RUC-BK-080901-2015201905

Undergraduate Thesis

Computer Assisted Language Learning System for Young English Learners

Author : Jiatong Shi
School : School of Information
Major : Computer Science
ID : 2015201905
Advisor: Qin Jin
Date :

Abstract

English has become one of the most important skills that people desire to acquire especially children under the globalization trend. Comparing to the English reading, writing and listening skills, speaking has been the most difficult part in the learning process as it requires assessment in communication and interaction which is not prevalent in the non-native culture environment. To solve the problem, previous researchers have proposed to utilize Computer-Assisted Language Learning (CALL) systems. With the significant breakthrough in automatic speech recognition, CALL systems have been widely used (such as the University Entrance Exam in China and the Test of English as a Foreign Language (TOEFL)).

As mentioned above, there are massive young English learners in China. However, few studies have focused on the children speech assessment. Evaluation method especially for children speech has not been thoroughly studied.

In this thesis work, we propose to develop a CALL system for young English learners. Firstly, we adopt the state-of-the-art subsampling Time Delay Neural Network (TDNN) and the Lattice Free Maximum Information Mutual Information to train the acoustic model, which leads to better phone recognition performance. Secondly, we propose a Salient Goodness of Pronunciation (SGOP) model based on the pronunciation characteristics of children and posteriorgram from the LF-MMI. The method changes the traditional form of the GOP and achieves a better evaluation. At last, we combine analysis from a word-level phonetic duration model and the SGOP into a Prosodic GOP (PGOP) model which achieves not only performance but also ability to offer prosodic suggestions on the CALL task.

Key words: Computer-Assisted Language Learning (CALL), Time Delay Neural Network (TDNN), Lattice-Free Maximum Mutual Information (LF-MMI), Prosodic Goodness of Pronunciation (PGOP)

Content

Figure List

Table List

Abbreviations

CALL	Computer Assisted Language Learning
CAPT	Computer Assisted Pronunciation Teaching
CE	Cross Entropy
CTC	Connectionist Temporal Classification
DNN	Deep Neural Network
DTW	Dynamic Time Warping
EM	Expectation Maximization
ERN	Extend Recognition Network
fMLLR	Feature-space Maximum Likelihood Linear Regression
FSA	Finite State Acceptor
GOP	Goodness of Pronunciation
HMM	Hidden Markov Model
L1	Native Speaker for certain language
L2	Non-native Speaker for certain language
LDA	Linear Discriminant Analysis
LF-MMI	Lattice-Free Maximum Mutual Information
LM	Language Model
LSTM	Long Short-Term Memory
LVCSR	Large Vocabulary Continuous Speech Recognition
MAE	Mean Absolute Error
MFCC	Mel-Frequency Cepstrum Coefficients
MLLR	Maximum Likelihood Linear Regression
MMI	Maximum Mutual Information
PDF	Probability Density Function
RNN	Recurrent Neural Networks
SAT	Speaker Adaptation Training
SGOP	Salient Goodness of Pronunciation
SVM	Support Vector Machine
TDNN	Time Delay Neural Network
TDNN-F	Time Delay Neural Network with Factorization
VTLN	Vocal Track Length Normalization
WER	Word Error Rate

1. Introduction

Computer Assisted Language Learning (CALL) has received much attention from both the academic and industrial fields. This chapter firstly introduces the motivation of the CALL and then explains the outline of the whole thesis including problems and goals, contributions, and the thesis organization.

1.1 Motivation

At the age of globalization, there is a huge demand for people with different mother tongues to communicate with each other. Consequently, lots of people begin to learn a second language for convenience. Comparing to the reading, writing, and listening skills, speaking has been the most difficult part in the learning process as it requires assessment in communication and interaction which is not prevalent in the non-native culture environment. Based on the success of Automatic Speech Recognition (ASR), Computer Assisted Language Learning (CALL) has become a helpful choice for language learners.

Comparing to traditional language learning processes, CALL has several advantages. 1). it can help to give specific suggestions for students. In traditional teaching within the classroom, due to the limited attention, the teacher cannot offer detailed advice to every student at most of the time. On the contrary, CALL systems have the potential to provide undivided focus to all the users according to the efficiency of computers. 2). judgment from CALL systems is more objective than humans. Under the same context, the CALL system can stay consistent and objective. Instead, the human's judgment is often affected by emotion and other factors. 3). a CALL system can act as an extremely patient teacher without rest. 4). CALL systems allow more flexibility since they can handle different materials instead of several textbooks. 5). from the students' aspect, CALL systems can protect their self-esteem avoiding embarrassment when making mistakes in the class.

However, there are also limitations of CALL systems, including: 1). CALL systems normally predefine the types of feedback to users which are often not enough. For example, the previous CALL systems on GOP [1] only focused on the scoring method. It can be successfully applied in spoken language assessment, but it cannot provide useful suggestions for students on how to correct the mistakes. 2). As a language is the tool for communication, some features of the language can only be detected in a social environment or interaction. For these situations, the CALL system cannot exert its best effect. 3). a perfect teacher always can choose the best way to teach according to his students' characteristics. However, current CALL systems have not taken into consideration the nature of students yet.

Around the beginning of the CALL research, Chapelle proposed 7 principles for CALL [2] including both abundant linguistic input and acoustic output. Reconsidering the criteria, for now, current

CALL systems mainly focused on acoustic output, while the linguistic input is more likely to be implemented in automatic dialogue systems. The nature of CALL that focusing on acoustic output, shows a strong similarity between automatic speech recognition (ASR) and CALL. Therefore, given the state-of-the-art technology, several ASR algorithms can be successfully adopted in CALL fields.

The ASR-based CALL system has been prevailing for recent years. Because of the equitable feature, CALL systems are widely employed in several Official Tests including Senior high school entrance examinations of several Chinese provinces (e.g. Beijing, Shanxi, etc.) and the TOEFL (Test of English as a Foreign Language).

Though the CALL system has drawn great attention from academic and industrial fields, there are few studies on CALL systems for young English learners. It has been long acknowledged by educational researchers that young English learners have a stronger ability in acquiring new language [3]. And according to the requirements of compulsory education in China, Chinese students need to learn a foreign language (mainly English). Therefore, there is also a great need for constructing CALL systems for young English learners.

Table 1-1 Chinese CALL Commercial Systems on English

Products / Company	Type	Functions
IFLYTEK	Backend	Accuracy, Fluency, Integrity, Intonation, Insertion / Deletion
Yun Zhi Sheng	Backend	Classification in 4 Classes (True, False, Skip, Unknown)
Tencent Cloud	Backend	Phoneme and Word Accuracy, Fluency, Stress Position, Integrity, Insertion / Deletion
English Liulishuo	Application	Sentence-level Pronunciation, Tempo, Fluency, and Accuracy
Shengtong	Application	Fluency, Sentence-level and Word-level Pronunciation Score
Alpaca PTE	Application	Word-level Pronunciation, Fluency
Renjiao Spoken Language	Application	Sentence-level Pronunciation
Walk Across American	Application	Word-level Pronunciation
English Reading	Application	Sentence-level Pronunciation
Spoken Language 100	Application	Sentence-level Pronunciation
Microsoft Wheatgrass	Application	Word-level Pronunciation, Fluency, Integrity

Table 1-1 above shows a summary of current commercial applications on CALL in China. From the table, it can be summed up that all the commercial applications basically achieve a scoring system on pronunciation. In addition to the pronunciation scoring, their functions vary on fluency, intonation, and integrity. The tests of these applications indicate that there are still several problems remain. Firstly, the system is weak in robustness. For example, in the *English Reading* shows in **Table 1-1**, if the user properly says only one of ten expected words, the sentence-level score is still high (around 70/100), indicating its weakness. Secondly, some of the applications can only offer an overview score on sentence-level which is not adequate for students to detect their errors in spoken language. In addition, few applications consider intonation or other prosodic features, which is a great perspective for the naturalness of language. At last, most of the applications are not designed for young English learners. Since speeches of young English learners are different from adults' speech to a large extent, the accuracy of the systems falls when a child is using the system.

1.2 Outline

This section shows the outline for this thesis. We first introduce the general problems, tasks and research goal. Then we explain the contributions of the thesis work. Finally, we present the thesis organization in the rest of the manuscript.

1.2.1 Problems and Goal

As discussed in the previous section, CALL systems can help to efficiently and objectively detect the pronunciation problems which is costly to get from a personal human language tutor. Due to the rapid development in ASR and computer hardware, CALL systems begin to have more potential in evolution for the purpose of assisting teachers and self-study. There are three major challenges in the study of CALL. 1). The accuracy on pronunciation error detection. Considering the test discussed on current commercial CALL applications, the CALL systems still have problems in detecting the pronunciation errors accurately. 2). The multi-dimensional evaluation in CALL. According to the consensus on spoken language, the naturalness of a speech is derived from several perspectives, not only from the pronunciation. A natural way in speeches is a combination of appropriate stress, intonation (pitch rise or fall), and rhythm (duration) in phonemes [4]. For CALL with multi-dimensional evaluation, there are two stages. Firstly, CALL systems need to accurately detect the three factors. Then, there should be a scoring model which can determine or find out better prosody for the learners. 3). There are middle states for pronunciation judgment despite exact pronunciation errors. Unlike grammar and spelling errors that have clearly defined boundaries between true and false, the middle states cannot be easily classified as true or false. Hence, the pronunciation scoring should not be converted into a classification task and the score between true and false should be smooth.

Based on the problems discussed, our goal in this thesis work is to develop a CALL system that has following functions:

- Accurately detect pronunciation errors.
- High-grained score utterances, which can separate speaker between true and false pronunciation.
- Highly support the CALL for young English learners
- Simultaneously implement prosodic evaluation to non-native speakers.

1.2.2 Contributions

The main contributions of this thesis are three folds.

1) the traditional Gaussian Mixture Model – Hidden Markov Model (GMM-HMM) acoustic model is replaced by the state-of-the-art Time Delay Neural Networks – Hidden Markov Model (TDNN-HMM) with discriminative training (Lattice Free – Maximum Mutual Information (LF-MMI)) for the CALL tasks for the first time to the best of our knowledge[①]. Unlike the Connectionist Temporal Classification (CTC) methods that cannot provide explicit posteriorgram on all the time span, the TDNN-HMM efficiently generates accurate posteriorgram for further Viterbi alignment which is necessary for CALL systems.

2) the SGOP (Salient Goodness of Pronunciation) is proposed to adapt to the TDNN-HMM model for phonemes scoring in the CALL of young English learners. The method is also robust to the misalignment in the decoding process.

3). a prosodic model for duration suggestion is proposed. The prosodic model can offer fluency suggestions for beginners with duration information. Based on it, the SGOP is further extended to the PGOP (Prosodic GOP) which significantly outperforms the GOP method.

In addition to the main technical contributions, we collect a corpus of English speeches from young English learners with annotations of pronunciation scores (namely CALL_2K, either for CALL or ASR). All the English learners in the CALL_2K are in primary school or kindergarten with a mother tongue of Chinese.

1.2.3 Thesis Organization

Figure 1-1 shows the structure of the thesis.

Chapter 2 discusses the background of the CALL system including an overview of the CALL system and literature review for the CALLs. The overview of the CALL includes well-accepted parts of a CALL system and how they relate to each other. This part is fundamental and necessary for the comprehension of basics of the thesis. The next section presents a literature review on previous CALL

① using the Chain structure trained with Lattice-Free Maximum Mutual Information (LF-MMI) criteria

works. The related works mainly focus on technical improvements instead of empirical studies on the application of CALL. The main parts of a CALL system are discussed in Chapter 3, 4, and 5 within a sequential order for speech processing. For a speech to be evaluated, it is firstly inputted into an acoustic model to detect phonematic features (Chap 3). Then the predicted probabilities are aligned to a standard template (Forced Alignment) based on a decoding model (Chap 4). With the decoded result, the phonetic scoring algorithm is performed (Chap 5). Apart from the pronunciation scoring process contained in Chap 3, 4, and 5, prosodic models are proposed based on the decoding model and it is further applied to improve the pronunciation scoring algorithm (Chap 6). Finally, the conclusions sum up the contributions and several possible directions in the future.

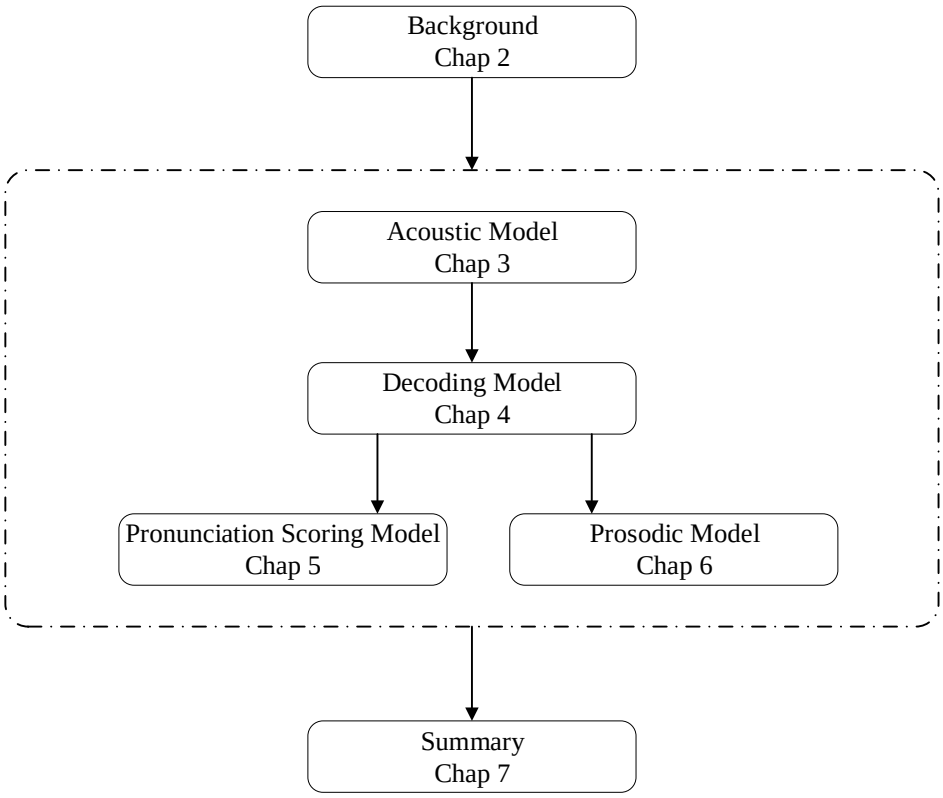


Figure 1-1 Thesis Organization

2. Background

In this chapter, we provide some background which consists of two parts. The first part discusses the overview of CALL systems including general system structure and annotation levels of CALL corpora and the second part presents related technical works for CALL systems.

2.1 CALL System: An Overview

Computer-Assisted Language Learning (CALL) is a general category that applies computer technology to give suggestions on a language learning process for non-native learners (L2 speaker). The general CALL includes four basic aspects of language learning: reading, listening, speaking, and writing. It has a long history from 1960s. The first recorded CALL system can be traced back to 1959, called PLATO (Programmed Logic/Learning for Automated Teaching Operations). It is designed for Russian grammar tutoring [5]. The PLATO system mainly serves language learning in the writing section. The writing is not the toughest part in language learning. Among the four basic parts (reading, writing, listening and speaking) for L2 students, the speaking is the most difficult since it can be easily intervened by L2's mother tongue and it is hard to find a suitable environment for practicing, especially for the L2 speaker. However, due to the poor computing resources and a lack of speech processing knowledge, little focus has been put on speaking training until the 1990s. Computer-Aided/Assisted Pronunciation Training (CAPT) is a specific name for CALL on speaking training which is also the focus of this thesis[①]. Riding on the wave of Automatic Speech Recognition (ASR), CALL drew much attention in a way of combining with the ASR [1]. As for now, CALL is still following the same structure as the ASR. The basic structure of a CALL system is shown in **Figure 2-1**.

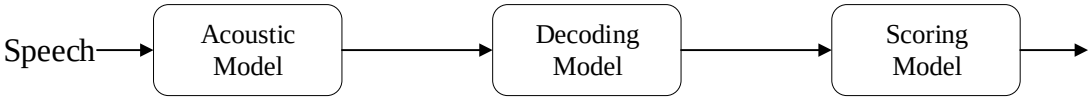


Figure 2-1 Structure of a CALL system

The general structure of a CALL system mainly consists of three components: an acoustic model, a decoding model, and a scoring model.

Acoustic model converts speech information to phonetic feature (sometimes accompanied with prosodic features). Its target is to accurately recognize phonemes and outputs a posteriorgram that represents posterior probabilities for phonemes at each frame (For most cases in ASR and CALL, the posteriorgrams are hidden in the system based on the HCLG where "H" is for the Hidden Markov Model, "C" is for the Context-Dependent phonemes, and "L" is for the Language Model. The HCLG is a Finite State Transducer). The acoustic model should be dependent on speakers, so speaker adaptation methods are often applied at this stage. The main challenges in the acoustic model for the CALL are low discriminativeness in similar phonemes and over-adaptation.

The decoding model aligns or recognizes features (i.e. posteriorgram) into phonetic segments. Inaccurate alignment and arbitrary speaking sentences are the main problems of the decoding model.

① CAPT and CALL are the same in the following parts

At last, a scoring model provides scores for phonetic sequences according to the segments. There are two types of scoring target. The first is a classification that dividing pronunciation into true or false. The second is to offer a continuous score that gradually changes. In addition, the same structure can be applied to prosodic evaluation as well. In most of the previous literature, the prosodic evaluation generally achieved within the scoring process.

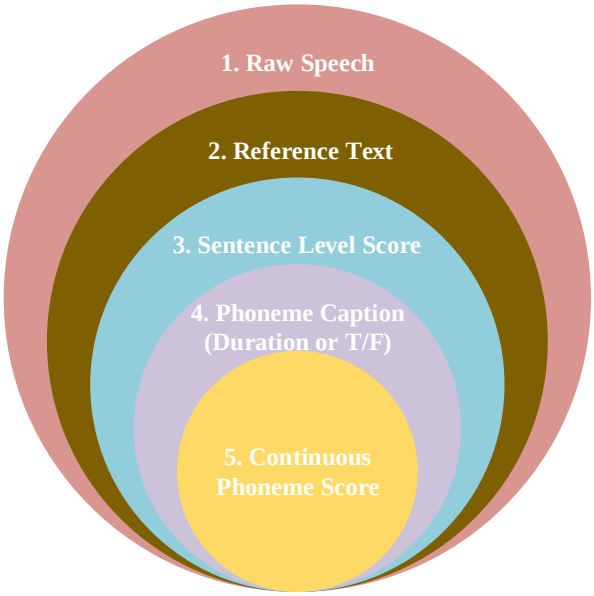


Figure 2-2 Annotation Level of CALL Datasets

The evaluation for each model varies and depends on the dataset's annotation. **Figure 2-2** shows the annotation levels of CALL datasets. For CALL problems, the raw speeches at the first level are difficult to evaluate. The second level datasets are with reference text. They can be applied to an ASR system or a CALL training process, but still not enough for CALL systems' evaluation. From the third level, the corpus begins to be able to CALL system training and evaluation. As the level goes deeper, the evaluation and training processes become more accurate, but the time devoted in construction of datasets would be doubled. Therefore, researches on CALL always have to choose a balance between accurate method evaluation and efficiency in dataset preparing.

