual cues of American English fricative consonants in noise," *The Journal of the Acoustical Society of America*, vol. 132, no. 4, pp. 2663–2675, 2012.

[6] K. Stevens and S. Blumstein, "Invariant cues for place of articulation in stop consonants," *The Journal of the Acoustical Society of America*, vol. 64, no. 5, pp. 1358–1368, 1978.

[7] J. M. Heinz and K. N. Stevens, "On the properties of voiceless fricative consonants," *The Journal of the Acoustical Society of America*, vol. 33, no. 5, pp. 589–596, 1961.

[8] L. Raphael, "Preceding vowel duration as a cue to the perception of the voicing characteristic of word-final consonants in American English," *The Journal of the Acoustical Society of America*, vol. 51, no. 4B, pp. 1296–1303, 1972.

[9] L. Wilde, "Analysis and synthesis of fricative consonants," Ph.D. dissertation, Massachusetts Institute of Technology, 1995.

[10] A. Khasanova, J. Cole, and M. Hasegawa-Johnson, "Detecting articulatory compensation in acoustic data through linear regression modeling," in *Proc. Interspeech,* Singapore, 2014.

[11] P. Scanlon, D. Ellis, and R. Reilly, "Using broad phonetic group experts for improved speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 803–812, 2007.

[12] H. Huang, Y. Liu, L. ten Bosch, B. Cranena, and L. Boves, "Locally learning heterogeneous manifolds for phonetic classification," *Computer Speech and Language*, vol. 38, pp. 28–45, 2016.

[13] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, *TIMIT Acoustic-Phonetic Continuous Speech Corpus*, Linguistic Data Consortium, Univ. Pennsylvania, Philadelphia, PA, 1993.

[14] L. Bai, P. Jancovic, M. Russell, P. Weber, and S. Houghton, "Phone classification using a non-linear manifold with broad phone class dependent dnns," *Proc. Interspeech 2017*, pp. 319–323, 2017.

[15] C. Wu and M. J. F. Gales, "Multi-basis adaptive neural network for rapid adaptation in speech recognition," in *icassp15*, 2015.

[16] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, and P. Schwarz, "The kaldi speech recognition toolkit," *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

References

Ackley, D. H., G. E. Hinton, and T. J. Sejnowski (1985). "A learning algorithm for Boltzmann machines". In: *Cognitive science* 9.1, pp. 147–169.

Andrychowicz, M., M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas (2016). "Learning to learn by gradient descent by gradient descent". In: *Advances in neural information processing systems*, pp. 3981–3989.

Ateeq, M., A. Hanani, and A. Qaroush (2018). "An Optimization Based Approach for Solving Spoken CALL Shared Task". In: *Proc. Interspeech,* Hyderabad, India, pp. 2369–2373.

Axtmann, N., C. Mehmet, and K. Berkling (2017). "The CSU-K Rule-Based Pipeline System for Spoken CALL Shared Task". In: *Proc. 7th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Stockholm, Sweden, pp. 85–90.

Bahl, L. R., P. F. Brown, P. V. de Souza, and R. L. Mercer (1986). "Maximum mutual information estimation of hidden Markov model parameters for speech recognition". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Tokyo, Japan, pp. 49–52.

Bai, L. (2018). "Speech analysis using very low-dimensional bottleneck features and phone-class dependent neural networks". PhD thesis. University of Birmingham.

Bai, L., P. Jancovic, M. Russell, P. Weber, and S. Houghton (2017). "Phone Classification using a Non-Linear Manifold with Broad Phone Class Dependent DNNs". In: *Proc. Interspeech,* Stockholm, Sweden, pp. 319–323.

Batliner, A. et al. (2005). "The PF_STAR children's speech corpus". In: *Proc. Interspeech,* Lisbon, Portugal, pp. 2761–2764.

Baum, L. E. (1972). "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes". In: *Inequalities* 3.1, pp. 1–8.

Baur, C. (2015). "The Potential of Interactive Speech-Enabled CALL in the Swiss Education System: A Large-Scale Experiment on the Basis of English CALL-SLT". PhD thesis. Université de Genève.

Baur, C., A. Caines, C. Chua, J. Gerlach, M. Qian, M. Rayner, M. Russel, H. Strik, and X. Wei (2018). "Overview of the 2018 spoken CALL shared task". In: *Proc. Interspeech,* Hyderabad, India, pp. 2354–2358.

Baur, C., A. Caines, C. Chua, J. Gerlach, M. Qian, M. Rayner, M. Russell, H. Strik, and X. Wei (2019). "Overview of the 2019 Spoken CALL Shared Task". In: *Proc. 8th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Graz, Austria, pp. 1–5.