2.2 Literature Review on CALL

The literature review on CALL is organized as follows. 1) we define categories of pronunciation errors. 2) we review the Goodness of Pronunciation (GOP) algorithm, a baseline in the CALL. Based on the weakness of the GOP, methods for confusing phonemes and poor alignment are discussed respectively. 3) other algorithms for scoring were also discussed later. 4) we introduce words in the speaker adaptation, specific language characteristics, and pronunciation caption tasks.

In CALL systems, the computer acts as a teacher that offers an evaluation for L2 students' pronunciation errors. According to previous researches [6], pronunciation errors can be categorized into phonetic errors and prosodic errors. A simple Venn diagram (**Figure 2-3**) identifies all related pronunciation errors under the two categories.

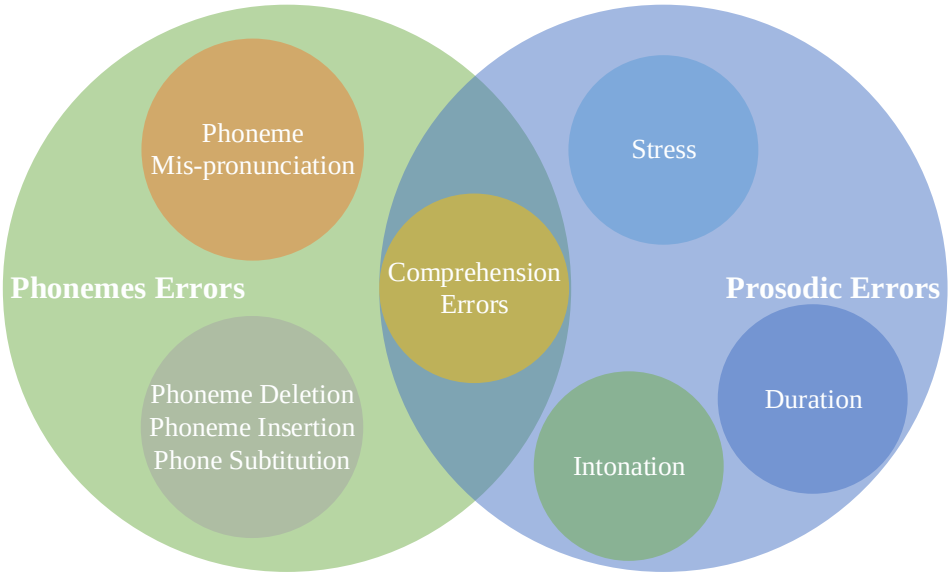


Figure 2-3 Pronunciation Errors

The phonetic errors are related to individual phoneme. The difference between phonetic mispronunciation and other categories (i.e. deletion, insertion, and substitution) is that phonetic mispronunciation occurs when the pronunciation can be poorly recognized while other statuses cannot be identified. Stress, intonation, and duration can be sum up to prosodic errors (namely rhythm in some literature). These topics concentrate on pronunciation errors within a multi-dimensional perspective. There is also an inter-error which belongs to both sides of the errors, called comprehension errors. The comprehension errors are severe errors that affect comprehension of listeners, which make the sentence hard to comprehend even by a skilled native speaker. Mostly the error occurs when there are too many phonetic errors and prosodic errors. The difficulties for recognizing this kind of error are how to determine a threshold. Based on these error types, the literature on CALL can be divided into phonetic CALL and prosodic CALL.

Studies on phonetic CALL were earlier than studies in prosodic CALL and they are more widely discussed in the CALL researches. Bernstein et al. [7] first proposed a pronunciation evaluation system based on ASR technology in 1990. He employed ASR to recognize L1 and L2 speeches, then compared the reference texts with the recognized tests for scoring. After correlation analysis with human raters, he proved the effectiveness of his system in providing feedbacks to language learners. The weakness of his study is that his corpus for CALL only contains limited sentences (6 sentences) which are not enough to draw to conclusions. After seven years, in 1997, Witt proposed a likelihood method for scoring [1],

namely Goodness of Pronunciation (GOP). Based on forced alignment with posteriorgram, the GOP was calculated as following **Formula (2-1)**.

$$GOP(p) = log\big(P(p|O)\big) / L(O) = log(\frac{P(O|p)P(p)}{\sum_{q\in Q} P(O|q)P(q)})/ L(O) \quad (2\text{-}1)$$

where O represents the acoustic segment of a phoneme, Q stands for phonetic dictionary, p represents the phoneme to be scored, and $L(O)$ represents the length of the segment. The GOP could compute scores for each phoneme for any given utterances and it was the first accepted algorithm in ASR-based CALL scoring methods. Because of its usefulness, the GOP scoring algorithm was often applied as a baseline model for CALL tasks. In the following paper [8], Witt further proposed an evaluation metric for human rater since human scoring is subjective. And in the same paper, she added adaptation methods in the acoustic modeling process and set individual thresholds for GOP scores for further mispronunciation classification.

The GOP method could help to efficiently detect most of the mispronunciation errors, but it still had several disadvantages including the bad performance in confusing phonemes and poor alignment accuracy.

For confusing phonemes, several methods were proposed. Tsubota applied ASR firstly to detect possible errors and then combined Linear Discriminant Analysis (LDA) with acoustic features to further verify the errors [9]. The result showed that the verification process could detect some dissimilarities in phonemes. Similarly, Yoon employed Support Vector Machine (SVM) to do the verification process [10]. Since the SVM with a non-linear kernel can handle non-linear transforms, which is more flexible than the LDA, the accuracy of the verification process was higher than the Tsubota's. The similar phonemes can also be discriminated with a better acoustic model. In Stanley's review on the CALL, he advocated further CALL researches to focus on the acoustic model with the discriminative training [11]. Several studies followed his suggestion. For example, Yan improved the acoustic model with Minimum Phone Error (MPE) and Minimum Word Error (MWE), and elevated CALL systems with more sensitivity of similar phonemes [12]. In addition to Yan's work, Huang also proposed an F1-based discriminative training process for the CALL [13]. Because F1 is frequently regarded as an evaluation metric for mispronunciation problems, the result greatly outperformed the GOP. After the introduction of deep learning and its successful applications for ASR problem, Deep Neural Network (DNN) has been put forward for acoustic models and achieved great progress. It offered an improvement in CALL system as well. For examples, Nicolao et al. and Gao et al. applied DNN method and received considerable elevation of the CALL performance [14, 15]. Instead of focusing on discriminativeness in similar phonemes, some methods were proposed to improve CALL systems under a situation of imperfect acoustic models. A method proposed by Abdou [16] for the confusing phonemes was to generate a confidence level to detect problems which can be regarded as a compromise to the inaccurate evaluation. Weighted GOP (wGOP) is another method to improve CALL effectiveness under low discriminativeness

in similar phonemes [17]. It calculated all similar phonemes of an aligned segment and applied a linear combination of their likelihood ratio to generate wGOP for a certain phoneme.

The alignment methods varied in previous researches. Some applied traditional ASR decoding method, while others performed fixed alignment due to the speech is highly related to the template[①]. Though the second alignment seems to be easier and it is the baseline in the CALL, there are still some obstacles (re-read, word deletion, and insertion) to the expected alignment. For problems in poor alignment, Chen employed an optional silence model to identify optional silences that occur between each phoneme [18]. Since alignment error cannot be ignored, there was a method that is robust to alignment error. After comparing different models for CALL scoring, Strik found an LDA-APF (Acoustic Phonetic Feature) model that was robust to alignment error [19]. Apart from template-based models that are specific for the CALL, some related works have been done in adapting traditional ASR decoding into a more CALL-friendly process. For example, Wang et al. mapped some L1 phonemes to L2 phonemes and aligned reference text with a multi-choice Viterbi alignment [20]. To avoid massive computation, pruning in decoding was proposed in the paper as well. Since some corpora in CALL provide phonetic duration information, the alignment can be refined under supervision. Based on the dataset annotation, Chen et al. proposed a Learning to Rank (LTR) function[②] and performed alignment on the LTR results for each frame [21].

In addition to GOP methods (also named as likelihood methods), other scoring methods can be classified into classifier-based scoring, Extend Recognition Network (ERN) scoring and unsupervised error discovery scoring [22]. Classifier-base scoring often based on feature extraction from force-aligned segments in L2 learners [23, 24]. Because of the limitation in task complexity and disability in offering instructions, the methods did not draw much attention. On the other hand, ERN based scoring and unsupervised error discovery scoring work on different ways to study the relationship between L1 and L2. Since mother tongue for L2 speakers has a significant impact on their accent and acoustic feature, the mispronunciation errors in their target learning language are more likely to generate from differences between the two languages. To solve the problem, some manual works showed their improvements. Zhou et al. processed a few lists of error trends for suggestions in CALL systems based on manual summary [25]. And Wang et al. increased the CALL performance with teaching experience [26]. However, the manual work is time consuming and cannot be easily extended to two random languages. The basic idea for ERN of CALL is to construct a decoding graph within a complexity between simple template derived from reference and full decoding graph in ASR, so it can also be regarded as an improvement in alignment [27, 28]. Unsupervised error discovery is to find out possible error pattern from L2 corpus. For example, Wang et al. applied clustering algorithms with an input of universal phoneme posteriorgram to detect error patterns automatically [29]. Lee et al. inferred errors from the

① In most of the CALL problem, the reference text is often pre-given. Therefore, the exactly content on the same speech is available for CALL.

② A technique frequently applied in information retravel (IR)

similarity between spectrograms and found it robust without supports from any non-native datasets [30].

In addition to scoring algorithmic improvement, there were many related works on adaptation as well. In CALL, previous Maximum Likelihood Linear Regression (MLLR) or feature-space Maximum Likelihood Linear Regression (fMLLR) may hurt the accuracy in detecting mispronunciation problems. There are two reasons. Firstly, when training MLLR or fMLLR, it is assumed that native speaker corpus and non-native speaker corpus have the same feature space or model-level space, which is not easy to reach in a real situation [31]. Next, it was proved that MLLR adaptation on CALL problems had a side effect on scoring [32]. Some of the error pronunciation might be transformed into correct pronunciation after adaptation that is initially for speaker-level normalizing. The phenomenon named "over-adaptation". The naïve method to ease the problem is to build up two models [33, 34, 35]. One is for mother tongue of L2, the other is for the target language. The result is a combination of the output of the two models. Following similar insights, bilingual models were proposed as well [31]. Luo et al. got his inspiration from "over-adaptation" and proposed a regularized MLLR speaker adaptation with linear combination of the MLLR matrixes of teachers [36]. After splitting students' MLLR matrixes into a linear combination of teachers who were assumed to have no pronunciation mistakes, the adverse effects of over-adaptation could be reduced.

Apart from general pronunciation problems, researchers also paid much attention to specific language characteristics for better CALL system of a certain group. For example, Chinese has tone concept that is a further extension of basic phoneme recognition [37, 38, 39]. Specific points in Arabic and Dutch speeches were discussed in [40, 41] as well.

For the caption, pronunciation caption is much harder than other tasks such as image classification and speech recognition. Therefore, some researches put their concentration into annotation efficiency. Some applied algorithms to discover hidden errors with limited annotation [42, 43], while others proposed a crowdsourcing way for massive caption [44].

3. Acoustic Model

The acoustic model is the beginning of the whole CALL system. Because the erroneous output would further pass down to the following decoding and scoring processes, the accuracy of the acoustic model should be as high as possible. In this thesis, we choose a state-of-the-art model (subsampling Time Delay Neural Network with Lattice-Free Maximum Mutual Information criterion) for phonemes modeling. In the experiment, we employ a large dataset with 1000-hour L1 dataset (Librispeech) for training.

To prevent the model from learning mispronunciation errors, corpora from L2 speakers should not be included for the selection of training corpora. So, we add none of the L2 data in the training dataset for the acoustic model. With a large training dataset, we assume that the L2 speaker variation has been included in the training corpus. Another problem may occur is that the L1 corpus has no speeches from

young English learners. Comparing to speaker adaptation studies, speech analysis for young English learners has drawn relatively little attention. After surveyed 16 speech-related papers for young English learners [45 - 60], all their data on young English learners was obtained by self-collecting and none of them had posted a public-available dataset for young English learners. Therefore, we do not include speech dataset of young English learners in the training corpus. We propose further implements for this potential problem in the Chapter 5.

This chapter discussed some current methods of constructing an acoustic model. The first section introduces traditional HMM-based acoustic model including feature processing process, estimation process and some other details in HMM for ASR. Section 2 focuses on the speaker adaptation for the acoustic model. Section 3 introduces time-delay Deep Neural Networks (TDNN) under the chain structure. And the last section discusses the experiment run on Librispeech corpus, an L1 speaker corpus. The last section summarizes the whole chapter.

3.1 HMM-Based Acoustic Model

Hidden Markov Model (HMM) [61] has long been regarded as a powerful statistical method to model sequential data in discrete time series. Not only can it efficiently integrate various data sequence to hidden patterns but can also reach a unified model accompany with the dynamic programming technique. The intuitive logic of HMM is close to speech's nature by modeling a sequence with dependency relationships. The study on the HMM and acoustic models emerged at a very early age. The first HMM acoustic model was contructed in 1975 [62]. As for several decades, the method gradually becomes the most fundamental tools for training an acoustic model, which still shines for pre-training steps in state-of-the-art models.

3.1.1 Feature Processing

The original speech wave is very difficult to handle since it has only one dimension on the time domain. To convert the data into an easier version, the Fourier Transform is introduced. With Fourier Transform, waveform data can be converted into the frequency domain which has been found to be easier for further modeling. Since the speech can be easily comprehended by human beings, there are several methods motivated by behavior from the human acoustic system, such as Mel-Frequency Cepstrum Coefficients (MFCC) and Perceptual Linear Prediction (PLP) [63]. This section mainly discusses the MFCC.

MFCC is the most widely-used features applied for speech processing because of its usefulness [64]. It is a developed version of traditional cepstrum. MFCC computation consists of the following steps: 1) the signal performs pre-emphasis on its high frequency to get the same Signal to Noise Ratio (SNR) for Fourier Transform. For some toolkits like Kaldi, the signal has also a process of dithering [65]

for feature robustness. 2) the signal is windowed at a duration of 10 to 30 milliseconds[①] with a smaller shift at half or less time of a window length each frame. For each frame, a window function is proposed for better Fourier Transform[②]. 3) each frame is converted into the frequency domain by Fourier Transform, namely "spectrogram". The spectrogram is then fed into triangular filters following the Mel-frequency scale. The Mel-frequency scale is designed to fit the sensitivity of humans' hearing system. 4) the output from filter banks further takes a logarithmic form, because it has been found that human ear distinguishes absolute changes for low-frequency signals but logarithmic changes for high-frequency signals[③]. 5) a Discrete Cosine Transform (DCT) is performed to generate the final cepstrum feature (MFCC). Before feeding the features into models, MFCC always computes with cepstral mean and variance normalization (CMVN) [66]. The method performs a rudimentary step in reducing the acoustic difference between speakers and compensate for long-term spectral effects from recording tools (e.g. microphones).

3.1.2 HMM Model

In this subsection, the HMM model is introduced as well as its estimation process.

The HMM model derives from the Markov Chain model. The Markov Chain model (for this research, the discrete Markov Chain is applied) is a series of random variables. All the finite random processes can be defined as follows [63]:

Let $\mathbf{X} = X_1, X_2, X_3, \dots, X_n$ as a sequence of n random variables chosen from a finite discrete set $O = \{o_1, o_2, o_3, \dots, o_m\}$. According to the Bayes rule, we have

$$P(X_1, X_2, X_3, \dots, X_n) = P(X_1) \prod_{i=2}^{n} P(X_i \mid X_1, X_2, X_3, \dots, X_{i-1}) \tag{3-1}$$

The Markov Chain model is in first-order with the Markov assumption that

$$P(X_i \mid X_1, X_2, X_3, \dots, X_{i-1}) = P(X_i \mid X_{i-1}) \tag{3-2}$$

Therefore, for the Markov Chain, **Formula (3-1)** can be rewrite as

$$P(X_1, X_2, X_3, \dots, X_n) = P(X_1) \prod_{i=2}^{n} P(X_i \mid X_{i-1}) \tag{3-3}$$

As the Markov Chain is associated with time-invariant events, the random variable X_i can be represented by finite state s_i. Therefore, for a Markov chain with n states, the parameters of it can be summarized as follows:

$$a_{ij} = P(s_i = j \mid s_{i-1} = i) \quad 1 \le i, j \le n \tag{3-4}$$

$$\pi_i = P(s_1 = i) \quad 1 \le i \le n \tag{3-5}$$

① The reason for 10 to 30 milliseconds is that there is a consensus that speech waveform can be regarded as stationary.
② Window function helps to smooth the edge of a window-size frame. Since Fourier Transform assume the input wave is recurrent, a mismatch in the edge will do harm to further feature extraction. Mostly the window functions are chosen from Hanning window or Hamming window. For Kaldi, there is a Povey window that forces two edges of the window to be zero.
③ The threshold is found to be around 1 kHz.

where a_{ij} is the transition probability from state i to state j. And π_i is the initial probability for the start of the Markov chain. The sum of a_{ij} and the sum of π_i are both 1.

The Markov chain is powerful for building an observable sequence with limited memory cost, but the states in the Markov chain only correspond to deterministically observable output. Therefore, it cannot infer observable symbols from relevant features. To extend the modeling capacity, a non-deterministic process for each state is proposed, which also known as the Hidden Markov Model (HMM). Because of the extension, an HMM has more parameter sets as follows.

$$a_{ij} = P(s_i = j \mid s_{i-1} = i) \quad 1 \le i, j \le n \tag{3-6}$$

$$\pi_i = P(s_1 = i) \quad 1 \le i \le n \tag{3-7}$$

$$b_i(k) = P(X_i = o_k \mid s_t = i) \tag{3-8}$$

Where $b_i(k)$ is an output function that stands for the probability of emitting o_k as in state *i*. The sum of $b_i(k)$ is 1 as well. The set of a_{ij} and $b_i(k)$ can be annotated as **A** and **B**. The model can be sum up to $\Phi(\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ with parameter sets of **A, B**, and $\boldsymbol{\pi}$. A traditional method for the **B** matrix's modeling is to apply Gaussian Mixture Model (GMM) trained with Expectation Maximization (EM) algorithm. Assume the Gaussian Mixture has *M* components. Then the $b_i(k)$ is given by

$$b_i(k) = \sum_{k=1}^{M} w_{ik} b_{ik}(o_t) \tag{3-9}$$

And for each mixture component, the probability can be given by

$$b_{ik}(o_t) = \frac{1}{2\pi^{\frac{n}{2}} |C_{ik}|^{\frac{1}{2}}} e^{-\frac{1}{2}(o_t - \mu_{ik})^T C_{ik}^{-1} (o_t - \mu_{ik})} \tag{3-10}$$

where μ_{ik} denotes the mean of the mixture (n is the size of the output symbol set). w_{ik} is the weight for each mixture. C_{ik} is a covariance matrix and it is set to be diagonal assuming the elements of feature elements are independent[①].

Given an HMM, the probabilities of an output string in $\boldsymbol{O}$ within T speech frames following the state sequence $\theta = < \theta_1, \theta_2, \ldots, \theta_T >$ is

$$P(\boldsymbol{O}, \theta) = \pi_{\theta_1} \cdot b_{\theta_1}(o_1) \cdot \prod_{t=2}^{T} a_{\theta_{t-1}\theta_t} \cdot b_{\theta_t}(o_t) \tag{3-11}$$

The Viterbi Algorithm was always employed to decode the states [67]. It applies dynamic programming when scanning the HMM graph. For each timestamp, the Viterbi algorithm computes probabilities by choosing the optimum previous path. The probability of the Viterbi algorithm at time *t* is

$$V_t(i) = P(X_1^t, S_1^{t-1}, s_t = i \mid \Phi) \tag{3-12}$$

where X_1^t is the observation till time *t*, the S_1^{t-1} is the previous state sequence. For recursion part, the

① The main reason for the it is to reduce massive computation cost.

choosing criteria is

$$V_t(j) = \underset{1\le i\le N}{\text{Max}}[V_{t-1}(i)\cdot a_{ij}]b_j(X_t) \tag{3-13}$$

$$B_t(j) = \underset{1\le i\le N}{Argmax}[V_{t-1}(i)\cdot a_{ij}] \tag{3-14}$$

Baum-Welch Algorithm is for HMM parameters estimation based on EM [63]. The parameters are iteratively re-estimated, and the process is repeated until the change is accepted by a pre-defined threshold. Given a parameter set λ of $\{\mathbf{A},\mathbf{B},\boldsymbol{\pi}\}$, and ϕ as the likelihood function, the target function for Baum-Welch Algorithm can be written as

$$Q(\lambda,\widehat{\lambda}) = \sum_{\theta}\phi(\theta \mid \boldsymbol{O},\boldsymbol{A},\boldsymbol{B})\log(\phi(\theta,\boldsymbol{O}\mid\widehat{\boldsymbol{A}},\widehat{\boldsymbol{B}})) \tag{3-15}$$

Since the output sequences are modeled with GMM, **Formula (3-15)** can be written as

$$Q(\lambda,\widehat{\lambda}) = c-\frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_m(t)\left(c_m+\log\left(\left|\widehat{\Sigma_m}\right|+(o_t-\widehat{\mu_m})^T\widehat{\Sigma_m^{-1}}(o_t-\widehat{\mu_m})\right)\right) \tag{3-16}$$

where M is the number of Gaussian Mixture components. c_m and c are constants in respect to λ. The $\gamma_m(t)$ denotes the probability of the state in mth mixture component at time t.

3.1.3 HMM for Acoustic Modeling

The HMM acts as a core in speech processing, but it cannot apply to the acoustic modeling directly. Before the HMM training process, several preparations need to be done (including the triphone model, decision trees for context clustering). In this section, we discuss specific factors for acoustic modeling with HMM.

For acoustic modeling, the basic unit is a phoneme. Instead of words that are various, phonemes are accurate, trainable and generalizable for acoustic modeling [63][①]. But a mono-phone model (consider each phoneme an individual unit) cannot model the context dependency problem given the Markov assumption. Consequently, the context dependency model was proposed with the triphone [68]. A triphone model considers neighboring phones as a unit other than a single phone. In addition, stress also affects the phonetic feature. Researches have a consensus that stressed phonemes tend to have higher pitches, longer duration and more energy comparing to unstressed ones [69]. Therefore, vowels are divided into stressed, unstressed, secondary stressed in the system as well.

Though triphone has a significant effect on the acoustic model, the computation and memory costs are at exponential expenses. Another point is that triphone assumes that the triphone's context is different from each other. However, there are similarities between the effect of neighboring phonemes. Therefore, Huang proposed clustering methods based on linguistic questions [70]. Through asking questions on linguistic features (such as nasal, sonorant or voiced), triphones are compressed into clusters with

① There are only less than 50 phonemes under different criteria, while the words are countless.

decision tree. The cluster from tri-phonetic events names senone [71]. A simple structure of HMM with senone is defined in **Figure 3-1**. There are two sub-HMM models. After clustering, the first two phonemes of both HMM are clustered to the same senone while the last two are different because of changes in context. By adjusting the decision tree for clustering, the number of senones can be modified to a specific range (around thousands) where allow us to balance the efficiency and accuracy.

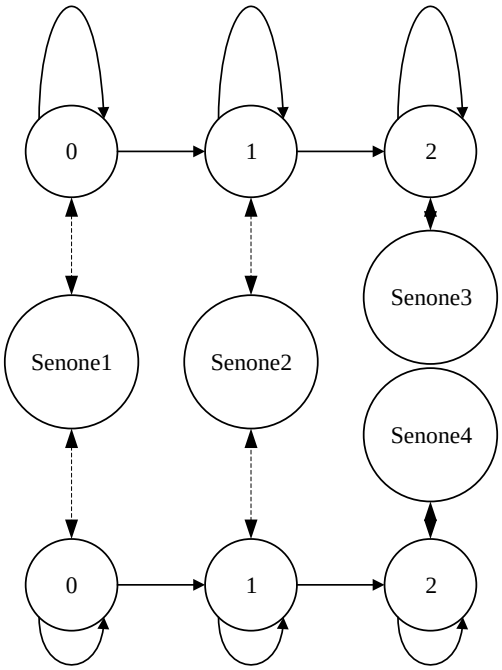


Figure 3-1 An HMM Structure with Senones

After determining the topology for HMMs, the acoustic model is ready for acoustic recognition. The phoneme recognition process can be summarized as following:

- Compute MFCC features for each frame.
- Decode the HMM with MFCC (the MFCC series are observable symbols of the HMM).
- Output the posteriorgram of phonemes' sequence (HMM's hidden states).

3.2 Speaker Adaptation

According to previous literature on CALL and speech processing for young English speakers, speaker adaptation techniques are always implemented as an important way. For CALL problem, the speaker adaptation has risks in over-adaptation as discussed in [31, 32, 33, 34, 35, 36], especially when training with non-native speech corpora. While for speech recognition of young English learners, speaker adaptation techniques are the main contribution to the recognition improvement such as MLLR, fMLLR, Vocal Tract Length Normalization (VTLN), and Maximum a Posteriori (MAP) [46, 51, 52, 53, 55, 57, 58, 59]. This section introduces methods applied in the following experimental section.

MLLR is a baseline adaptation model that introduced in [72] and developed in [73]. It re-estimates

GMM's parameters with linear transformation given a speaker independent acoustic model. There are two forms of MLLR, known as unconstrained MLLR and constrained MLLR (cMLLR, also known as fMLLR). And both of their estimation processes apply the EM algorithm. Experiments had shown that there is not a superior method comparing MLLR and fMLLR [74]. However, for large speech corpora, fMLLR simplifies the training process and thus has a better runtime performance.

When taking the speaker effect for speakers $\boldsymbol{R} = < s_1, \dots, s_R >$ into the HMM training process, the $\widehat{\mu_m}$ and $\widehat{\Sigma_m}$, are

$$\widehat{\mu_m} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{rm}(t)o_{rt}}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{rm}(t)} \tag{3-17}$$

$$\widehat{\Sigma_m} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{rm}(t)(o_{rt} - \widehat{\mu_m})(o_{rt} - \widehat{\mu_m})^T}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_{rm}(t)} \tag{3-18}$$

where $\gamma_{rm}(t)$ denotes the speaker r's probability of the state at time t for GMM mixture m.

For MLLR, $\widehat{\mu_m}$ and $\widehat{\Sigma_m}$ are adapted as **Formula (3-19)** and **Formula (3-20)**:

$$\hat{\mu} = \boldsymbol{W}\mu + \boldsymbol{b} \tag{3-19}$$

$$\hat{\Sigma} = \boldsymbol{B}\Sigma\boldsymbol{B}^T \tag{3-20}$$

For fMLLR, $\widehat{\mu_m}$ and $\widehat{\Sigma_m}$ are changed as **Formula (3-21)** and **Formula (3-22)** where the transformation matrix is constrained as the same $\boldsymbol{A}$:

$$\hat{\mu} = \boldsymbol{A}\mu + \boldsymbol{b} \tag{3-21}$$

$$\hat{\Sigma} = \boldsymbol{A}\Sigma\boldsymbol{A}^T \tag{3-22}$$

As **Formula (3-21)** and **Formula (3-22)** are substituted into **Formula (3-16)**, yielding:

$$Q(\lambda, \widehat{\lambda}) = c - \frac{1}{2}\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_m(t)\left(c_m + \log\left(\left|\widehat{\Sigma_m}\right| + (\widehat{o_t} - \widehat{\mu_m})^T\widehat{\Sigma_m^{-1}}(\widehat{o_t} - \widehat{\mu_m})\right)\right) \tag{3-23}$$

$$\widehat{o_t} = \boldsymbol{A^{-1}}o_t + \boldsymbol{A^{-1}b} \tag{3-24}$$

Formula (3-23) indicates that fMLLR only conducts feature space transform and leaves the model parameters still. A detailed mathematical proves can be found in [73].

Speaker Adaptive Training (SAT) can adapt to speaker variations based on the fMLLR. It focuses on speaker-dependent transforms. Another advantage of fMLLR is its efficiency in Speaker Adaptation Training (SAT). Comparing to MLLR, fMLLR can be fitted into SAT procedures with minimum changes [73].

The fMLLR is effective for the GMM models, but it is not useable to the DNN structure. In the GMM, means and variances have statistical meanings and can be transformed together within the model. Unlike the GMM's parameters, the weight factors in DNN have no well-formed structure for the linear transformation. Therefore, when discussing the DNN methods, traditional fMLLR cannot work. There are researches that studied similar strategies as fMLLR in a DNN structure [76], but it is trained under Cross Entropy criterion other than Maximum Likelihood. Another substitute method is to apply fMLLR that estimates with GMM-HMMs to get speaker adapted features. However, it has to be admitted that fMLLR are trained assuming with the GMM-HMMs other than DNN-HMM. The adapted fMLLR features are not sure to be suitable for the DNN. To put forward a "DNN-like" speaker adaptation method,

Saon et al. proposed an I-vector method that extracts speaker information through EM processes [77]. I-vector method is a popular technique for studies in speaker recognition or verification, which aims to find a linear dependence from Universal Background Model (UBM)[①] to speaker dependent distribution. The estimated I-vectors are cascaded after basic MFCC features in the DNN input feature.

3.3 Time Delay Neural Networks

Neural networks have proved to be a powerful tool for several speech tasks. This section firstly discusses how Deep Neural Networks are implemented in acoustic modeling. Then, the base of Time Delay Neural Networks is introduced. At last of the section, the Chain Structure is discussed, which applies Lattice-Free Maximum Mutual Information (LF-MMI) discriminative training.

A basic framework of DNN-HMM work is shown in **Figure 3-2**. Based on the senones and the deep structure, it can substitute the GMM part in the traditional architecture which brings much improvement with a minimum-modified decoding process [78].

Time Delay Neural Networks (TDNN) is a context-dependent neural network for speech processing and phoneme recognition [79]. It is regarded as a pioneer of Convolutional Neural Networks (CNN). The basic structure of the TDNN is shown in **Figure 3-3**.

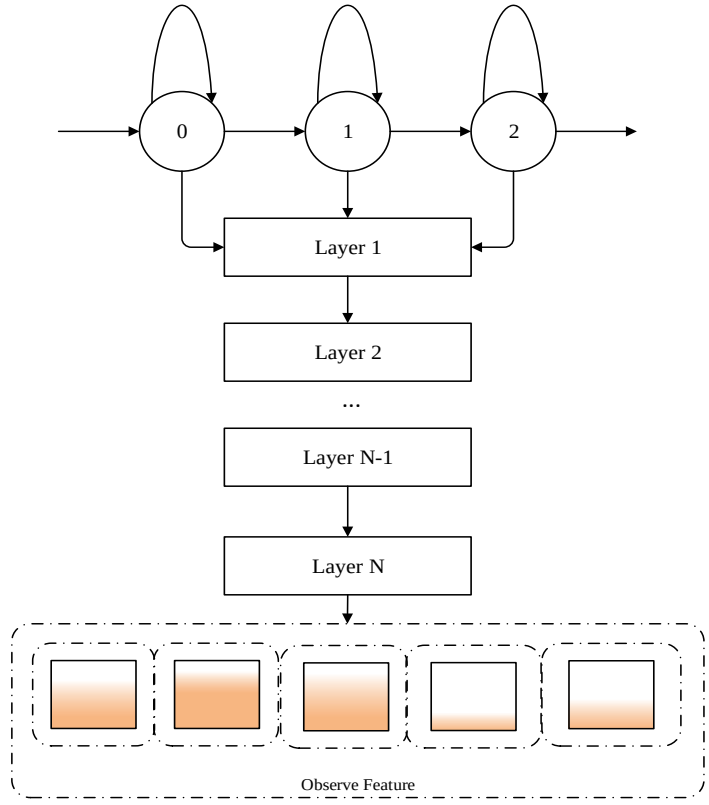


Figure 3-2 A DNN-HMM Structure

[①] The UBM is a GMM trained with speaker independent audio wave. Therefore, it can be considered as speaker independent and a useful verification tool to verify whether the feature is speaker dependent.

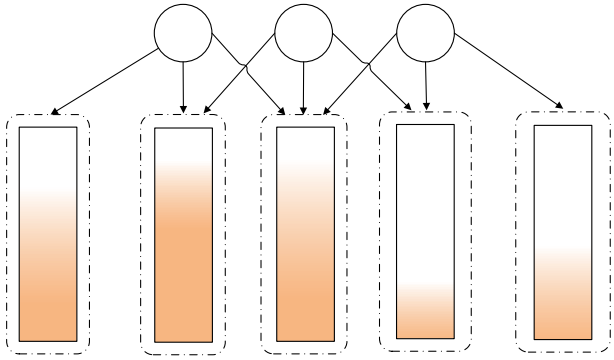


Figure 3-3 A Traditional TDNN Structure

Figure 3-3 shows the input layer of the TDNN. Circles represent filters before the network and the orange blocks stand for the input feature frames. The filters of the TDNN only accept input from partial three features and they share the same weights for all parameters. The outputs from filters can then be fed into hidden layers to hybrid the TDNN with DNN-HMM model [80]. The TDNN has desirable properties that it can accommodate to the temporal context between successive acoustic events and it is time invariant. The traditional TDNN excels in the ASR performance comparing to other neural networks until the prevailing of deep Recurrent Neural Networks (RNN) [81]. The RNN (mostly Bidirectional Long Short-Term Memory, BiLSTM) applies a Connectionist Temporal Classification (CTC) loss to form an end-to-end training without alignment [81, 82, 83]. The framework is promising with Large Vocabulary Continuous Speech Recognition (LVCSR) that contains thousands of hours. However, the training process is very slow for its disability in parallelization. To solve the problem, Peddinti et al. proposed a new TDNN framework with subsampling [84]. For the traditional TDNN, considering a context window of 5, each feature will be scanned fifth except for the start and the end. Based on the assumption that neighboring activations are continuous, the subsampling TDNN accepted gaps between each frame as shown in **Figure 3-4**. During the recognition process, the subsampled TDNN also applies asymmetric input contexts in high layers with more stress on the left following empirical tests. Subsampling significantly reduces runtime to 5 times less compared to the traditional TDNN and returns the model with a much smaller size. And the model remains superior to DNN models and even unfolded RNN models.

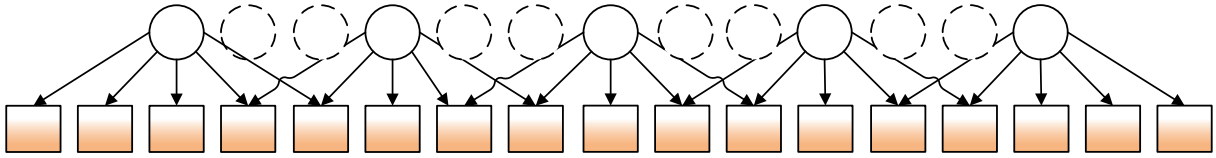


Figure 3-4 A TDNN Structure with Subsampling

In addition to the problem in training speed, the CTC methods were found to be unsuccessful with hundreds of hours dataset, while the discriminative training methods for DNN-HMM are superior [85]. As for the DNN-HMM system above, Cross Entropy (CE) loss function is often employed to minimize the phonemes prediction error rate. The CE criterion evaluates each speech frame independently. Hence, the training process ignores the context information among phonemes series. To address the error,

discriminative training methods for DNN were proposed. The discriminative criteria for DNN include Maximum Mutual Information (MMI), Minimum Phone Error (MPE), boosted MMI (BMMI) and Minimum Bayesian Risk (MBR). According to the report from [86], the introduction of the criteria can offer an improvement of 1.5% to 2% comparing to DNN with CE loss. Traditional discriminative training processes require lattices generated from a preliminary model such as GMM-HMMs and DNN-HMMs with CE loss [87]. The lattices are used to provide a simple approximation for possible phoneme sequences (or word sequences) which can limit the computation cost to a controllable range. A typical MMI loss can be computed as follows.