Baur, C., C. Chua, J. Gerlach, M. Rayner, M. Russel, H. Strik, and X. Wei (2017). "Overview of the 2017 spoken CALL shared task". In: *Proc. 7th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Stockholm, Sweden, pp. 71–78.

Baur, C., J. Gerlach, E. Rayner, M. Russell, and H. Strik (2016). "A Shared Task for Spoken CALL?" In: *Proc. 10th Int. Conf. on Language Resources and Evaluation (LREC),* Portorož, Slovenia, pp. 237–244.

Bellman, R. (1966). "Dynamic programming". In: *Science* 153.3731, pp. 34–37.

Bengio, Y., R. Ducharme, P. Vincent, and C. Jauvin (2003). "A neural probabilistic language model". In: *Journal of Machine Learning Research* 3.Feb, pp. 1137–1155.

Bengio, Y., P. Simard, and P. Frasconi (1994). "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on Neural Networks* 5.2, pp. 157–166.

Bouselmi, G., D. Fohr, and I. Illina (2007). "Combined acoustic and pronunciation modelling for non-native speech recognition". In: *Proc. Interspeech,* Antwerp, Belgium, pp. 1449–1452.

Bridle, J. S. and L. Dodd (1991). "An Alphanet approach to optimising input transformations for continuous speech recognition". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Toronto, Ontario, Canada, pp. 277–280.

Browning, S. R. (2007). "Analysis of Italian children's English pronunciation". In: URL: http://www.thespeechark.com/PF-STAR/pf-star-italian-children-2004-v1.2.pdf.

Brümmer, N. (2007). "FoCal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition Scores — Tutorial and user manual". In: *Tutorial and User Manual. Spescom DataVoice.* URL: http://sites.%20google.%20com/site/nikobrummer/focalmulticlass.

Caines, A. (2017). "Spoken CALL Shared Task system description". In: *Proc. 7th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Stockholm, Sweden, pp. 79–84.

Carletta, J. (2006). "Announcing the AMI meeting corpus". In: *The ELRA Newsletter* 11.1, pp. 3–5. URL: http://groups.inf.ed.ac.uk/ami/corpus/overview.shtml.

Carletta, J. et al. (2005). "The AMI meeting corpus: A pre-announcement". In: *Int. Workshop on Machine Learning for Multimodal Interaction.* Springer, pp. 28–39.

Chen, X., X. Liu, A. Ragni, Y. Wang, and M. J. Gales (2017). "Future word contexts in neural network language models". In: *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 97–103.

Cheng, J., X. Chen, and A. Metallinou (2015). "Deep neural network acoustic models for spoken assessment applications". In: *Speech Communication* 73, pp. 14–27.

Davis, K. H., R. Biddulph, and S. Balashek (1952). "Automatic recognition of spoken digits". In: *The Journal of the Acoustical Society of America* 24.6, pp. 637–642.

Davis, S. and P. Mermelstein (1980). "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 28.4, pp. 357–366.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1, pp. 1–22.

Dersch, W. (1962). "Shoebox — A voice responsive machine". In: *Datamation* 8.6, pp. 47–50.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186.

Dutta, D. (2017). *Developing an Intelligent Chat-bot Tool to assist high school students for learning general knowledge subjects*. Tech. rep. Georgia Institute of Technology.

Eide, E. and H. Gish (1996). "A parametric approach to vocal tract length normalization". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Atlanta, Georgia, USA, pp. 346–348.

Elenius, D. and M. Blomberg (2005). "Adaptation and normalization experiments in speech recognition for 4 to 8 year old children". In: *Proc. Interspeech,* Lisbon, Portugal, pp. 2749–2752.

Evanini, K., M. Mulholland, E. Tsuprun, and Y. Qian (2017). "Using an Automated Content Scoring Engine for Spoken CALL Responses: The ETS submission for the Spoken CALL Challenge". In: *Proc. 7th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Stockholm, Sweden, pp. 97–102.

Evanini, K., M. Mulholland, R. Ubale, Y. Qian, R. A. Pugh, V. Ramanarayanan, and A. Cahill (2018). "Improvements to an Automated Content Scoring System for Spoken CALL Responses: the ETS Submission to the Second Spoken CALL Shared Task". In: *Proc. Interspeech,* Hyderabad, India, pp. 2379–2383.

Evermann, G. and P. Woodland (2000). "Posterior probability decoding, confidence estimation and system combination". In: *Proc. Speech Transcription Workshop*. Vol. 27. Citeseer, pp. 78–81.

Finn, C., P. Abbeel, and S. Levine (2017). "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International Conference on Machine Learning*. PMLR, pp. 1126–1135.

Fringi, E. (2019). "The Effects of Child Language Development on The Performance of Automatic Speech Recognition". PhD thesis. University of Birmingham.