The $\boldsymbol{o}^m = < \boldsymbol{o}_1^m, \boldsymbol{o}_2^m, \dots, \boldsymbol{o}_{T_m}^m >$ is defined as the observed sequence of the m[th] speech where T_m is the frame number, and the $\boldsymbol{w}^m = < \boldsymbol{w}_1^m, \boldsymbol{w}_2^m, \dots, \boldsymbol{w}_{N_m}^m >$ is defined as words' caption of the m[th] speech where N_m is the number of words. For the whole training set that has M speech samples (denoted as $\mathbf{S}$), the MMI is

$$\begin{aligned} \mathcal{L}_{MMI}(\theta; \boldsymbol{S}) &= \sum_{m=1}^{M} \mathcal{L}_{MMI}(\theta;\ \boldsymbol{o}^m, \boldsymbol{w}^m) \\ &= \sum_{m=1}^{M} logP(\boldsymbol{w}^m \mid \theta;\ \boldsymbol{o}^m) \\ &= \sum_{m=1}^{M} log(\frac{p(\boldsymbol{o}^m|\boldsymbol{s}^m;\ \theta)^{\kappa} P(\boldsymbol{w}^m)}{\sum_{\boldsymbol{w}} p(\boldsymbol{o}^m|\boldsymbol{s}^m;\ \theta)^{\kappa} P(\boldsymbol{w})}) \end{aligned} \tag{3-25}$$

where θ is the parameter set for the model (i.e. DNN) and $\boldsymbol{s}^m$ is for states in the HMM. κ is a hyperparameter as the acoustic scaling factor.

According to chain rules, the derivative of $\mathcal{L}_{MMI}(\theta; \boldsymbol{S})$ is as **Formula (3-26)**:

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{MMI}(\theta; \boldsymbol{S}) &= \sum_{m=1}^{M} \sum_{t=1}^{T} \nabla_{z_{mt}} \mathcal{L}_{MMI}(\theta; \boldsymbol{S}) \cdot \frac{\partial z_{mt}}{\partial \theta} \\ &= \sum_{m=1}^{M} \sum_{t=1}^{T} \left(\kappa \left(\frac{\alpha_t^{num}(\boldsymbol{r}) \beta_t^{num}(\boldsymbol{r})}{\sum_r^N \alpha_t^{num}(\boldsymbol{r})} - \frac{\alpha_t^{de}(\boldsymbol{r}) \beta_t^{de}(\boldsymbol{r})}{\sum_r^N \alpha_t^{de}(\boldsymbol{r})} \right) \right) \cdot \frac{\partial z_{mt}}{\partial \theta} \end{aligned} \tag{3-26}$$

where z_{mt} is the input of the final Softmax layer. The computation of $\frac{\partial z_{mt}}{\partial \theta}$ is the same as CE. $\boldsymbol{r}$ represents the state sequence. $\frac{\alpha_t^{num}(\boldsymbol{r}) \beta_t^{num}(\boldsymbol{r})}{\sum_r^N \alpha_t^{num}(\boldsymbol{r})}$ denotes the posterior probability vector for $\boldsymbol{r}$[①]. It is computed via a forward-backward algorithm (like the Baum-Welch algorithm discussed above) on the numerator lattice graph and the denominator lattice graph. The traditional MMI discriminate training must apply the lattice, otherwise, the computation cost is huge. However, the lattice also introduces some losses in accuracy and it is still time-consuming. The framework is shown in **Figure 3-5**.

① The Formula 26 is simplified for interpretation. A full version can be found in [87].

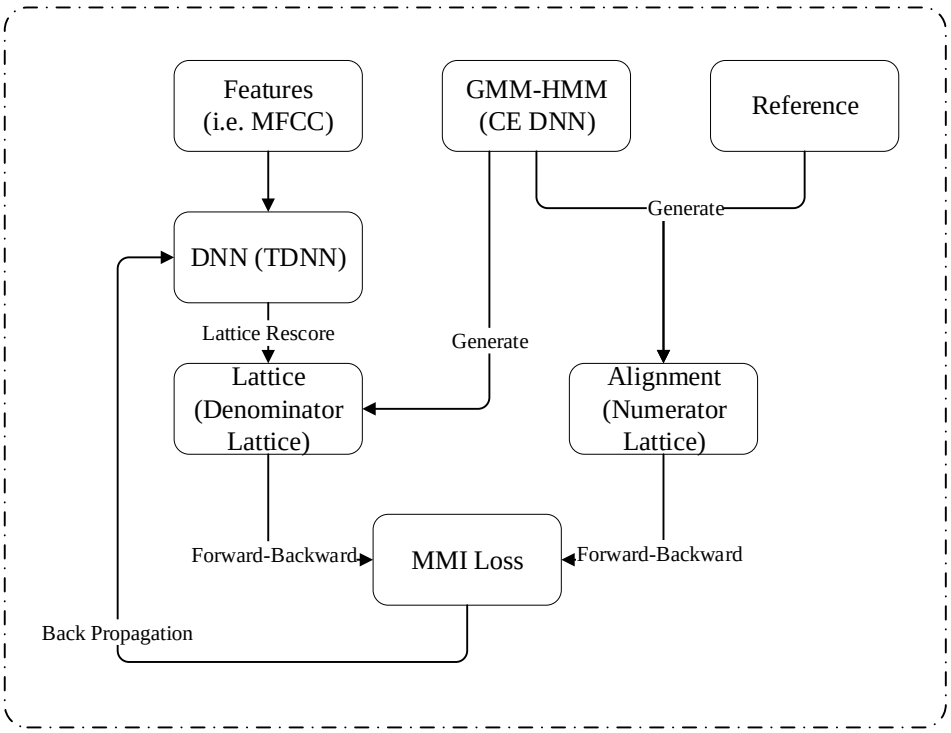


Figure 3-5 The MMI Training Process

The Lattice-Free MMI (LF-MMI) was proposed in [85] to solve the problem[①]. It applies a phone-level n-gram language model for denominator graph with a more efficient minimization algorithm under non-deterministic Finite State Acceptor (FSA). The simplification of the denominator graph enables the forward-backward algorithm in the whole graph rather than confined graph from the lattice. The alignment is still necessary in order to form the numerator graph. And the numerator graph is also split to chunks with time constraints and subsampling. Since the sequence-level training has a problem of overfitting [88], three regularization techniques are also proposed including a multi-task on CE, an L2 regularization and Leaky HMMs (stop and restart a new HMM when encountering certain probability on each frame). In addition, in the Chain model, the model unit was bi-phone instead of triphone, which can further faster the training process with an even better recognition result. The L2 regularization for hidden layers is further extended to the TDNN-F (factorized forms for tradition TDNNs). The extension solved the stability problems which frequently occurred in traditional Singular Value Decomposition (SVD) based fine-tuning techniques for neural network training [89].

The Chain model is a combination of subsampling TDNN-F and LF-MMI with an integrated frame size (from 10ms to 30ms) and an unconventional HMM topologies that changes the fixed 3-state structure to a single state. The chain model is implemented in Kaldi [65].

3.4 Experiments on Librispeech

This section introduces experiments done on Librispeech with the Chain structure model discussed above. The acoustic model is further applied to form the relating part within the CALL system.

① The methods only avoid the usage of denominator lattice and the alignment (numerator lattice) is still needed.

Librispeech is a corpus of English Reading speeches [90]. It contains almost 1000 hours (960 hours for training and 40 hours for testing) of reading speeches at a sampling rate of 16 kHz and it is also gender balanced at both the speaker level and the utterance level. Among public-available datasets, Librispeech is the largest dataset at the best of our knowledge. In addition, the corpus is based on audiobook which is the same task (the speaking task) for the CALL system. The corpus has 7 partitions as listed in **Table 3-1**.

Table 3-1 Corpus Partitions in Librispeech

Partitions	Hours	Minutes per speaker	Female Speakers	Male-speaker	Total Speakers
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166
test-clean	5.4	8	20	20	40
test-other	5.1	10	17	16	33
dev-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33

Most of the speech corpus has no phoneme-level caption, neither does the Librispeech dataset. Therefore, we evaluated the acoustic training process with the Word Error Rate (WER) from the ASR task. and pre-built Language Models (LM) from Librispeech is applied for LM rescoring. As discussed above, a GMM-HMM model should be first trained. The corpus is gradually added into the training pool.

In the step of feature extraction, the MFCC is computed with a conventional parameter set (25 milliseconds frame length, 10 milliseconds frameshift, 13 cepstral bins, 23 Mel bins, 0.97 pre-emphasis coefficient, sample frequency at 16,000, with energy usage and adaptation through Vocal Track Length Normalization (VTLN)).

To commence for the training process, a mono-phone system is trained with the 2,000 shortest utterances from the train-clean-100 partition for easier alignment. Combined with a small trigram LM, the mono-phone model reaches a WER of 43%. Next, a triphone GMM-HMM is trained with 5,000 utterances' delta and delta-delta MFCC. The state number for the HMM is limited to 2,000 and each state will have a GMM with 5 components. Similar to this kind of process, the GMM-HMM is further trained by gradually adding more data and employing speaker adaptation methods such as fMLLR and SAT.

The final TDNN trained on LF-MMI applies the recipe of release version in Kaldi [65]. The input of the neural network is a combination of 100-dimension i-vector and the 40-dimension MFCC. It has 17 hidden TDNN-F layers besides an LDA affine layer next to the input and a linear layer next to the

output. There are three separate layers in a TDNN-F layer: a linear layer for factorization (160 dimensions), a central layer (which is a traditional TDNN layer with 1536 dimensions) and a scaling layer [89]. The time strip unit for subsampling is from 1 to 3 which relates to TDNN layers from the up to the bottom. Multi-task Learning is implemented with two losses, the LF-MMI loss, and the CE loss. The L2 regularization coefficient is 0.008 for most layers except for the last layer's 0.002. There are four epochs set in the training process. The test result of several experiments is reported in **Table 3-2**.

Table 3-2 WER of Experiments on Librispeech

	Test Clean Set	**Test Other Set**
Mono-Phone (2k)	43.43%	73.94%
Triphone (5k)	16.13%	48.91%
Triphone (10k + LDA + MLLT)	13.20%	44.70%
Triphone (10k + LDA + MLLT + SAT)	10.89%	35.11%
Triphone (100h + LDA + MLLT + SAT)	9.01%	30.28%
Triphone (460h + fMLLR + SAT)	8.11%	27.20%
Triphone (960h + fMLLR + SAT)	7.77%	21.90%
Kaldi -Reported (TDNN + LF-LMM) [65]	4.17%	10.62%
TDNN + LF-MMI	3.81%	8.80%

As shown from the results in **Table 3-2**, the TDNN structure with LF-MMI has greatly improved the traditional GMM-HMM structure with speaker adaptation. Not only performs the method perfect on the clean dataset, but it also proves to be robust in various environments. There is an improvement observed from the Kaldi-Reported training result with TDNN + LF-LMM. After checking the source codes, it is found that the Kaldi-Reported Result only applied TDNN structure instead of TDNN-F. The improvement, therefore, is because of the introduction of TDNN-F.

3.5 Summary

In this chapter, we chronological review the current techniques in the acoustic modeling, from the basic GMM-HMM model to the TDNN-HMM model with LF-MMI.

As the base for all the acoustic model, the first section introduces the traditional GMM-HMM model and how its working mechanism in acoustic modeling.

The next section drills into speaker adaptation which plays an important role in children speech processing.

The third section introduces the state-of-the-art TDNN method. Instead of the traditional one, the TDNN is subsampling for efficiency and wider context. In addition, the training process applies LF-

MMI training criteria with more accuracy improvement.

To test the result, we further conduct experiments on Librispeech. The experiment shows that the method not only significantly excels in the clean dataset, but also keeps the robustness in speeches with noise. For the excellence of the TDNN-HMM structure, it is applied as the base acoustic model in the following chapters.

4. Decoding Model

This section introduces the decoding model that decodes phonemes' posteriorgrams into phoneme sequences (it can be word sequences as well). Most of the speech corpora are without phoneme caption and the phoneme caption is unrealistic in CALL applications. The alignment is necessary for specific phonetic evaluation. Therefore, a CALL system must have a strong decoding model.

For the beginning, we introduce a subsection of forced alignment in which comparing the differences between CALL and ASR and providing a specific alignment technique for CALL. In the next part, we evaluate the alignment with the TIMIT dataset. The TIMIT has a phone-level annotation which can be applied for alignment evaluation. To evaluate the model, we employ the acoustic model trained on Librispeech (discussed in the prior section). The last section summarizes the whole chapter.

4.1 Forced Alignment

The Forced Alignment is to align a series of data to another sequential data, which is frequently applied in speech processing and analysis. The technique is essential to building up the training set for several tasks including Text-to-Speech (TTS) and multimedia web searching [91, 92]. Unlike the ASR's decoding process, the Forced Alignment is based on determined transcripts.

This section is divided into two subsections. Though the CALL is very similar to the ASR, there are still several differences according to previous discussions, especially for the decoding process. Therefore, a detailed comparison between the CALL and the ASR is discussed in the first subsection. The second subsection introduces the method applied in the CALL system for the thesis.

4.1.1 CALL and ASR

The CALL and the ASR has a very close relationship in their structure with a slightly different purpose. The purpose of the CALL is discussed before, which is to teach non-native speakers with pronunciation error detection and evaluation. While the ASR's purpose is to recognize speech (mostly focus on the robustness, such as the far-field ASR and the ASR in noisy environments). In a word, the CALL is a discriminator while the ASR is a recognizer. The following part firstly revisits the similarities of the two systems and then heads into the differences.

Basically, the acoustic models for phonetic modeling are the same in the CALL and the ASR. Both need to perform phonetic modeling to identify phonetic information from the raw audio wave. Because the studies on CALL is relatively fewer than the ASR's, the acoustic model of the CALL benefits much from the abundant studies on the ASR. Though the CALL concentrates more on phonetic classification other than sequential recognition, there remains a limitation of the speech datasets. Comparing to a large amount of data with references, the phone-level caption corpora are in shortage (it is difficult to caption these corpora since the caption needs to be conducted with professional knowledge). Therefore, direct phoneme classification cannot be trained with a large corpus. A compromising way[①] from the ASR is to link the data with word-level references and to EM the phonetic alignment with gradually training processes. Consequently, the CALL always adopts the same acoustic training structure as the ASR.

The mismatch purposes in the CALL and the ASR result in three other differences.

For starters, the attitude of two systems towards non-native speeches varies especially for the training process of the acoustic model. The CALL attempts to distinguish non-native speeches from the native ones. But the ASR accommodates to all speeches regardless of the non-natives and the natives. Its aim is to successfully recognize correct sentences even if there are some mistakes. Therefore, the acoustic training set for the CALL is often with pure native speakers[②].

For decoding, an ASR system mainly works on searching the best sequence that best fits the speech signals while the CALL works on an alignment that aligns speech signals to the reference text. The searching process in the ASR is not alone with acoustic models (HMM-based models)[③] but with LM. The LM is added to offer word-level penalties such as extra insertion or deletion. Due to the complexity of searching processes for a large lexicon, clever pruning methods are often adopted such as beam search, bi-directional search, and some heuristic searching techniques. The alignment for CALL is much simple. Since the reference is available, the target can be a phonetic sequence. What need to consider is the phone-level insertion or deletion (optional silence is important for the CALL as well). Because the graph size is much smaller than the ASR's, the pruning strictness can be also looser than the ASR's which can receive improvement on accuracy.

The CALL has another scoring part where the ASR does not need.

Though the above discussions reveal some differences in most of the CALL and the ASR systems. they are the same in some real-world applications [24]. In [24], the evaluation was performed on comparing the differences between the ASR outputs and the references. This system is preliminary for its coarse-grained evaluation on word-level.

① The only way up until now.
② Sometimes, a small portion of non-native speakers is added as well.
③ As discussed before, the CTC-based RNN methods are not considered in phonetic alignment of the CALL. Therefore, only the HMM-based acoustic model is considered.

4.1.2 Alignment for CALL

This section introduces the basics of forced alignment and its implements on the CALL. As introduced in the previous section, forced alignment for speech aims to align a series of phonemes with a raw audio wave.

An initial method is Dynamic Time Warping (DTW) [93]. The DTW is first proposed as a speech recognizing model for continuous or isolated human word recognition. It uses a dynamic programming method to align the time series of features (e.g. MFCC) and a specific word template in order to minimize the distance across the whole alignment. A sample for the word "helpful" is shown in **Figure 4-1**. The integer in each colorful block stands for the frame size aligned to target templates.

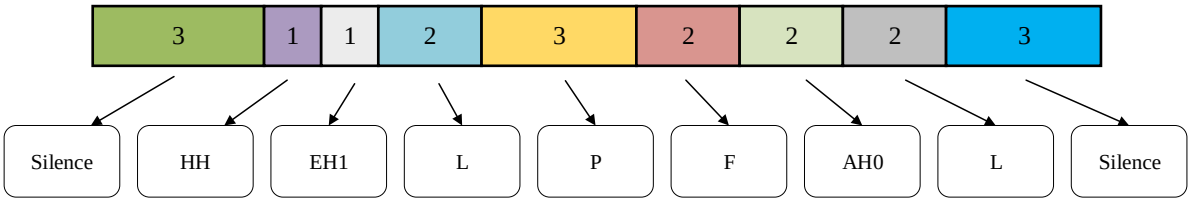


Figure 4-1 An Alignment Sample for the Word "Helpful"

The warping process is as follows:

$S = < s_1, s_2, s_3, \ldots, s_n >$ is denoted as the feature series on time and $T = < t_1, t_2, t_3, \ldots, t_n >$ is denoted as the desired templates. As arranged the S and the T to a Cartesian coordinate system, the grid point (i, j) represents an alignment on s_i and t_j. The warping path $P = < p_1, p_2, p_3, \ldots, p_n >$ is a sequence of grid points under the algorithmic restrictions. An example of the DTW is shown in **Figure 4-2**. In the example, s_1 corresponds to t_1 while s_2 is still aligned to t_1. As shown in the figure, the problem can be solved through a dynamic programming problem.

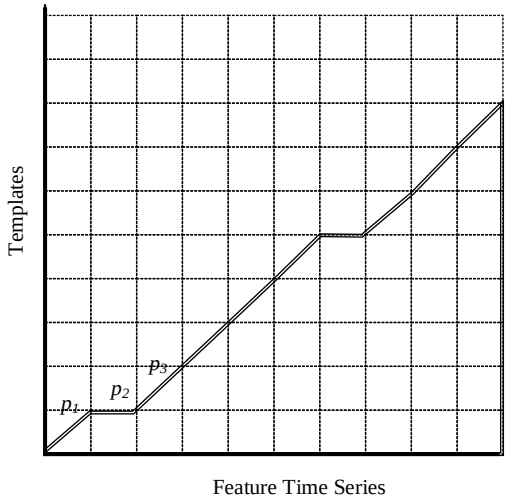


Figure 4-2 The DTW Grid Example

To perform the dynamic programming, there are three main problems to work out including the distance function and the recurrent function.

The distance function is computed for the feature time series. Several distance measures are possible for the function, the most frequent method applied is the square of the difference. If the function is denoted as δ, the objective function is to minimize the total difference as shown in **Formula (4-1)**.

$$DTW(S,T) = min(\sum_{k=1}^{K} \delta(p_k)) \tag{4-1}$$

The recurrent function is the core of the dynamic programming. A typical recurrent function is defined in **Formula (4-2).**

$$c(i,j) = \delta(i,j) + min[c(i-1,j), c(i-1,j-1), c(i,j-1)] \tag{4-2}$$

where $c(i,j)$ represents a cumulative distance that sums up the previous optimum distances. The **Formula (4-2)** is a symmetric mode for dynamic programming that equally treats the feature sequence and the template sequence. An asymmetric formulation applies $c(i-1,j)$ or $c(i,j-1)$ instead of both. For the ASR on words, the asymmetric formulation performs better [94].

The distance function and the recurrent function determines two major restrictions for the DTW. Firstly, the path sequence of grid points must follow a monotonical order which means that $i_{k-1} \leq i_k$ and $j_{k-1} \leq j_k$. Next, each step cannot skip frame on both the feature sequence and the template sequence, which means that $i_k - i_{k-1} \leq 1$ and $j_k - j_{k-1} \leq 1$.

The DTW for CALL decoding is easy. Firstly, the posteriorgram of the CALL is computed and it is the feature sequence. Then, the speech sentences are extended into phoneme sequences. The alignment is performed with the two sequences. Mostly, an asymmetric formulation is implemented. A difference from the original DTW is the changes of the distance function and the target function. For the grid point (i,j), the $(t_j)^{th}$ post probability of the s_i feature is the distance. And the objective function becomes

$$DTW(S,T) = max(\sum_{k=1}^{K} \delta(p_k)) \tag{4-3}$$

For a CALL decoding problem, the DTW has several shortcomings. The most essential one among them is that the DTW cannot model optional silence. It is natural to have optional silence between words or in the words (there is often a silence before plosive phonemes). However, the DTW cannot efficiently skip chosen phonemes. To solve the problem, an optional silence implementation based on the Viterbi algorithm [95] is applied. Before applying the Viterbi algorithm, a decoding graph is needed. Taking the "helpful" example again. If there is an optional silence occurred between the "L" phoneme and the "P" phonemes, the graph can be regarded as following **Figure 4-3**.

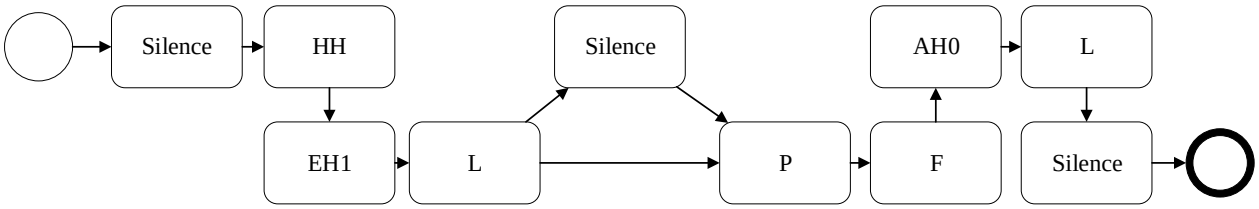


Figure 4-3 The Decoding Graph for "Helpful"

To represents the optional silence in "helpful, a silence unit is added between the "L" and the "P". In addition, an epsilon arc is added as well for the direct skip. Based on the features of posteriorgram, the Viterbi algorithm is suitable for alignments.

The graph construction can be more complicated with respect to more alignment requirements. For example, it can model the differences of the English accent and the American accent and choose the best route to distinguish the accents. Besides, the graph can also enlarge for word repetition (the situation is common for young English learners), word insertion and word deletion.

The graph method can solve several problems in optional phonetic problems, but there remain some disadvantages for the method. Firstly, the graph may be a huge cyclic graph which violates the restriction of monotonical processing. The violations extremely slow down the programming process. In addition, word level deletion and insertion may encounter errors when there are similar phoneme patterns in a sentence template. An example is shown in **Figure 4-4**. There are two words: "data" and "dates". They have the same three phoneme subsequence ("D", "EY1", "T") and different phonemes at the last ("AH0" and "T"). Because of the similarity, it will be confusing for the system whether the "dates" is repeatedly spoken or the whole "data dates" are spoken. Therefore, we only consider optional silence between words using the graph model and the Viterbi algorithm.

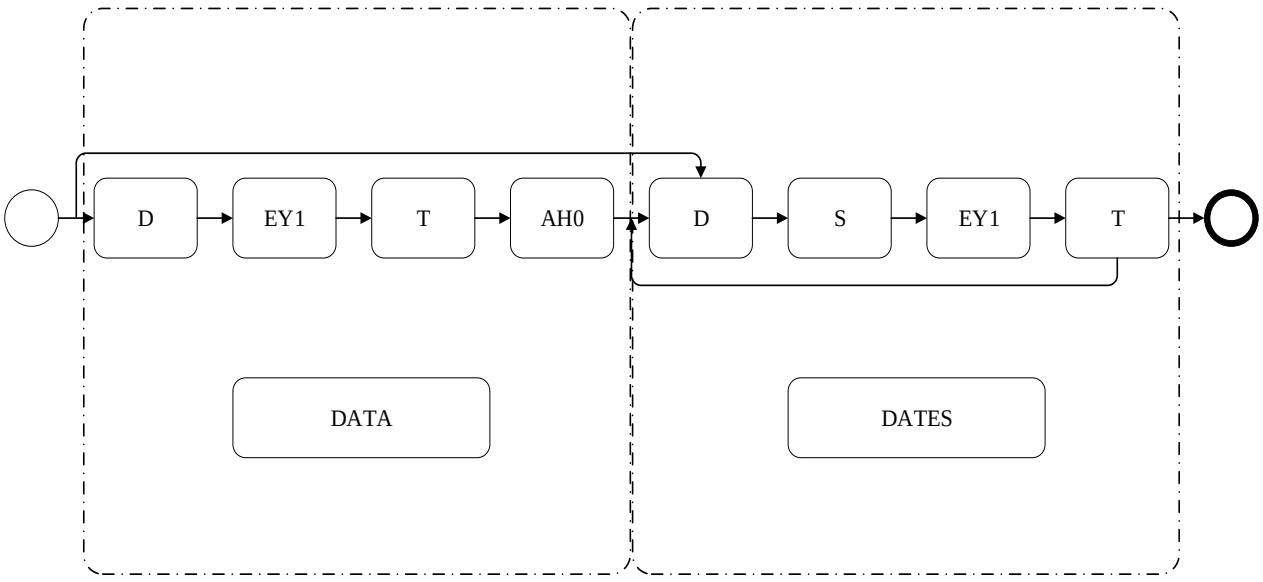


Figure 4-4 An Error Example for the Graph-based Decoding

4.2 Alignment Evaluation on TIMIT

The TIMIT (Texas Instruments and Massachusetts Institute of Technology) dataset is a pioneer phone-level database for speech analysis [96]. Up till now, the TIMIT is relatively small comparing to various corpora with thousands of hours on LVCSR. However, due to its fine-grained transcription, the TIMIT still receives much attention in the academic and industrial fields. There are 6300 sentences recorded at 20kHz (then down sampled to 16kHz) from 630 speakers (10 sentences for each) in the TIMIT database. 439 speakers (approximately 70%) are male while the others are female.

The outstanding transcription was obtained by three steps. Firstly, a phonetician recognized the acoustic-phonetic sequences by the means of repeating listening and visual examination. Next, the phonemes were aligned to the speech wave with a dynamic programming method (the method is like the DTW). The automatic alignment has three steps as well. It first classified a 5ms frame window into five categories (obstruent, sonorant, voice-sonorant, nasal and silence). Then sequences of these categories were aligned to the given phonetic sequence combining a search strategy and some phonetic rules. Some phonetic contextual knowledge is applied for further segmentation. For the last step of the transcription, each phonetic boundary was verified and refined by more experienced phoneticians.

Our alignment evaluation applies the whole set of the TIMIT (6300 sentences). On a 30ms frame size, the posteriorgram is computed by the acoustic model trained with LF-MMI and TDNN-F. Due to the different annotations for the Librispeech (that we adopted) and the TIMIT, several preprocessing works should be done before conducting the experiment.

Firstly, because the TIMIT phone-level caption is on a unit of 0.0625 ($\frac{1}{16}$) ms, a 30ms frame size is equal to 480 units of the caption in the TIMIT. After integration, both the forced alignment results and the TIMIT captions are transformed into two sequences. The first sequence is for the phoneme sequence while the next sequence has the same length where denotes the duration of the corresponding phonemes. Next, the Librispeech adopts the ArpaBet phonetic transcription while the TIMIT has its own transcription standard. Therefore, a dictionary is applied for mapping[①]. Additionally, the Librispeech system considers phonetic stress which means that each vowel has four states for its stress status. They are also integrated for the comparison. In addition, all the silence frames are cleaned out. There are two reasons for the split. For the first reason is that the forced alignment model only considers the optional silence between words, but the TIMIT caption has optional silences in the words as well. Cleaning out the silence phonemes makes it possible for comparison under the same restrictions. For the second reason, the speech length is fixed. If there are more silence frames detected, there must be some deletion in other phonemes' duration. Consequently, the align error is counted twice.

① The mapping is not explicit since the phonemes are not ono-to-one correspondent to each other.

After the preprocessing, the TIMIT caption still cannot directly compare to the forced alignment because there are phonetic insertion and deletion occurred in the TIMIT caption. A rudimental method is to only compute phonemes from the head and the tail where the phonemes are corresponding to each other. Another method is to apply the DTW technique to align the two sequence. The distance function for (i, j) can be whether the i^{th} phonemes in the forced alignment is the same as the j^{th} phoneme in the TIMIT caption. The extra phonemes for two sequences are discarded.

Two evaluation metrics are employed to evaluate the model. One is the direct error (mean absolute error, MAE), the other is correlation index (Pearson index). The result is shown in **Table 4-1**. Two models are in comparison. The Fixed Silence model is the forced alignment model with a fixed silence between words. The Optional Silence model has an optional silence during the alignment.

Table 4-1 Test Results on TIMIT

	MAE	**Pearson**
Fixed Silence	1.188	0.400
Optional Silence	1.220	0.411

Table 4-1 shows that forced alignment based on the acoustic model performs well on an average variation of 36ms (1.2×30ms) . Considering the human reaction time is about 300ms as reported in [97], the alignment works well enough for further usage. The Pearson index also indicates that there is a linear relationship between the alignment and the TIMIT. However, the two models on silence have no general preferences for the forced-alignment. Since MAE is more focused, the fixed silence model is applied as the default decoder.

4.3 Summary

This section generally focuses on the decoding process. The decoding process is a part that connects the preceding with the following. And it is the most different parts between the CALL and the ASR.

Therefore, in the beginning of the first section, it compares the CALL and the ASR with their similarities and distinctions. The distinctions mainly come from the mismatch in purpose. Led by the mismatch, the thesis further reviews three major differences. The next part starts from the traditional DTW algorithm and its further extension of the Viterbi decoding.

The experiments on the decoding process is performed on the TIMIT, a phone-level captioned speech dataset. The decoding result shows that the alignment only mismatch on an average of 36ms which is far less than the human reaction time of 300ms. The result confirms its usability for further sections.

5. Pronunciation Scoring Model

This chapter introduces the pronunciation scoring model. Unlike the acoustic model and the decoding model that the CALL has several similarities with the ASR system, the scoring model is the unique part implemented in the CALL.

As discussed in the background introduction, this chapter looks back into the GOP methods in the first section. Next section proposed some implements for the basic GOP method including rescaling methods for young English Learners. The methods change the basic GOP to better accommodate young English learns. The methods can not only accommodate the error alignments but also provide a more convincing score based on a given acoustic model. Section 3 first introduces a self-collected dataset and then evaluates the scoring method on the dataset. The last section summarizes the whole chapter.

5.1 Goodness of Pronunciation (GOP)

For all the pronunciation scoring methods, the fundamental target is to compute high scores for correct phonemes and low scores for mispronounced phonemes. The GOP is considered as the dominating method up until now.

Before the GOP was proposed, there are two methods frequently applied to the word verification task (a similar task to the CALL[①]). The first is based on "a-posteriori" likelihood and the second is modeled with binary classifiers. The "a-posteriori" likelihood method contains two steps [98, 99]. The first step was to spot keywords that may be incorrect by checking the "a-posteriori" probability. For the next step, a classifier was proposed to double check the correct/incorrect labeling. Instead of maximizing the likelihood, the binary classifiers aimed to minimize the prediction errors. The classifiers could be various in which neural networks were always involved [100, 101, 102]. Both methods worked well in word verification. However, they both concentrate on word-level assessment and cannot perform phonetic assessment since they only consider if the word occurs as expected.

The GOP is an "a-posteriori" like methods that measure the phone-level pronunciation scoring. It assumes that the reference text of the utterance spoken by a language learner is given. A typical scenario is a reading task that asks a language learner to read a given text. Recall the **Formula (2-1),** since the sum of the denominator is approximately the same as the maximum, the **Formula (2-1)** can be reinterpreted into following **Formula (5-1)**.

$$GOP(p) = log\big(P(p|O)\big) = |log(\frac{P(O|p)P(p)}{Max_{q\in Q}P(O|q)})| \,/\, L(O) \tag{5-1}$$

The boundary of the segment O is computed through the Viterbi alignments. Under an HMM

① Similar but not the same. The CALL process focuses on the assessment while the word verification task measures whether a given word hypothesis corresponds to its actual occurrence.

structure, the numerator can be generated with a forced alignment to the reference text while the denominator is computed with an unconstrained phoneme loop. Since the maximum of the $P(O|q)$ may not be all the same for the whole segment O, the score of the denominator often applies the sum of the log likelihood per frame through the whole segment. For a DNN-HMM system, the process becomes simple. Both the numerator and the denominator on the frame level can be directly generated from the last Softmax layer.

The GOP score cannot be applied for scoring directly since the range of it varies on each phoneme. Therefore, the GOP is always interpreted as binary classes to determine whether the phoneme is accepted or rejected. Since the HMM fits differently on each phone (the vowels tend to be stable while fricatives are more variable), phone-dependent thresholds are employed. With respect to the GOP's basic statistics, the threshold for p can be defined as **Formula (5-2)**:

$$T_p = \mu_p + \alpha\sigma_p + \beta \tag{5-2}$$

where μ_p and σ_p are the mean and the standard deviation of the GOP score of the phoneme p. In practice, the parameter α and β are determined empirically to yield a similar scale for the global threshold (with fluctuation for specific phonemes).

5.2 Rescaling Methods for Young English Learners

This section firstly reviews some related works on the CALL for young English learners. Then, a Salient GOP (SGOP) is proposed for implementation.

At best of our knowledge, the first work on the CALL for young English learners was done by Hacker et al. in 2005 [103]. The method was in hope to evaluate the children speeches on three levels (word-level, utterance-level, and the speaker-level). The 14-dimension features (defined manually and reduced with Principle Component Analysis, PCA) were applied to train an LDA classifier based on human ratings. The feature dimension was then extended to more and was adjusted specifically for the three levels in his Ph.D. thesis [104]. The method was validated by an L2 children speech datasets of his laboratory. However, his method can only offer an evaluation to a general level (i.e. words, sentences, and speakers) instead of error position (i.e. phone-level) and the scoring solution is still universal for adults and children. In the following part, a Salient GOP method is proposed to solve the weakness above.

The Salient GOP (denoted as SGOP) is shown in **Formula (5-3)**.

$$sGOP(p|O) = \underset{o\in(\frac{1}{4}L(O),\frac{3}{4}L(O))}{Max} \left| \frac{log(Max_{q\in Q}P(o|q))}{log(P(o|p))} \right| \tag{5-3}$$

According to the previous sections, the GOP has three problems. Firstly, the GOP is not in a determined range which is a deficiency for the continuous phonetic score. Next, the Chain acoustic model detects the phonetic spikes instead of the whole segment of the phoneme. Considering the introduction and the departure of a phoneme, the middle part of the segments should weigh more when

scoring. At last, the English sentences are considered to have rhythms including some rules such as slurred phonemes and liaisons between specific phonemes. Therefore, an integration of the sentence level score should take the prosodic rhythm into account.

The proposed SGOP can solve the head two problems. Since the DNN-HMM cannot directly compute the segmental level $P(O|p)$, the segmental level posterior probabilities are represented by the mean of the frame-level probabilities. With the fact that the Chain acoustic model is based on phoneme spike (in other words, introduce the "blank" for each bi-phone unit). Hence, the maximum posterior probability from the middle part of the phoneme ($\frac{1}{4}\ to\ \frac{3}{4}$) is applied to stand for the $P(O|p)$. This process can also reduce the misalignment. Next, the denominator and the numerator are exchanged with their logarithm forms. Under this transformation, the Formula still contains the previous information but confined into an $[0,1]$ interval. The method is more useful for slow speaker since the short phonetic duration is not enough for the process. Therefore, the method fits the young English learners well, because their speaking speed is much slower than the native (The average phonetic duration of young English learner is about 0.176s while the statistic for native is 0.0792s[①]). For the last problem, a further developed GOP method is introduced in the next chapter, based on the combination of rhythm knowledge and the GOP.

5.3 Pronunciation Scoring Evaluation on Speech Dataset of Young English Learners

This section focuses on the pronunciation scoring evaluation with a self-collected dataset of young English learners. For starters, the dataset is discussed including the collection process, caption process, and the data description. Next, there is a comparison between the traditional GOP method and the proposed Salient GOP method.