Fringi, E., J. F. Lehman, and M. Russell (2015). "Evidence of phonological processes in automatic recognition of children's speech". In: *Proc. Interspeech,* Dresden, Germany, pp. 1621–1624.

Fringi, E., J. F. Lehman, and M. J. Russell (2016). "The role of phonological processes and acoustic confusability in phone errors in children's ASR." In: *Proc. 5th Workshop on Child Computer Interaction (WOCCI),* San Francisco, USA, pp. 10–15.

Fringi, E. and M. J. Russell (2018). "Analysis of Phone Errors Attributable to Phonological Effects Associated With Language Acquisition Through Bottleneck Feature Visualisations." In: *Proc. Interspeech,* Hyderabad, India, pp. 2573–2577.

Gales, M. J., K. M. Knill, and A. Ragni (2015). "Unicode-based graphemic systems for limited resource languages". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Brisbane, Australia, pp. 5186–5190.

Gales, M. J., A. Ragni, H. AlDamarki, and C. Gautier (2009). "Support vector machines for noise robust ASR". In: *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 205–210.

Gamper, J. and J. Knapp (2002). "A review of intelligent CALL systems". In: *Computer Assisted Language Learning* 15.4, pp. 329–342.

Garofolo, J., L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Linguistic Data Consortium. URL: https://catalog.ldc.upenn.edu/LDC93S1.

Gerosa, M. and D. Giuliani (2004). "Investigating automatic recognition of non-native children's speech". In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP),* Jeju Island, Korea, pp. 1521–1524.

Gerosa, M., D. Giuliani, and F. Brugnara (2007). "Acoustic variability and automatic recognition of children's speech". In: *Speech Communication* 49.10-11, pp. 847–860.

Gerosa, M., D. Giuliani, S. Narayanan, and A. Potamianos (2009). "A review of ASR technologies for children's speech". In: *Proc. 2nd Workshop on Child, Computer and Interaction (WOCCI),* Cambridge, MA, USA, pp. 1–8.

Gers, F. A. and J. Schmidhuber (2000). "Recurrent nets that time and count". In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN)*, pp. 189–194.

Ghahremani, P., V. Manohar, H. Hadian, D. Povey, and S. Khudanpur (2017). "Investigation of transfer learning for ASR using LF-MMI trained neural networks". In: *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 279–286.

Ghai, S. and R. Sinha (2015). "Pitch adaptive MFCC features for improving children's mismatched ASR". In: *International Journal of Speech Technology* 18.3, pp. 489–503.

Gibson, M. and T. Hain (2006). "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition". In: *Proc. Interspeech,* Pittsburgh, PA, pp. 2406–2409.

Giuliani, D., M. Gerosa, and F. Brugnara (2006). "Improved automatic speech recognition through speaker normalization". In: *Computer Speech and Language* 20.1, pp. 107–123.

Goodman, J. T. (2001). "A bit of progress in language modeling". In: *Computer Speech and Language* 15.4, pp. 403–434.

Graves, A., N. Jaitly, and A.-r. Mohamed (2013). "Hybrid speech recognition with deep bidirectional LSTM". In: *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 273–278.

Gretter, R., M. Matassoni, S. Bannò, and D. Falavigna (2020a). "TLT-school: a Corpus of Non Native Children Speech". In: *Proc. 12th Int. Conf. on Language Resources and Evaluation (LREC)*, pp. 378–385.

Gretter, R., M. Matassoni, and D. Falavigna (2019). "The FBK system for the 2019 Spoken CALL Shared Task". In: *Proc. 8th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Graz, Austria, pp. 6–10.

Gretter, R., M. Matassoni, D. Falavigna, K. Evanini, and C. W. Leong (2020b). "Overview of the Interspeech TLT2020 Shared Task on ASR for Non-Native Children's Speech". In: *Proc. Interspeech,* Shanghai, China, pp. 245–249.

Hacker, C. (2009). "Automatic Assessment of Children Speech to Support Language Learning". PhD thesis. Universität Erlangen-Nürnberg.

Hannun, A. et al. (2014). "Deep Speech: Scaling up end-to-end speech recognition". In: arXiv: 1412.5567 [cs.CL].

Heift, T. (2017). "History and key developments in intelligent computer-assisted language learning (ICALL)". In: *Language, Education and Technology*, pp. 1–12.

Heinz, J. M. and K. N. Stevens (1961). "On the properties of voiceless fricative consonants". In: *The Journal of the Acoustical Society of America* 33.5, pp. 589–596.

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

Hinton, G. E. (2002). "Training products of experts by minimizing contrastive divergence". In: *Neural Computation* 14.8, pp. 1771–1800.

Hinton, G. E., S. Osindero, and Y.-W. Teh (2006). "A fast learning algorithm for deep belief nets". In: *Neural Computation* 18.7, pp. 1527–1554.

Hinton, G. E. et al. (2012). "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". In: *IEEE Signal Processing Magazine* 29.6, pp. 82–97.

Hinton, G. E. (2012). "A practical guide to training restricted Boltzmann machines". In: *Neural Networks: Tricks of the Trade*. Springer, pp. 599–619.

Hochreiter, S. (1991). "Untersuchungen zu dynamischen neuronalen Netzen". In: *Diploma, Technische Universität München* 91.1.

Hochreiter, S. and J. Schmidhuber (1997). "Long short-term memory". In: *Neural Computation* 9.8, pp. 1735–1780.

Hou, W., Y. Wang, S. Gao, and T. Shinozaki (2021). "Meta-Adapter: Efficient Cross-Lingual Adaptation With Meta-Learning". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 7028–7032.

Hsu, J.-Y., Y.-J. Chen, and H.-y. Lee (2020). "Meta learning for end-to-end low-resource speech recognition". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 7844–7848.

Hu, G. and D. Wang (2010). "A tandem algorithm for pitch estimation and voiced speech segregation". In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.8, pp. 2067–2079.

Huang, H., Y. Liu, L. ten Bosch, B. Cranena, and L. Boves (2016). "Locally learning heterogeneous manifolds for phonetic classification". In: *Computer Speech and Language* 38, pp. 28–45.

Huang, J.-X., K.-S. Lee, O.-W. Kwon, and Y.-K. Kim (2017). "A chatbot for a dialogue-based second language learning system". In: *CALL in a climate of change: adapting to turbulent global conditions*, p. 151.

Huisman, M., J. N. van Rijn, and A. Plaat (2021). "A survey of deep meta-learning". In: *Artificial Intelligence Review*, pp. 1–59.

Imseng, D., R. Rasipuram, and M. Magimai-Doss (2011). "Fast and flexible Kullback-Leibler divergence based acoustic modeling for non-native speech recognition". In: *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 348–353.

Jansen, A. and P. Niyogi (2006). "Intrinsic Fourier analysis on the manifold of speech sounds". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Toulouse, France, pp. 241–244.

Johnson, W. L. and A. Valente (2009). "Tactical language and culture training systems: Using AI to teach foreign languages and cultures". In: *AI magazine* 30.2, pp. 72–83.

Jülg, D., M. Kunstek, C. P. Freimoser, K. Berkling, and M. Qian (2018). "The CSU-K Rule-Based System for the 2nd Edition Spoken CALL Shared Task". In: *Proc. Interspeech,* Hyderabad, India, pp. 2359–2363.

Kaiser, J., B. Horvat, and Z. Kačič (2002). "Overall risk criterion estimation of hidden Markov model parameters". In: *Speech Communication* 38.3-4, pp. 383–398.

Kanters, S., C. Cucchiarini, and H. Strik (2009). "The goodness of pronunciation algorithm: a detailed performance study". In: *Proc. 3rd ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Wroxall Abbey Estate, Warwickshire, England, pp. 49–52.

Khasanova, A., J. Cole, and M. Hasegawa-Johnson (2014). "Detecting articulatory compensation in acoustic data through linear regression modeling". In: *Proc. Interspeech,* Singapore, pp. 925–929.

Khayrallah, H., G. Kumar, K. Duh, M. Post, and P. Koehn (2017). "Neural lattice search for domain adaptation in machine translation". In: *Int. Joint Conference on Natural Language Processing (IJCNLP)*, pp. 20–25.

Kingsbury, B. (2009). "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Taipei, Taiwan. IEEE, pp. 3761–3764.

Kingsbury, B., T. N. Sainath, and H. Soltau (2012). "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization". In: *Proc. Interspeech,* Portland, Oregon, USA, pp. 10–13.

Knill, K. M., M. J. Gales, K. Kyriakopoulos, A. Ragni, and Y. Wang (2017). "Use of graphemic lexicons for spoken language assessment". In: *Proc. Interspeech,* Stockholm, Sweden, pp. 2774–2778.

Knill, K. M., L. Wang, Y. Wang, X. Wu, and M. J. Gales (2020). "Non-Native Children's Automatic Speech Recognition: the INTERSPEECH 2020 Shared Task ALTA Systems". In: *Proc. Interspeech,* Shanghai, China, pp. 255–259.

Ko, T., V. Peddinti, D. Povey, and S. Khudanpur (2015). "Audio augmentation for speech recognition". In: *Proc. Interspeech,* Dresden, Germany, pp. 3586–3589.

Ko, T., V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur (2017). "A study on data augmentation of reverberant speech for robust speech recognition". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* New Orleans, Louisiana, USA, pp. 5220–5224.