Since at our best knowledge, there is no public speech corpus on children[②], not to mention young English learners, the evaluation dataset is self-collected. The corpus contains 2.823 hours (10161.4s) English speech from young English learners. The ages of the learners range from post-kindergarten to fifth grade of primary school. There are 128 speakers in total with 59 (46.1%) males and 69 (53.9%) females[③]. Each speaker donates 1 paragraph of speech. Because the paragraph is lengthy for speech processing (most speech recognition system can only afford speech up to 30s. Long speeches results in much more risks in decoding process), the speeches are split into sentences. The splitting process is time-consuming. A software named "A CUT"[④] is employed for splitting. It was developed firstly in 2016 to

① The data is by comparing the TIMIT and the self-collected dataset that is introduced in the next section.
② By the way, there are some corpora for commercial usage available (such as the PF_STAR children speech corpus recorded from British Children and some non-native children speech from European countries).
③ Some of the data does not contains information of speaker. Consequently, there are situations that some speeches are from the same speakers though are separated as different speakers in the dataset. It will not do harm to the general system following except for little losses on the CMVN computation process.
④ The name is translated from Chinese, the real name in Chinese is "一刀切".

generate audio resources of a reading machine for babies. Voice Activity Detection (VAD) is employed to detect the sentences' boundary. The impressive part of the software is the interactive UI to modify the boundaries generated from VAD. After splitting, the 128 passages are turned into 2055 sentences. In average, each sentence contains 28.1 non-silence phonemes and has a 4.95-second duration. For simplification, the corpus names CALL_2K in the following sections. In order to remove the effect of omitting words, all the words in the CALL_2K's transcription have been added to the lexicon dictionary (an implemented version of the CMU Dictionary). A traditional ASR process is performed on the CALL_2K. The recognition WER (the WER is computed with the reference text) is 51.38%. Comparing to the result in **Table 3-2**, the CALL_2K can be validated as a corpus for non-native.

The CALL_2K corpus is the third-level CALL dataset that is with the sentence-level annotation. Two CALL indexes are defined as following **Table 5-1**.

Table 5-1 CALL Evaluation Indexes

Indexes	Description
Pronunciation	The naturalness of the pronunciation[①]
Fluency	The fluency naturalness.

Both the indexes are range from 0 to 10 where 0 is for the worst and 10 is for the best. Two raters are chosen for the scoring process. Both are postgraduate with professional English training. The strictness of the two raters is different (the means of the pronunciation are 7.5 and 8.8) but on a consensus that most of the score is in a range of $(6, 10)$ with a similar scoring curve as shown in **Figure 5-1** where the light gray curves stand for the Rater 1 and the darker blue curves represent the Rater 2. Since the details are not the same, the evaluation of the two raters is not integrated together and is employed separately in the following part of the section.

Pearson, Spearman and Maximal Information Coefficient (MIC) [105] correlation indexes are applied for scoring. Since Pearson index measures the absolute similarities and the Spearman index measures the ranks, the Spearman is preferred according to the subsection **1.2.1**. Since both Pearson and Spearman is linear based, MIC is also applied to detect non-linear relationships.

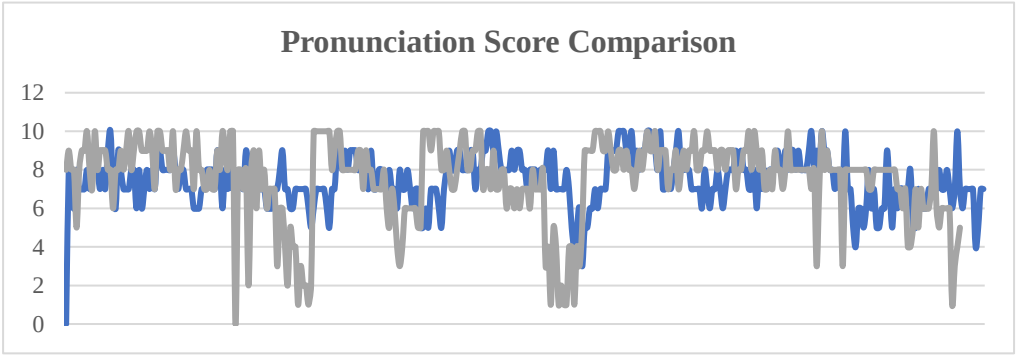
Since the GOP and SGOP only can compute phone-level measures, methods to compute the three indexes are proposed. The computation method is listed in **Formula (5-4)** to **Formula (5-5)**.

$$Pronunciation = (\sum_{O \in \Theta} GOP(p)\,|O|) \,/\, |\Theta| \tag{5-4}$$

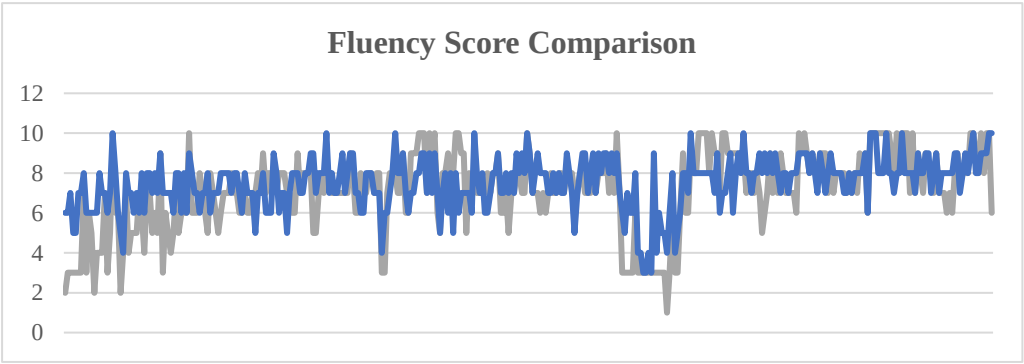
$$Fluency = \sum_{s \in S_{between\_word}} |s| \,/\, |S_{between\_word}| \tag{5-5}$$

① The naturalness is if the word is spoken in a native way that sound fluently. The rater is asked to consider the sentence integrity (if the words in the sentence can be properly recognized) as well.

where Θ denotes the phonetic segments of the sentence, $S_{between\_word}$ represents the silence segments between words. The experiment applies phoneme dependent thresholds for the traditional GOP with parameters $\alpha = 1.0$ and $\beta = -1.5$. If the phoneme is rejected, then the score for it is 0 while the opposite score for acceptance is 100.



(a)



(b)

Figure 5-1 Scoring Curve of Rater 1 and Rater 2

The result is showing in the following **Table 5-2** and **Table 5-3**.

Table 5-2 The Pronunciation Scoring Experiments

Method / Rater	Pearson-1	Pearson-2	Spearman-1	Spearman-2	MIC-1	MIC-2
Rater 1		0.573		0.573		0.276
GOP	0.425	**0.297**	0.370	**0.315**	0.182	0.143
SGOP	**0.452**	0.287	**0.409**	0.304	**0.212**	**0.173**

Table 5-3 The Fluency Scoring Experiments

Method / Rater	Index	Pearson-1	Pearson-2	Spearman-1	Spearman-2	MIC-1	MIC-2
Rater 1	Flu.		0.637		0.567		0.209
Fluency	Flu.	0.542	0.493	0.589	0.498	0.345	0.269

From the correlation indexes between the two raters, it can be found that the two raters generally have consensus on the whole corpus. **Table 5-2** shows that the SGOP method is more similar to the Rater 1 while the GOP method is more like Rater 2. There is an about 3% improvement observed from the SGOP based on all the indexes comparing to the Rater 1. The GOP only outperforms in 1% on linear measures for the Rater2, but it is still worse than SGOP when considering the MIC. Generally, the SGOP performs better than the GOP method.

Since the fluency in this chapter only based on the decoding process, there is no difference between the GOP and the SGOP. A surprising finding is that the fluency from forced-alignment even has more MIC value to each rater, comparing to the MIC between the two raters.

5.4 Summary

The scoring model is the core of the CALL system. It sums all the information gained from previous model to offer scores for L2 Leaners.

In the first section, we explore the GOP method, the baseline of the CALL task.

However, the GOP method generally bases on the GMM-HMM / DNN-HMM model and it has no considerations on mis-alignment. The Salient GOP (SGOP) is proposed for the above problems. It uses the maximum activated posterior probability in the middle of the aligned segments and modifies the GOP to scale it into a fixed range.

The experiments in the last section show that the SGOP slightly outperforms the GOP on most of the test indexes.

6. Prosodic Model

A spoken language without prosody is a human without spirit. It is hard to imagine a speech that is full of equal-length accurate phonemes. The various duration of each phoneme brings emotion and energy into sentences. The prosody is so important, but it was lack of attention comparing to the pronunciation scoring. As discussed in Chapter 2, the prosodic errors are also part of the pronunciation error. Therefore, this chapter mainly focuses on the modeling and scoring of the prosodic feature of speeches.

After a brief review of studies on prosodic modeling and evaluation, a duration model is proposed for modeling the duration of phonemes. Next, the section discusses the prosodic scoring with the duration model. Based on the duration model, a prosodic based GOP method named PGOP is proposed. Moreover, experiments are conducted with the CALL_2K corpus. Finally, the last section summarizes the whole chapter.

6.1 Prosodic Feature and Modeling

According to [6], there are three prosodic parts including stress, intonation, and rhythm. Early in 20 years ago, it had been found that the human judgment on the prosodic evaluation could reach a high consensus [106]. As discussed by Zhang et al., prosodic proficiency is an important part with respect to the judgment from the native speakers [107]. The two facts indicate that people have a clear image for good prosodies and the prosody for a spoken language is important.

Several rudimentary experiments were proposed for prosodic evaluation. Bernstein et al. showed that human judgment on prosody had a linear relationship with fluency measures[①] [108]. And [109] went further to apply multiple learn regression for prosodic assessment. The Support Vector Regression (SVR) is also proposed for the assessment [110]. These attempts on prosodic CALL showed that the general prosodic assessment from human raters is predictable. However, the assessments cannot offer related suggestions since they are too general. To fill the gap between assessments and suggestions, Chen proposed a word-level duration model that can model the duration likelihood based on the context-dependent word [111]. Not only can the method perform the evaluation, but it can also provide suggestions through comparing the test word with the same word occurring in the training set. However, since the evaluation fully dependents on the training set, the system cannot work unless all the context-dependent words are included in the training set (not to mention an unmet word).

6.1.1 Duration Model

It has been widely accepted that the natural language is context dependent. The phonetic duration for spoken languages is the same. There are conventional rules on the phonetic rhythm from phoneticians [112]. However, it can be inferred that there are still hidden conventions that are not included. So far, little attention has been paid to the phonetic duration modeling with the computer instead of some dictionary-based methods [111]. This section proposed a word-level duration model with Bidirectional Recurrent Neural Network (BiRNN) for phonetic duration modeling as shown in **Figure 6-1**.

① The measures are some preliminary measures such as the word per minutes, total pause time and rate of speech.

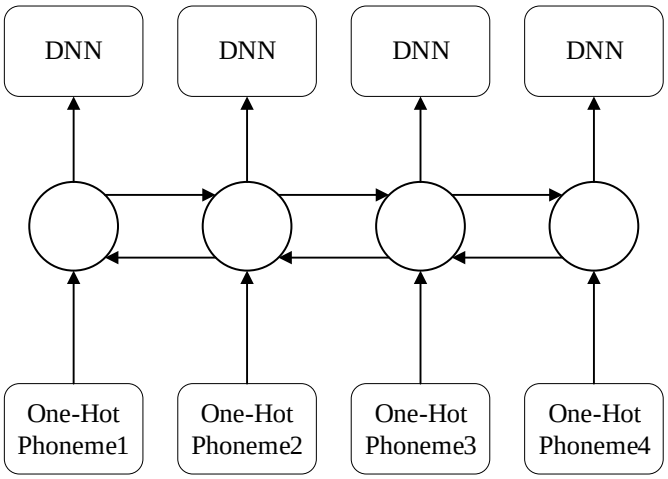


Figure 6-1 The Phonetic Duration Model

The input of the model is a one-hot phonetic sequence and the target is to predict a sequence of duration. To modeling the long-time dependency, the Long Short-Term Memory (LSTM) model is chosen for the RNN model in the framework. To be aware of the context information before and after the phoneme, the LSTM is bi-directional. For the duration, a preliminary thought is to cluster duration for each phoneme into bins and convert the problem into a classification problem. However, the distribution of the phonetic duration is reported to be in an exponential distribution that is not suitable for clustering. Therefore, each output of the RNN is fed into a Fully Connected Layer (FC) without a Softmax layer for regression. Comparing to the model defined in [111], the framework can extract context information on phonemes and adapt to unmet words with the knowledge from the training process.

Two datasets are constructed based on the validated forced alignment decoding model. The source data is from the Librispeech dataset. According to the previous introduction on Librispeech, there are generally two parts in the Librispeech: a cleaned dataset with 460 hours of speeches and another one with 500 hours. To test if the duration model can be more accurate in a clean environment (because the WER in the ASR experiment on the clean test set is far lower than the WER on the “other test” set), a set of 460 hours and a set with 960 hours is separately trained. The phoneme sequences for the words are generated with the implemented CMU Dictionary (discussed in Section 5.3). Since the experiment on the TIMIT shows that the forced alignment decoding process is accurate enough to align speeches into their phonetic sequence. The forced alignment decoding model with fixed silence (proposed in Section 4.2) is applied to generate the duration information. After the alignment, the sentences are split into word-level samples[①].

For the words with the same phonetic sequence, the duration is further normalized. The normalization process scales the same words into the median duration (the median to prevent effects

① Since the words are represents by phonemes, there maybe some confliction due to words that have the same phonetic sequence (such as “sight” and “site”, “straight” and “strait”). Mostly, they are the same. But for some situations, the context may lead to different rhythms. It is admitted that there remains systematical error in the system.

from abnormal durations) of the word but keeping their relative relationship on the duration. The normalization assumes that the phonetic rhythm is comprehended on relative relationships instead of absolute relationships[①]. And the method eases the variation on words' duration with different speaking pace (namely Rate of Speech, ROS in some literature).

6.1.2 Prosodic Scoring

There are three levels for the scoring process including the phone-level the word-level and the sentence-level. The scoring process is as follows:

Firstly, the speech is processed to the forced alignment according to the reference text. Then, the aligned speech is split into word-level. Meanwhile, the transcript split to word-level is fed into the duration model to compute a template duration relationship for the speech. With the normalization technique introduced, the word-level alignments are rescaled.

Since the relative phonetic duration is not asked to be fixed in the spoken language (not the same as music). A variation threshold needs to be computed. The computation process considers the absolute errors of the duration model's evaluation process. The computation extracts the mean and the standard deviation of the absolute errors extracted from the duration model training. The scoring method on word-level is shown in **Formula (6-1)** and **(6-2)**.

$$Prosody = -(\sum_{p \in P} \delta(p)) / |P| \tag{6-1}$$

$$\delta(p) = Max(abs(|(D_{True} - D_{Pred})|) - Mean(E_p) - Std(E_p), 0) \tag{6-2}$$

where $\delta(p)$ is the phone-level scoring function that measures the error exceeding the threshold. The $D_{True} - D_{Pred}$ represents the difference between the true label and the prediction label. The E_p is the absolute prediction erroneous set observed from the duration model training process (To be specific, the test set of the duration model). The sentence level prosodic score can be computed as the same of the word-level.

To combine the prosodic assessment into the previous fluency measure **(Formula (5-5)**), an interpolation method is employed as shown in **Formula (6-3)**.

$$Fluency_{new} = \alpha Fluency_{pre} + \beta Prosody + c \tag{6-3}$$

where α, β, and c are estimated with a linear regression.

6.1.3 Prosodic GOP (PGOP)

The duration of a phoneme often indicates phonetic importance. A stressed phoneme often has a longer duration. Therefore, the speech prominence (stress) detection always implemented duration factors [69, 113]. There are also several topics on speech comprehension that the phonetic duration

① It is like music. The same music piece can be played with different tempo but remains its rhythm through keeping the relative durations.

extensively affected the speech processing mechanism of human [114]. Because the speech comprehension process is partially based on the duration prosody, it can be assumed that the pronunciation with longer duration plays a more important role in determinization of the integrity (if the word can be properly recognized) and naturalness (if the word is spoken fluently) for the language.

Since the duration model can output a reference duration length for a word-level sequence, a Prosodic GOP is proposed as follows:

$$PGOP(p) = \alpha_p \cdot sGOP(p) + \beta_p \cdot \delta(p) * sGOP(p) + c_p \quad (6\text{-}4)$$

where α_p, β_p, and c_p are estimated with linear regression. The construction of the PGOP based on an assumption that a wrong pronunciation with wrong duration does more harm than a wrong pronunciation with better duration[①]. Therefore, the PGOP is a context-related scoring method instead of an independent pronunciation error detector.

6.2 Prosodic Scoring Evaluation on Speech Dataset of Young English Learners

This section evaluates the previously proposed methods on the dataset. For the first part, the section introduces the construction of duration model and its performance in predicting the phonetic duration. Next, the new version of fluency is tested based on the CALL_2K model. At last, the sentence-level PGOP measure is evaluated.

The duration model is a BiRNN with two hidden layers including a bi-directional LSTM and a fully connected layer. The Adam optimizer is adopted. The learning rate of the model and the hidden units of the two layers are tuned with the Grid Search. The final size of the two layers is 512 and the learning rate is determined to be 0.001 at the beginning. Since it is a regression problem, the model applies Mean Square Error (MSE) as its loss and Mean Absolute Error (MAE) as its validation metric. Based on the experimental training status, there are 4 epochs for training in total.

Four experiments on the duration model are conducted. The experimental datasets are randomly split into three parts at a ratio of 8:1:1 for the training set, the validation set, and the testing set. For each dataset, the speeches are first fed into the acoustic model for posteriorgram and then generate duration labels through the forced-align decoder. The atomic unit is a 30ms frame due to the acoustic model. The normalization is conducted as introduced. The result is shown as following **Table 6-1**.

Table 6-1 A Comparison between the Duration Models

Test	Data Set	Normalization	MAE on the Test Set
1	460h-clean	No	0.9037
2	960h-all	No	0.8902
3	460h-clean	Yes	0.6860
4	960h-all	Yes	**0.6647**

① For SGOP that is 0, a small positive dithering factor is added to prevent it from erasing the prosody effect.

From the above table, the MAE on the test set is above 0.6647 frame (about 20ms), which is acceptable for duration modeling. In addition, it can be observed that the introduction of noise data (960-all dataset) can improve the duration modeling. The reason may due to the forced-align decoding process. For the ASR task, the reference text is not pre-defined. The word decoding needs an effective searching process. For the CALL task, the searching errors are avoided, which improve the alignment accuracy. To further validate the duration model, correlation indexes on Pearson, Spearman, and MIC is computed. All the indexes are over 0.5 and the MIC reaches 0.741.

Based on the duration model, the phonetic prosodic thresholds are shown as following **Figure 6-2**. The thresholds vary from 1.02 frame at "UH" to 2.72 frames at "S". And the average threshold is 1.68 frame (about 0.05s).

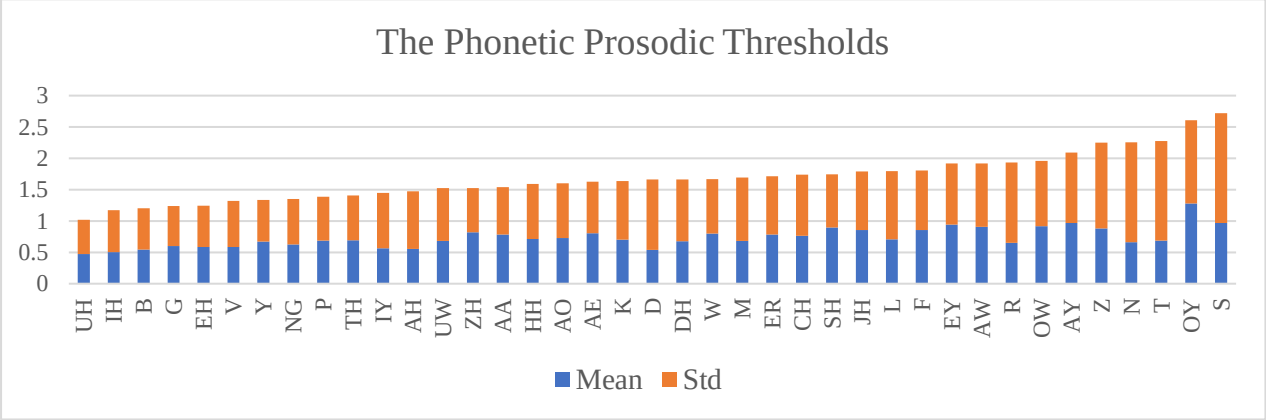


Figure 6-2 The Phonetic Prosodic Thresholds

For the fluency modeling, the linear regression is performed on 70% of the randomly selected dataset of the CALL_2K with separate raters. The final estimators are the mean from the two models (still ranged to $[0, 100]$). The α, β, and c are 0.4086, 0.2115, and 40.15. The evaluation is performed on the other 30% set. From **Table 6-2**, it can be observed that the fluency measures improve much on all the indexes for both raters. Since all values of the indexes exceed the inter-correlation indexes between the two raters, the method is proved to be robust.

Table 6-2 The Fluency Scoring Experiments

Method / Rater	Pearson-1	Pearson-2	Spearman-1	Spearman-2	MIC-1	MIC-2
Rater 1		0.637		0.567		0.209
Traditional Fluency	0.542	0.493	0.589	0.498	0.345	0.269
Duration Scaled	**0.659**	**0.640**	**0.659**	**0.608**	**0.690**	**0.578**

Table 6-3 The PGOP Scoring Experiments

Method / Rater	Pearson-1	Pearson-2	Spearman-1	Spearman-2	MIC-1	MIC-2
Rater 1		0.573		0.573		0.276
GOP	0.425	0.297	0.370	0.315	0.182	0.143
SGOP	0.452	0.287	0.409	0.304	0.212	0.173
PGOP	**0.511**	**0.347**	**0.420**	**0.365**	**0.232**	**0.187**

Like the fluency modeling, the parameters for PGOP is estimated with 70% of the CALL_2K. The α_p, β_p, and c_p are 0.3883, -0.0049 and 51.3421. The result is shown in **Table 6-3**. On all the measures, the PGOP outperforms the GOP method and the SGOP method. It validates the prosodic importance to the human's language comprehension process.

6.3 Summary

The prosody is an always-neglected feature in pronunciation evaluation. This chapter focuses on prosody prediction and evaluation based on the context of the phonetic sequences.

In the first section, the basic duration model is introduced. It models the relative durations given a phonetic sequence. The fluency measure and SGOP are further combined with the predicted phonetic duration sequences. And we name "PGOP" (Prosodic GOP) for the combination of SGOP and prosodic feature.

The second part conducts several experiments on new fluency and the PGOP. It reveals that the introduction of the prosody greatly improves the scoring performances.

7. Conclusions

This Thesis started from a review of the current state of the art in the CALL tasks, especially its combination with the ASR system. A group of algorithms is proposed and evaluated. These methods focus on the three main challenges in the CALL for young English learners including the acoustic modeling, the scoring structure, and the prosodic assessment. This chapter reviews the results of the research and shows several possible directions for further research on the CALL system with young English learners.

For the acoustic model, the Chain model is first employed in a CALL system. It was reported that the Chain model that combines the subsampling TDNN-F and the LF-MMI outperforms the CTC methods for most of the datasets less than 1000 hours. Based on a traditional DNN-HMM topology, the method achieves a WER at 3.8% in the experiment, remarking great modeling for the native. Since previous literature stressed much on the speaker adaptation, the training process takes fMLLR and i-

vectors into consideration as well. For the decoding process, a preliminary Viterbi alignment with the graph is performed. However, to be surprised, the result shows that the optional silence model does not beat the fixed silence model. Therefore, we apply the fixed silence model for its faster decoding process. For the scoring algorithm, a SGOP method is proposed. Based on the acoustic feature of children, the method performed better than the baseline GOP algorithm on the CALL_2K (a self-collected corpus). The prosodic part introduces two application of a phonetic word-level duration model. With duration modeling, we show that the relative phonetic duration is predictable. The prediction result significantly improves the fluency scoring and the pronunciation scoring through a regression method.

Three major contributions can be identified. Firstly, the TDNN-HMM with LF-MMI is proved to be a great tool for the CALL acoustic modeling with higher recognition accuracy and more explicit posteriorgram. Next, the SGOP is proposed specifically for young English learners. The algorithm fills the gap of the specific CALL for children. At last, a duration model is applied to predict the relative word-level phonetic duration. Its interpolation forms with the basic fluency measure and the SGOP method significantly exceed the baseline algorithms. Apart from the main contributions, the thesis also presents a corpus of Chinese young English learner that is self-collected from students range from the kindergarten to primary school.

In future research, the proposed methods can be further extended.

For the acoustic model, the bi-linguistic model may be further direction on the CALL. According to previous literature, a bi-linguistic model can help to detect the errors specifically, and it can accurately dig out the confusing phonemes for the non-native. However, the method was not successful due to the poor acoustic model. In addition, the native children speech corpora should be added into the acoustic training. For this research, the training set lacks native children speech. It may hurt the adaptation process with fMLLR and i-vector.

Another possible extension for the research is the prosodic model. As reported in this thesis, the prosodic duration-based model significantly improves the CALL tasks' performances. The prosody of a speech generally contains the stress (prominence), the duration (rhythm) and the pitch (intonation). Since the phonetic duration is found to have a relative relationship, it is reasonable to assume that the prominence and intonation may have a similar pattern. Therefore, a possible direction for the research is to model the phonetic stress and the phonetic pitch. Moreover, only is the duration of word-level studied. With so many natural language processing techniques, the duration prediction may be more accurate with more context information (e.g. other words or phonemes in the context).

For newly sprouting studies, some works were proposed to borrow knowledge from mature algorithms from other fields. For example, there was CALL research that borrowed the insights from machine translation [115]. It treated standard pronunciation as the original language and L2 pronunciation as the object language. With the mechanism from the machine translation, it can infer possible erroneous phonetic sequence.

It must be mentioned that the CALL is not stereotyped. There are several new methods other than the traditional process. Basically, the new attempts can be classified into four categories: audiovisual studies, gamification, and personalization. The audiovisual studies researched on how to build the 3-dimensional model with acoustic features that can not only detect pronunciation problems but also teach students how to pronounce like the native [116, 117, 118]. Another attempt on the CALL is the gamification that employed game systems for a better learning efficiency [119, 120, 121]. The personalization methods generally adopt the Text-to-Speech (TTS) methods [121, 122, 123]. A typical process of their training is to first extract voice feature from users and then generates (or converts) the personalized voice for the users. It assumes that users can understand their errors faster with their own voices.

Acknowledgments

Firstly, I would like to express my deepest gratitude to my supervisor, Mrs. Jin, for her patient and experienced guidance and constant encouragement. Without her illuminating suggestions, the thesis could not settle down into the present form.

Secondly, my thanks would go to my beloved family. For the thesis, my father devoted much of his time working on the construction of CALL_2K corpus and my mother invited several friends to join the data collection process. I also own my sincere gratitude to my younger cousins who have donated their voices into the CALL_2K corpus.

Last but not the least, I appreciate the tutoring from Miss. Xiao, the mentor of my former internship and would like to express my sincere gratitude to her, as she led me into the world of the CALL and taught me many basics in the related fields.

References

[1] Witt, S., & Young, S. (1997). Computer-assisted pronunciation teaching based on automatic speech recognition. *Language Teaching and Language Technology Groningen, The Netherlands*.

[2] Chapelle, C. (1998). Multimedia CALL: Lessons to be learned from research on instructed SLA. *Language learning & technology*, *2*(1), 22.

[3] Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive science*, *14*(1), 11-28.

[4] Imoto, K., Tsubota, Y., Raux, A., Kawahara, T., & Dantsuji, M. (2002). Modeling and automatic detection of English sentence stress for computer-assisted English prosody learning system. In *Seventh International Conference on Spoken Language Processing*.

[5] Beatty, K. (2013). *Teaching & researching: Computer-assisted language learning*. Routledge.

[6] Witt, S. M. (2012). Automatic error detection in pronunciation training: Where we are and where we need to go. *Proc. IS ADEPT*, *6*.

[7] Bernstein, J., Cohen, M., Murveit, H., Rtischev, D., & Weintraub, M. (1990). Automatic evaluation and training in English pronunciation. In *First International Conference on Spoken Language Processing*.

[8] Witt, S., & Young, S. (1998, May). Performance measures for phone-level pronunciation teaching in CALL. In *Proc. of the Workshop on Speech Technology in Language Learning* (pp. 99-102).

[9] Tsubota, Y., Kawahara, T., & Dantsuji, M. (2002). Recognition and verification of English by Japanese students for computer-assisted language learning system. In *Seventh International Conference on Spoken Language Processing*.

[10] Yoon, S. Y., Hasegawa-Johnson, M., & Sproat, R. (2010). Landmark-based automated pronunciation error detection. In *Eleventh Annual Conference of the International Speech Communication Association*.

[11] Stanley, T., & Hacioglu, K. (2012). Improving L1-specific phonological error diagnosis in computer assisted pronunciation training. In *Thirteenth Annual Conference of the International Speech Communication Association*.

[12] Yan, K., & Gong, S. (2011). Pronunciation proficiency evaluation based on discriminatively refined acoustic models. *International Journal of Information Technology and Computer Science*, *3*(2), 17-23.

[13] Huang, H., Wang, J., & Abudureyimu, H. (2012). Maximum F1-score discriminative training for automatic mispronunciation detection in computer-assisted language learning. In *Thirteenth Annual Conference of the International Speech Communication Association*.

[14] Nicolao, M., Beeston, A. V., & Hain, T. (2015, April). Automatic assessment of English learner pronunciation using discriminative classifiers. In *2015 IEEE International Conference on Acoustics,*

Speech and Signal Processing (ICASSP) (pp. 5351-5355). IEEE.

[15] Gao, Y., Xie, Y., Cao, W., & Zhang, J. (2015). A study on robust detection of pronunciation erroneous tendency based on deep neural network. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[16] Abdou, S. M., Hamid, S. E., Rashwan, M., Samir, A., Abdel-Hamid, O., Shahin, M., & Nazih, W. (2006). Computer aided pronunciation learning system using speech recognition techniques. In *Ninth International Conference on Spoken Language Processing*.

[17] van Doremalen, J. J. H. C., Cucchiarini, C., & Strik, H. (2010). Using Non-Native Error Patterns to Improve Pronunciation Verification. *Proceedings of Interspeech 2010*, CD.

[18] Chen, J. C., Lo, J. L., & Jang, J. S. (2004, December). Computer assisted spoken English learning for Chinese in Taiwan. In *2004 International Symposium on Chinese Spoken Language Processing* (pp. 337-340). IEEE.

[19] Strik, H., Truong, K. P., Wet, F. D., & Cucchiarini, C. (2007). Comparing classifiers for pronunciation error detection. In *Eighth Annual Conference of the International Speech Communication Association*.

[20] Wang, L., Feng, X., & Meng, H. M. (2008). Automatic generation and pruning of phonetic mispronunciations to support computer-aided pronunciation training. In *Ninth Annual Conference of the International Speech Communication Association*.

[21] Chen, L. Y., & Jang, J. S. R. (2010). Automatic pronunciation scoring using learning to rank and DP-based score segmentation. In *Eleventh Annual Conference of the International Speech Communication Association*.

[22] Chen, N. F., & Li, H. (2016, December). Computer-assisted pronunciation training: From pronunciation scoring towards spoken language learning. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (pp. 1-7). IEEE.

[23] Zhao, T., Hoshino, A., Suzuki, M., Minematsu, N., & Hirose, K. (2012, December). Automatic Chinese pronunciation error detection using SVM trained with structural features. In *2012 IEEE Spoken Language Technology Workshop (SLT)* (pp. 473-478). IEEE.

[24] Wang, H., Waple, C. J., & Kawahara, T. (2009). Computer assisted language learning system based on dynamic question generation and error prediction for automatic speech recognition. *Speech Communication*, *51*(10), 995-1005.

[25] Zhou, W., Qin, L., You, X., Zhang, J., Chiu, T., & Ye, W. (2007, August). A computer assisted language learning system based on error trends grouping. In *2007 International Conference on Natural Language Processing and Knowledge Engineering* (pp. 256-261). IEEE.

[26] Wang, Y. B., & Lee, L. S. (2012, March). Improved approaches of modeling and detecting error patterns with empirical analysis for computer-aided pronunciation training. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5049-5052). IEEE.

[27] Lo, W. K., Zhang, S., & Meng, H. (2010). Automatic derivation of phonological rules for mispronunciation detection in a computer-assisted pronunciation training system. In *Eleventh Annual Conference of the International Speech Communication Association*.

[28] Harrison, A. M., Lo, W. K., Qian, X. J., & Meng, H. (2009). Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training. In *International Workshop on Speech and Language Technology in Education*.

[29] Wang, Y. B., & Lee, L. S. (2013, May). Toward unsupervised discovery of pronunciation error patterns using universal phoneme posteriorgram for computer-assisted language learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8232-8236). IEEE.

[30] Lee, A., & Glass, J. (2015). Mispronunciation detection without nonnative training data. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[31] Ohkawa, Y., Suzuki, M., Ogasawara, H., Ito, A., & Makino, S. (2009). A speaker adaptation method for non-native speech using learners' native utterances for computer-assisted language learning systems. *Speech Communication*, *51*(10), 875-882.

[32] Luo, D. (2009). Quantitative analysis of the adverse effect of speaker adaptation on pronunciation evaluation. In *Proc ASJ Spring Meeting* (pp. 173-176).

[33] Ito, A., Nagasawa, T., Ogasawara, H., Suzuki, M., & Makino, S. (2006). Automatic detection of English mispronunciation using speaker adaptation and automatic assessment of English intonation and rhythm. *Educational technology research*, *29*(1-2), 13-23.

[34] Witt, S., & Young, S. (1998). Estimation of models for non-native speech in computer-assisted language learning based on linear model combination. In *Fifth International Conference on Spoken Language Processing*.

[35] Ito, A., Lim, Y. L., Suzuki, M., & Makino, S. (2007). Pronunciation error detection for computer-assisted language learning system based on error rule clustering using a decision tree. *Acoustical science and technology*, *28*(2), 131-133.

[36] Luo, D., Qiao, Y., Minematsu, N., Yamauchi, Y., & Hirose, K. (2010). Regularized-MLLR speaker adaptation for computer-assisted language learning system. In *Eleventh Annual Conference of the International Speech Communication Association*.

[37] Chen, J. C., Jang, J. S. R., Li, J. Y., & Wu, M. C. (2004, June). Automatic pronunciation assessment for Mandarin Chinese. In *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)* (Vol. 3, pp. 1979-1982). IEEE.

[38] Hu, W., Qian, Y., & Soong, F. K. (2014, May). A DNN-based acoustic modeling of tonal language and its application to Mandarin pronunciation training. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 3206-3210). IEEE.

[39] Liao, H. C., Guan, Y. H., Tu, J. J., & Chen, J. C. (2014). A prototype of an adaptive Chinese pronunciation training system. *System*, *45*, 52-66.

[40] Al Hindi, A., Alsulaiman, M., Muhammad, G., & Al-Kahtani, S. (2014, November). Automatic pronunciation error detection of nonnative Arabic Speech. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)* (pp. 190-197). IEEE.

[41] Burgos, P., Cucchiarini, C., van Hout, R. W. N. M., & Strik, H. (2013). Pronunciation errors by Spanish learners of Dutch: A data-driven study for ASR-based pronunciation training.

[42] Lee, A., & Glass, J. R. (2014). Context-dependent pronunciation error pattern discovery with limited annotations. In *Fifteenth Annual Conference of the International Speech Communication Association*.

[43] Tong, R., Lim, B. P., Chen, N. F., Ma, B., & Li, H. (2014, May). Subspace Gaussian mixture model for computer-assisted language learning. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5347-5351). IEEE.

[44] Carranza, M., Cucchiarini, C., Burgos, P., & Strik, H. (2014). Non-native speech corpora for the development of computer assisted pronunciation training systems. *Proceedings of Edulearn*, 3624-3633.

[45] Russell, M., Brown, C., Skilling, A., Series, R., Wallace, J., Bonham, B., & Barker, P. (1996, October). Applications of automatic speech recognition to speech and language development in young children. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96* (Vol. 1, pp. 176-179). IEEE.

[46] Potamianos, A., Narayanan, S., & Lee, S. (1997). Automatic speech recognition for children. In *Fifth European Conference on Speech Communication and Technology*.

[47] Das, S., Nix, D., & Picheny, M. (1998, May). Improvements in children's speech recognition performance. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*(Vol. 1, pp. 433-436). IEEE.

[48] Lee, S., Potamianos, A., & Narayanan, S. (1999). Acoustics of children's speech: Developmental changes of temporal and spectral parameters. *The Journal of the Acoustical Society of America*, *105*(3), 1455-1468.

[49] Li, Q., & Russell, M. J. (2001). Why is automatic recognition of children's speech difficult?. In *Seventh European Conference on Speech Communication and Technology*.