Krogh, A. and S. K. Riis (1999). "Hidden neural networks". In: *Neural Computation* 11.2, pp. 541–563.

Kusner, M., Y. Sun, N. Kolkin, and K. Weinberger (2015). "From word embeddings to document distances". In: *Int. Conference on Machine Learning*, pp. 957–966.

Lee, J. (2010). *Introduction to topological manifolds*. Vol. 202. Springer Science & Business Media.

Lee, K., S.-O. Kweon, S. Lee, H. Noh, and G. G. Lee (2014). "POSTECH immersive English study (POMY): Dialog-based language learning game". In: *IEICE Transactions on Information and Systems* 97.7, pp. 1830–1841.

Lee, L. and R. C. Rose (1996). "Speaker normalization using efficient frequency warping procedures". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Atlanta, Georgia, USA, pp. 353–356.

Lee, S., A. Potamianos, and S. Narayanan (1999). "Acoustics of Children's Speech: Developmental Changes of Temporal and Spectral Parameters". In: *The Journal of the Acoustical Society of America* 105.3, pp. 1455–1468.

Li, F., A. Menon, and J. B. Allen (2010). "A psychoacoustic method to find the perceptual cues of stop consonants in natural speech". In: *The Journal of the Acoustical Society of America* 127.4, pp. 2599–2610.

Li, F., A. Trevino, A. Menon, and J. B. Allen (2012). "A psychoacoustic method for studying the necessary and sufficient perceptual cues of American English fricative consonants in noise". In: *The Journal of the Acoustical Society of America* 132.4, pp. 2663–2675.

Li, Q. and M. J. Russell (2001). "Why is automatic recognition of children's speech difficult?" In: *7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 2671–2674.

Li, Q. and M. J. Russell (2002). "An analysis of the causes of increased error rates in children's speech recognition". In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP),* Denver, CO, pp. 2337–2340.

Li, X. and X. Wu (2015). "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Brisbane, Australia, pp. 4520–4524.

Liao, H. et al. (2015). "Large vocabulary automatic speech recognition for children". In: pp. 1611–1615.

Liu, H., K. Simonyan, O. Vinyals, C. Fernando, and K. Kavukcuoglu (2018). "Hierarchical Representations for Efficient Architecture Search". In: *6th International Conference on Learning Representations (ICLR).*

Liu, X., X. Chen, Y. Wang, M. J. Gales, and P. C. Woodland (2016). "Two efficient lattice rescoring methods using recurrent neural network language models". In: *IEEE Transactions on Audio, Speech, and Language Processing* 24.8, pp. 1438–1449.

Liu, Y. and K. Kirchhoff (2013). "Graph-based semi-supervised learning for phone and segment classification". In: *Proc. Interspeech,* Lyon, France, pp. 1840–1843.

Livescu, K. and J. Glass (2000). "Lexical modeling of non-native speech for automatic speech recognition". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Istanbul, Turkey, pp. 1683–1686.

Lux, F. and N. T. Vu (2021). "Meta-Learning for improving rare word recognition in end-to-end ASR". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 5974–5978.

Magooda, A. and D. J. Litman (2017). "Syntactic and semantic features for human like judgement in spoken CALL". In: *Proc. 7th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Stockholm, Sweden, pp. 109–114.

Matassoni, M., R. Gretter, D. Falavigna, and D. Giuliani (2018). "Non-native children speech recognition through transfer learning". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Calgary, Alberta, Canada, pp. 6229–6233.

Mermelstein, P. (1976). "Distance measures for speech recognition, psychological and instrumental". In: *Pattern Recognition and Artificial Intelligence*, pp. 374–388.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013a). "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations (ICLR),* Scottsdale, Arizona, USA. URL: http://arxiv.org/abs/1301.3781.

Mikolov, T., S. Kombrink, A. Deoras, L. Burget, and J. H. Cernocky (2011). "RNNLM - Recurrent Neural Network Language Modeling Toolkit". In: *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 196–201. URL: https://www.microsoft.com/en-us/research/publication/rnnlm-recurrent-neural-network-language-modeling-toolkit/.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013b). "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems*, pp. 3111–3119.

Minematsu, N. (2007). "Are Learners Myna Birds to the Averaged Distributions of Native Speakers? - A Note of Warning from a Serious Speech Engineer". In: *Proc. 2nd ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Farmington, PA, USA, pp. 100–103.

Mohamed, A.-r., G. Dahl, and G. Hinton (2009). "Deep belief networks for phone recognition". In: *NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*. Vancouver, Canada.

Nguyen, H., L. Chen, R. Prieto, C. Wang, and Y. Liu (2018). "Liulishuo's System for the Spoken CALL Shared Task 2018". In: *Proc. Interspeech,* Hyderabad, India, pp. 2364–2368.