[50] Hagen, A., Pellom, B., & Cole, R. (2003, December). Children's speech recognition with application to interactive books and tutors. In *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721)* (pp. 186-191). IEEE.

[51] Giuliani, D., & Gerosa, M. (2003, April). Investigating recognition of children's speech. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).* (Vol. 2, pp. II-137). IEEE.

[52] Potamianos, A., & Narayanan, S. (2003). Robust recognition of children's speech. *IEEE Transactions on speech and audio processing*, *11*(6), 603-616.

[53] Gerosa, M., & Giuliani, D. (2004). Preliminary investigations in automatic recognition of English sentences uttered by Italian children. In *InSTIL/ICALL Symposium 2004*.

[54] Kim, I. S. (2006). Automatic speech recognition: Reliability and pedagogical implications for teaching pronunciation. *Educational Technology & Society*, *9*(1), 322-334.

[55] Russell, M., & D'Arcy, S. (2007). Challenges for computer recognition of children's speech. In *Workshop on Speech and Language Technology in Education*.

[56] Neri, A., Mich, O., Gerosa, M., & Giuliani, D. (2008). The effectiveness of computer assisted pronunciation training for foreign language learning by children. *Computer Assisted Language Learning*, *21*(5), 393-408.

[57] Gray, S. S., Willett, D., Lu, J., Pinto, J., Maergner, P., & Bodenstab, N. (2014). Child automatic speech recognition for US English: child interaction with living-room-electronic-devices. In *WOCCI* (pp. 21-26).

[58] Shivakumar, P. G., Potamianos, A., Lee, S., & Narayanan, S. (2014, September). Improving speech recognition for children using acoustic adaptation and pronunciation modeling. In *WOCCI* (pp. 15-19).

[59] Serizel, R., & Giuliani, D. (2014, December). 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). IEEE.

[60] Liao, H., Pundak, G., Siohan, O., Carroll, M. K., Coccaro, N., Jiang, Q. M., ... & Bacchiani, M. (2015). Large vocabulary automatic speech recognition for children. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[61] Baum, L. E., & Eagon, J. A. (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, *73*(3), 360-363.

[62] Baker, J. (1975). The DRAGON system--An overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *23*(1), 24-29.

[63] Huang, X., Acero, A., Hon, H. W., & Reddy, R. (2001). *Spoken language processing: A guide to theory, algorithm, and system development* (Vol. 1). Upper Saddle River: Prentice hall PTR.

[64] Campbell, J. P., Tremain, T. E., & Welch, V. C. (1991). The dod 4.8 kbps standard (proposed federal standard 1016). In *Advances in Speech Coding* (pp. 121-133). Springer, Boston, MA.

[65] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Silovsky, J. (2011). *The Kaldi speech recognition toolkit* (No. CONF). IEEE Signal Processing Society.

[66] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., ... & Valtchev, V. (1997). The HTK book, vol. 2. *Entropic Cambridge Research Laboratory Cambridge*, *4*.

[67] Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, *13*(2), 260-269.

[68] Deng, L., Lennig, M., Seitz, F., & Mermelstein, P. (1990). Large vocabulary word recognition using context-dependent allophonic hidden Markov models. *Computer Speech & Language*, *4*(4), 345-357.

[69] Tamburini, F. (2003). Automatic prosodic prominence detection in speech using acoustic features: an unsupervised system. In *Eighth European Conference on Speech Communication and Technology*.

[70] Huang, X., Acero, A., Alleva, F., Hwang, M., Jiang, L., & Mahajan, M. (1996). From Sphinx-II to whisper—making speech recognition usable. In *Automatic Speech and Speaker Recognition* (pp. 481-508). Springer, Boston, MA.

[71] Hon, H. W., & Lee, K. F. (1991, April). CMU robust vocabulary-independent speech recognition system. In *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing* (pp. 889-892). IEEE.

[72] Leggetter, C. J., & Woodland, P. C. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer speech & language*, *9*(2), 171-185.

[73] Gales, M. J. (1998). Maximum likelihood linear transformations for HMM-based speech recognition. *Computer speech & language*, *12*(2), 75-98.

[74] Ganitkevitch, J. (2005, August). Speaker adaptation using maximum likelihood linear regression. In *Rheinish-Westflesche Technische Hochschule Aachen, the course of Automatic Speech Recognition, www-i6. informatik. rwthaachen. de/web/Teaching/Seminars/SS05/ASR/Juri Ganitkevitch Ausarbeitung. pdf*.

[75] Anastasakos, T., McDonough, J., Schwartz, R., & Makhoul, J. (1996, October). A compact model for speaker-adaptive training. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96* (Vol. 2, pp. 1137-1140). IEEE.

[76] Seide, F., Li, G., Chen, X., & Yu, D. (2011, December). Feature engineering in context-dependent deep neural networks for conversational speech transcription. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding* (pp. 24-29). IEEE.

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

[78] Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, *20*(1), 30-42.

[79] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. J. (1995). Phoneme recognition using time-delay neural networks. *Backpropagation: Theory, Architectures and Applications*, 35-61.

[80] Dugast, C., Devillers, L., & Aubert, X. (1994). Combining TDNN and HMM in a Hybrid System. *IEEE Transactions on Speech and Audio Processing*, *2*(1 PART II), 217.

[81] Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 6645-6649). IEEE.

[82] Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006, June). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning* (pp. 369-376). ACM.

[83] Sak, H., Senior, A., Rao, K., Irsoy, O., Graves, A., Beaufays, F., & Schalkwyk, J. (2015, April). Learning acoustic frame labeling for speech recognition with recurrent neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 4280-4284). IEEE.

[84] Peddinti, V., Povey, D., & Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[85] Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., ... & Khudanpur, S. (2016, September). Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. In *Interspeech* (pp. 2751-2755).

[86] Veselý, K., Ghoshal, A., Burget, L., & Povey, D. (2013, August). Sequence-discriminative training of deep neural networks. In *Interspeech* (Vol. 2013, pp. 2345-2349).

[87] Povey, D. (2005). *Discriminative training for large vocabulary speech recognition* (Doctoral dissertation, University of Cambridge).

[88] Su, H., Li, G., Yu, D., & Seide, F. (2013, May). Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6664-6668). IEEE.

[89] Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohamadi, M., & Khudanpur, S. (2018, September). Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Proceedings of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH 2018), Hyderabad, India*.

[90] Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015, April). Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5206-5210). IEEE.

[91] Sjölander, K. (2003, June). An HMM-based system for automatic segmentation and alignment of speech. In *Proceedings of Fonetik* (Vol. 2003, pp. 93-96).

[92] Moreno, P. J., Joerg, C., Thong, J. M. V., & Glickman, O. (1998). A recursive algorithm for the forced alignment of very long audio segments. In *Fifth International Conference on Spoken Language Processing*.

[93] Berndt, D. J., & Clifford, J. (1994, July). Using dynamic time warping to find patterns in time series. In *KDD workshop* (Vol. 10, No. 16, pp. 359-370).

[94] Sakoe, H., Chiba, S., Waibel, A., & Lee, K. F. (1990). Dynamic programming algorithm optimization for spoken word recognition. *Readings in speech recognition*, *159*, 224.

[95] Forney, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, *61*(3), 268-278.

[96] Zue, V., Seneff, S., & Glass, J. (1990). Speech database development at MIT: TIMIT and beyond. *Speech communication*, *9*(4), 351-356.

[97] Agus, T. R., Thorpe, S. J., Suied, C., & Pressnitzer, D. (2010, May). Characteristics of human voice processing. In Proceedings of 2010 IEEE International Symposium on Circuits and Systems (pp. 509-512). IEEE.

[98] Rivlin, Z., Cohen, M., Abrash, V., & Chung, T. (1996, May). A phone-dependent confidence measure for utterance rejection. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings* (Vol. 1, pp. 515-517). IEEE.

[99] Cox, S., & Rose, R. (1996, May). Confidence measures for the switchboard database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings* (Vol. 1, pp. 511-514). IEEE.

[100] Colton, D., Fanty, M., & Cole, R. A. (1995). Utterance Verification Improves Closed-set Recognition and Out-of-vocabulary Rejection. In *Fourth European Conference on Speech Communication and Technology*.

[101] Lleida, E., & Rose, R. C. (1996, May). Efficient decoding and training procedures for utterance verification in continuous speech recognition. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings* (Vol. 1, pp. 507-510). IEEE.

[102] Rose, R. C., Juang, B. H., & Lee, C. H. (1995, May). A training procedure for verifying string hypotheses in continuous speech recognition. In *1995 International Conference on Acoustics, Speech, and Signal Processing*(Vol. 1, pp. 281-284). IEEE.

[103] Hacker, C., Batliner, A., Steidl, S., Nöth, E., Niemann, H., & Cincarek, T. (2005). Assessment of non-native children's pronunciation: Human marking and automatic scoring. *Proc. SPEECOM*, *1*, 123-126.

[104] Hacker, C. (2009). *Automatic assessment of children speech to support language learning* (Vol. 30). Logos Verlag Berlin GmbH.

[105] Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., ... & Sabeti, P. C. (2011). Detecting novel associations in large data sets. *science*, *334*(6062), 1518-1524.

[106] Cucchiarini, C., Strik, H., & Boves, L. W. J. (1998). Quantitative assessment of second language learners' fluency: an automatic approach.

[107] Zhang, Y., & Elder, C. (2011). Judgments of oral proficiency by non-native and native English speaking teacher raters: Competing or complementary constructs?. *Language Testing*, *28*(1), 31-50.

[108] Bernstein, J., Cheng, J., & Suzuki, M. (2011). Fluency changes with general progress in L2

proficiency. In *Twelfth Annual Conference of the International Speech Communication Association*.

[109] Hönig, F., Batliner, A., Weilhammer, K., & Nöth, E. (2010). Automatic assessment of non-native prosody for english as l2. In *Speech Prosody 2010-Fifth International Conference*.

[110] Maier, A., Hönig, F., Zeißler, V., Batliner, A., Körner, E., Yamanaka, N., ... & Nöth, E. (2009). A language-independent feature set for the automatic evaluation of prosody. In *Tenth Annual Conference of the International Speech Communication Association*.

[111] Cheng, J. (2011). Automatic assessment of prosody in high-stakes English tests. In *Twelfth Annual Conference of the International Speech Communication Association*.

[112] Hayes, B. (1984). The phonology of rhythm in English. *Linguistic inquiry*, *15*(1), 75-102.

[113] Wik, P., & Hjalmarsson, A. (2009). Embodied conversational agents in computer assisted language learning. *Speech communication*, *51*(10), 1024-1037.

[114] Cutler, A., Dahan, D., & Van Donselaar, W. (1997). Prosody in the comprehension of spoken language: A literature review. *Language and speech*, *40*(2), 141-201.

[115] Stanley, T., Hacioglu, K., & Pellom, B. (2011). Statistical machine translation framework for modeling phonological errors in computer assisted pronunciation training system. In *Speech and Language Technology in Education*.

[116] Wang, L., Chen, H., Li, S., & Meng, H. M. (2012). Phoneme-level articulatory animation in pronunciation training. *Speech Communication*, *54*(7), 845-856.

[117] Iribe, Y., Manosavan, S., Katsurada, K., Hayashi, R., Zhu, C., & Nitta, T. (2012, March). Improvement of animated articulatory gesture extracted from speech for pronunciation training. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5133-5136). IEEE.

[118] Kröger, B. J., Birkholz, P., Hoffmann, R., & Meng, H. (2010). Audiovisual tools for phonetic and articulatory visualization in computer-aided pronunciation training. In *Development of multimodal interfaces: Active listening and synchrony* (pp. 337-345). Springer, Berlin, Heidelberg.

[119] Su, P. H., Wang, Y. B., Yu, T. H., & Lee, L. S. (2013, May). A dialogue game framework with personalized training using reinforcement learning for computer-assisted language learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 8213-8217). IEEE.

[120] Su, P. H., Wu, C. H., & Lee, L. S. (2015). A recursive dialogue game for personalized computer-aided pronunciation training. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(1), 127-141.

[121] Tejedor-García, C., Escudero-Mancebo, D., González-Ferreras, C., Cámara-Arenas, E., & Cardeñoso-Payo, V. (2016). Improving L2 production with a gamified computer-assisted pronunciation training tool, TipTopTalk!.

[122] Handley, Z., & Hamel, M. J. (2005). Establishing a methodology for benchmarking speech synthesis for computer-assisted language learning (CALL). *Language Learning & Technology*, *9*(3), 99-

120.

[123] De Meo, A., Vitale, M., Pettorino, M., Cutugno, F., & Origlia, A. (2013). Imitation/self-imitation in computer-assisted prosody training for Chinese learners of L2 Italian.

Appendix: An Example of the CALL System

This appendix gives an example of the proposed CALL system. The example is chosen from the CALL_2K corpus. The speech is about 6.5s from a 9-year-old girl. The reference text is “An ant is small. But it is strong. It can carry fifty times its weight.”. There are 15 words. In total, 47 phonemes are in the sentences where includes 17 vowels and 30 consonants. The raw waveform of the speech is shown in **Figure A-1**.

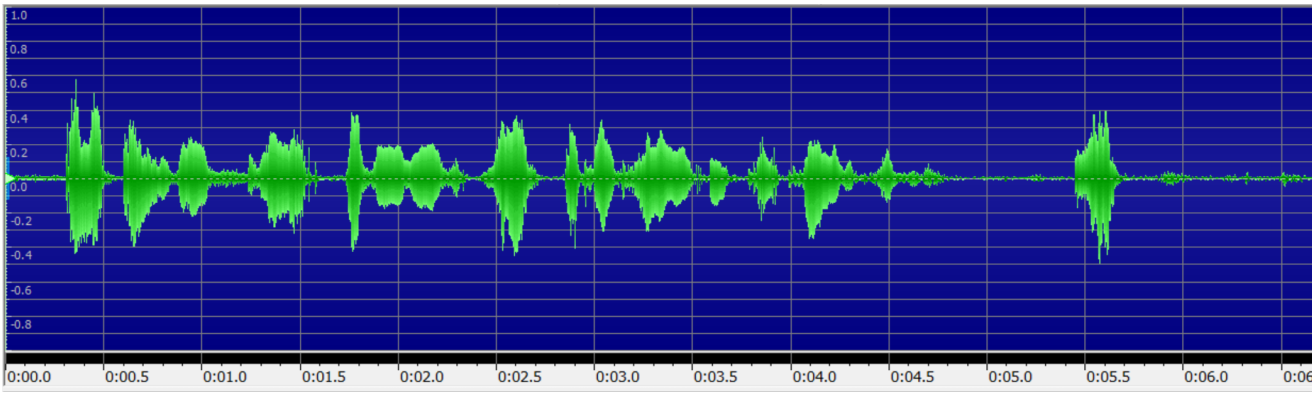


Figure A-1 The Example for the CALL System

The example sentence has an obvious error occurred in the last word “weight”. The true phonetic sequence of the word “weight” is “W EY1 T”. However, the actual spoken word is “W AY1 T”, like the word “white”. The word is essential and greatly affects the comprehension process. The forced alignment process aligns the word “weight” to occurred from 180th frame to 200th frame (which is from 5.4s to 6.0s). **Figure A-2** zooms in the specific word and its alignment details. Based on the posteriorgram, the SGOP for the three phonemes are 100, 0, and 100 respectively and their durations are 7, 1, 12 frames (1 frame has 30ms). The excellence in error detection comes from the powerful acoustic model.

The recommend duration generated from the duration for “weight” is [1.138, 1.639, 1.709]. The duration sequence is then divided by 4.486 (the sum of the duration sequence) and multiplied 20. The sequence hence becomes [5.073, 7.307, 7.619]. Comparing with the error threshold (pre-computed from the duration model) [1.6691, 1.6738, 2.2753], the phonetic-prosodic errors are [0.2579, 7.9808, 2.1057]. With traditional GOP method, the prosodic error cannot be detected. Therefore, the average GOP for the erroneous word is still high (95 / 100 for the “weight”). The PGOP (70 / 100) on the otherwise can detect the abnormal duration and boost the error for the word with prosody.

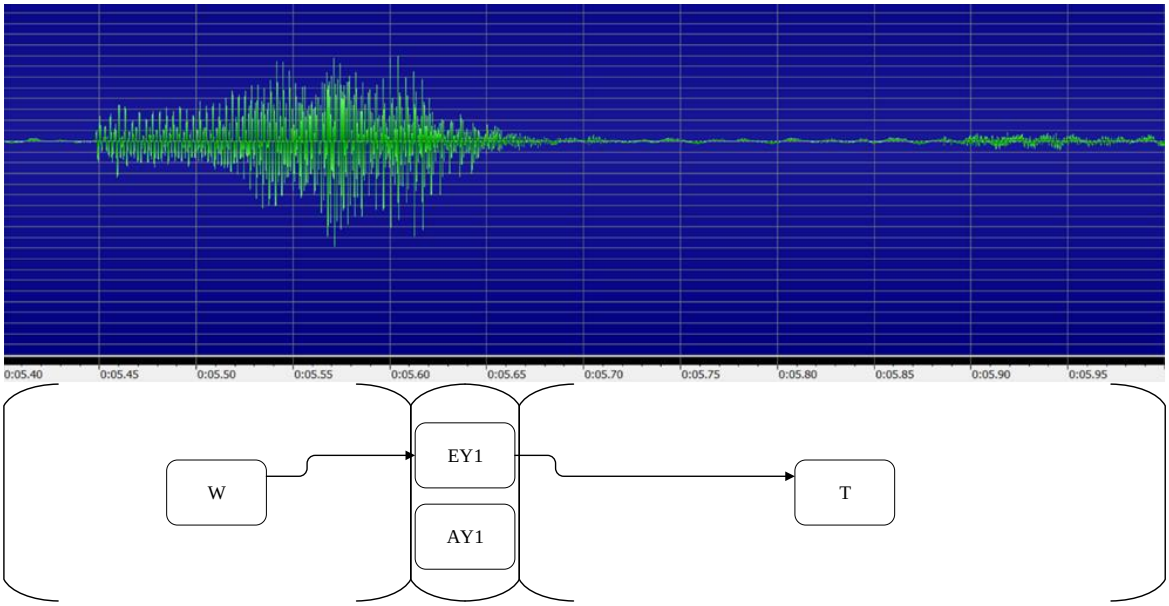


Figure A-2 The Error Word "Weight"

Additionally, the prosodic duration can offer suggestions instead of purely working for scoring. The prosodic error computed in the previous discussion can be generated for prosody suggestions. With a high accurate duration prediction, the suggestions can be useful.