Noack, R. and L. Gamio (2015). "The world's languages, in 7 maps and charts". In: *The Washington Post* 4.23, pp. 65–70.

Novak, J. R., N. Minematsu, and K. Hirose (2016). "Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework". In: *Natural Language Engineering* 22.6, pp. 907–938.

Oh, Y. R., H.-B. Jeon, H. J. Song, B. O. Kang, Y.-K. Lee, J. G. Park, and Y.-K. Lee (2017). "Deep-Learning Based Automatic Spontaneous Speech Assessment in a Data-Driven Approach for the 2017 SLaTE CALL Shared Challenge". In: *Proc. 7th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Stockholm, Sweden, pp. 103–108.

Oh, Y. R., J. S. Yoon, and H. K. Kim (2007). "Acoustic model adaptation based on pronunciation variability analysis for non-native speech recognition". In: *Speech Communication* 49.1, pp. 59–70.

Olah, C. (2014). *Neural Networks, Manifolds, and Topology*. URL: https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/.

Olah, C. (2015). *Understanding LSTM Networks*. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

Paul, D. B. and J. M. Baker (1992). "The design for the Wall Street Journal-based CSR corpus". In: *Proc. Int. Conf. on Spoken Language Processing (ICSLP),* Banff, Alberta, Canada, pp. 899–902.

Peddinti, V., D. Povey, and S. Khudanpur (2015). "A time delay neural network architecture for efficient modeling of long temporal contexts". In: *Proc. Interspeech,* Dresden, Germany, pp. 3214–3218.

Pele, O. and M. Werman (2008). "A linear time histogram metric for improved sift matching". In: *European Conference on Computer Vision*. Springer, pp. 495–508.

Pele, O. and M. Werman (2009). "Fast and robust earth mover's distances". In: *12th International Conference on Computer Vision*. IEEE, pp. 460–467.

Pennington, J., R. Socher, and C. D. Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Peters, M., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 2227–2237. URL: https://arxiv.org/abs/1802.05365.

Pham, H., M. Guan, B. Zoph, Q. Le, and J. Dean (2018). "Efficient neural architecture search via parameters sharing". In: *International Conference on Machine Learning*. PMLR, pp. 4095–4104.

Poritz, A. B. (1988). "Hidden Markov Models: A Guided Tour". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* New York, NY, pp. 7–13.

Potamianos, A., S. Narayanan, and S. Lee (1997). "Automatic speech recognition for children". In: *5th European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 2371–2374.

Povey, D. et al. (2011). "The Kaldi Speech Recognition Toolkit". In: *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

Povey, D. (2005). "Discriminative training for large vocabulary speech recognition". PhD thesis. University of Cambridge.

Povey, D., G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur (2018). "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks." In: *Proc. Interspeech,* Hyderabad, India, pp. 3743–3747.

Povey, D., V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur (2016). "Purely sequence-trained neural networks for ASR based on lattice-free MMI". In: *Proc. Interspeech,* San Francisco, CA, USA, pp. 2751–2755.

Proença, J., G. Raboshchuk, Â. Costa, P. Lopez-Otero, and X. Anguera (2019). "Teaching American English pronunciation using a TTS service". In: *Proc. 8th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Graz, Austria, pp. 59–63.

Protalinski, E. (2017). *Google's speech recognition technology now has a 4.9% word error rate.* URL: https://venturebeat.com/2017/05/17/googles-speech-recognition-technology-now-has-a-4-9-word-error-rate/.

Pulugundla, B., M. K. Baskar, S. Kesiraju, E. Egorova, M. Karafiát, L. Burget, and J. Cernocký (2018). "BUT System for Low Resource Indian Language ASR." In: *Proc. Interspeech,* Hyderabad, India, pp. 3182–3186.

Qian, M., L. Bai, P. Jančovič, and M. Russell (2018a). "Phone Recognition Using a Non-Linear Manifold with Broad Phone Class Dependent DNNs." In: *Proc. Interspeech,* Hyderabad, India, pp. 3753–3757.

Qian, M., P. Jancovic, and M. Russell (2019). "The University of Birmingham 2019 Spoken CALL Shared Task Systems: Exploring the importance of word order in text processing". In: *Proc. 8th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Graz, Austria, pp. 11–15.

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).* IEEE, pp. 1–5.

Qian, M., X. Wei, P. Jančovič, and M. Russell (2017). "The University of Birmingham 2017 SLaTE CALL shared task systems". In: *Proc. 7th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Stockholm, Sweden, pp. 91–96.

Qian, M., X. Wei, P. Jančovič, and M. Russell (2018b). "The University of Birmingham 2018 spoken CALL shared task systems". In: *Proc. Interspeech,* Hyderabad, India, pp. 2374–2378.

Qian, Y., K. Evanini, X. Wang, C. M. Lee, and M. Mulholland (2017). "Bidirectional LSTM-RNN for Improving Automated Assessment of Non-Native Children's Speech." In: *Proc. Interspeech,* Stockholm, Sweden, pp. 1417–1421.

Rabiner, L. R. (1989). "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". In: *Proceedings of the IEEE* 77.2, pp. 257–286.

Raphael, L. (1972). "Preceding vowel duration as a cue to the perception of the voicing characteristic of word-final consonants in American English". In: *The Journal of the Acoustical Society of America* 51.4B, pp. 1296–1303.

Ravi, S. and H. Larochelle (2017). "Optimization as a Model for Few-Shot Learning". In: *5th International Conference on Learning Representations (ICLR), Toulon, France.*

Rayner, E., C. Baur, C. Chua, and N. Tsourakis (2015). "Supervised learning of response grammars in a spoken CALL system". In: *Proc. 6th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Leipzig, Germany, pp. 83–88.

Rayner, E., P. Bouillon, N. Tsourakis, J. Gerlach, Y. Nakao, and C. Baur (2010). "A multilingual CALL game based on speech translation". In: *Proc. 7th Int. Conf. on Language Resources and Evaluation (LREC),* Valetta, Malta, pp. 1531–1538.

Rayner, E., N. Tsourakis, C. Baur, P. Bouillon, and J. Gerlach (2014). "CALL-SLT: A spoken CALL system based on grammar and speech recognition". In: *Linguistic Issues in Language Technology* 10.2.

Real, E., A. Aggarwal, Y. Huang, and Q. V. Le (2019). "Regularized evolution for image classifier architecture search". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 33. 01, pp. 4780–4789.

Robinson, T., J. Fransen, D. Pye, J. Foote, and S. Renals (1995). "WSJCAM0: a British English speech corpus for large vocabulary continuous speech recognition". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Detroit, Michigan, USA, pp. 81–84.

Rong, X. (2014). "word2vec Parameter Learning Explained". In: arXiv: 1411.2738 [cs.CL].

Rouse, M. (2016). "What is chatbot". In: *Definition from WhatIs. com.*

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). "Learning representations by back-propagating errors". In: *Nature* 323.6088, pp. 533–536.

Russell, M. (2003). *PF_STAR: Preparing for Future Multisensorial Interaction Research. Deliverable D10.*

Russell, M., S. D'Arcy, and L. Qun (2007). "The Effects of Bandwidth Reduction on Human and Computer Recognition of Children's Speech". In: *IEEE Signal Processing Letters* 14.12, pp. 1044–1046.

Russell, M. and S. D'Arcy (2007). "Challenges for computer recognition of children's speech". In: *Proc. 2nd ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Farmington, PA, USA, pp. 108–111.

Russell, M. (2004). *PF_STAR: Preparing for Future Multisensorial Interaction Research. Deliverable D10.*

Sak, H., A. Senior, and F. Beaufays (2014). "Long short-term memory recurrent neural network architectures for large scale acoustic modeling". In: *Proc. Interspeech,* Singapore, pp. 338–342.

Scanlon, P., D. Ellis, and R. Reilly (2007). "Using Broad Phonetic Group Experts for Improved Speech Recognition". In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.3, pp. 803–812.

Serizel, R. and D. Giuliani (2014). "Vocal tract length normalisation approaches to DNN-based children's and adults' speech recognition". In: *2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 135–140.

Serizel, R. and D. Giuliani (2017). "Deep-neural network approaches for speech recognition with heterogeneous groups of speakers including children". In: *Natural Language Engineering* 23.3, pp. 325–350.

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

Shivakumar, P. G., A. Potamianos, S. Lee, and S. S. Narayanan (2014). "Improving speech recognition for children using acoustic adaptation and pronunciation modeling." In: *Proc. 4th Workshop on Child, Computer and Interaction (WOCCI),* Portland, OR, USA, pp. 15–19.

Sokhatskyi, V., O. Zvyeryeva, I. Karaulov, and D. Tkanov (2019). "Embedding-based system for the Text part of CALL v3 shared task". In: *Proc. 8th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Graz, Austria, pp. 16–19.

Stevens, K. and S. Blumstein (1978). "Invariant cues for place of articulation in stop consonants". In: *The Journal of the Acoustical Society of America* 64.5, pp. 1358–1368.

Stolcke, A., J. Zheng, W. Wang, and V. Abrash (2011). "SRILM at sixteen: Update and outlook". In: *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU).* Vol. 5.

Subramanya, A. and J. A. Bilmes (2009). "Entropic graph regularization in non-parametric semi-supervised classification". In: *Advances in Neural Information Processing Systems*, pp. 1803–1811.

Sundermeyer, M., Z. Tüske, R. Schlüter, and H. Ney (2014). "Lattice decoding and rescoring with long-span neural network language models". In: *Proc. Interspeech,* Singapore, pp. 661–665.

Tafazoli, D., C. A. Huertas Abril, and M. E. Gómez Parra (2019). "Technology-based review on Computer-Assisted Language Learning: A chronological perspective". In: *Pixel-Bit: Revista de Medios y Educación* 54, pp. 29–43.

Vanschoren, J. (2018). "Meta-learning: A survey". In: *arXiv preprint arXiv:1810.03548.*

Veselý, K., A. Ghoshal, L. Burget, and D. Povey (2013). "Sequence-discriminative training of deep neural networks". In: *Proc. Interspeech,* Lyon, France. Vol. 2013, pp. 2345–2349.

Vries, B. P. de, S. Bodnar, C. Cucchiarini, H. Strik, and R. van Hout (2013). "Spoken grammar practice in an ASR-based CALL system". In: *Proc. 5th ISCA Workshop on Speech and Language Technology in Education (SLaTE),* Grenoble, France, pp. 60–65.

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, G. and K. C. Sim (2011). "Sequential classification criteria for NNs in automatic speech recognition". In: *Proc. Interspeech,* Florence, Italy, pp. 441–444.

Wang, Y., M. Gales, K. M. Knill, K. Kyriakopoulos, A. Malinin, R. C. van Dalen, and M. Rashid (2018). "Towards automatic assessment of spontaneous spoken English". In: *Speech Communication* 104, pp. 47–56.

Wang, Z., T. Schultz, and A. Waibel (2003). "Comparison of acoustic model adaptation techniques on non-native speech". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Hong Kong, pp. 540–543.

Warschauer, M. (2000). "CALL for the 21st Century". In: *IATEFL and ESADE Conference.* Vol. 2.

Wegmann, S., D. McAllaster, J. Orloff, and B. Peskin (1996). "Speaker normalization on conversational telephone speech". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Atlanta, Georgia, USA, pp. 339–341.

Weide, R. L. (1998). "The CMU pronouncing dictionary". In: *Carnegie Melon University.* URL: http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

Wilde, L. (1995). "Analysis and synthesis of fricative consonants". PhD thesis. Massachusetts Institute of Technology.

Wilpon, J. G. and C. N. Jacobsen (1996). "A study of speech recognition for children and the elderly". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Atlanta, Georgia, USA, pp. 349–352.

Winkler, R. and M. Soellner (2018). "Unleashing the Potential of Chatbots in Education: A State-Of-The-Art Analysis". In: *Academy of Management Proceedings*. Vol. 2018. 1, p. 15903.

Witt, S. M. (2012). "Automatic error detection in pronunciation training: Where we are and where we need to go". In: *International Symposium on Automatic Detection on Errors in Pronunciation Training*, p. 15903.

Witt, S. M. and S. J. Young (2000). "Phone-level pronunciation scoring and assessment for interactive language learning". In: *Speech Communication* 30.2-3, pp. 95–108.

Wu, C. and M. J. Gales (2015). "Multi-basis adaptive neural network for rapid adaptation in speech recognition". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Brisbane, Australia, pp. 4315–4319.

Wu, F., L. P. García-Perera, D. Povey, and S. Khudanpur (2019). "Advances in Automatic Speech Recognition for Child Speech Using Factored Time Delay Neural Network." In: *Proc. Interspeech,* Graz, Austria, pp. 1–5.

Xiong, W., J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig (2017). *Achieving Human Parity in Conversational Speech Recognition*. arXiv: 1610.05256 [cs.CL].

Xu, H., T. Chen, D. Gao, Y. Wang, K. Li, N. Goel, Y. Carmiel, D. Povey, and S. Khudanpur (2018). "A Pruned RNNLM Lattice-Rescoring Algorithm for Automatic Speech Recognition". In: *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Calgary, Alberta, Canada, pp. 5929–5933.

Young, S. et al. (2002). "The HTK book". In: *Cambridge University Engineering Department*.

Young, S. J., J. J. Odell, and P. C. Woodland (1994). "Tree-based state tying for high accuracy modelling". In: *Proceedings of the workshop on Human Language Technology,* Plainsboro, New Jersey, pp. 307–312.

Yu, D. and L. Deng (2010). "Deep learning and its applications to signal and information processing". In: *IEEE Signal Processing Magazine* 28.1, pp. 145–154.

Yu, D. and L. Deng (2016). *Automatic Speech Recognition: A Deep Learning Approach*. Springer.

Zoph, B., V. Vasudevan, J. Shlens, and Q. V. Le (2018). "Learning transferable architectures for scalable image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8697–8710.