ON THE USE OF FRAME AND SEGMENT-BASED METHODS FOR THE DETECTION AND CLASSIFICATION OF SPEECH SOUNDS AND FEATURES

by

JUN HOU

A Dissertation submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in

Electrical and Computer Engineering

written under the direction of

Professor Lawrence Rabiner

and approved by

____

____

____

____

____

New Brunswick, New Jersey

October, 2009

ABSTRACT OF THE DISSERTATION

On the Use of Frame and Segment-Based Methods for the Detection and Classification of Speech Sounds and Features

by Jun Hou

Dissertation Director:

Professor Lawrence Rabiner

Statistical data-driven methods and knowledge-based methods are two recent trends in Automatic Speech Recognition (ASR). Hidden Markov Model (HMM)-based speech recognition techniques have achieved great success for controlled tasks and environments. However, when we require improved accuracy and robustness (closer to Human Speech Recognition (HSR)), HMM algorithms for speech recognition gradually fail. Hence a need has emerged to incorporate higher level linguistic information into ASR systems in order to further discriminate between speech classes or phonemes with high confusion rates. The Automatic Speech Attribute Transcription (ASAT) project is one of the recent research efforts that has tried to bridge the gap between ASR and HSR.

In this thesis we focus on the design and optimization of the front end processing of the ASAT system, whose goal is to estimate a set of attribute and phoneme probability lattices which can be combined with information from higher level knowledge sources in a set of speech event verification modules in order to make a final recognition decision.

We propose a set of both frame-based methods and segment-based methods to improve the recognition performance of distinctive features and phonemes in English.

We also study and evaluate both a parallel speech feature organization and a hierarchical phoneme topology. There are 4 main parts in this thesis work. In the first part, we use frame-based methods to estimate the likelihood of static sounds (e.g., steady vowels, fricatives, etc), and implement the parallel feature detection using Multi-Layer Perceptrons (MLPs) in order to detect the 14 Sound Pattern of English (SPE) features. In the second part, we use segment-based methods to classify dynamic sounds (e.g., stop consonants, diphthongs, etc), and use Time-Delay Neural Networks (TDNNs) to recognize phoneme classes in a hierarchical phoneme and feature organization. In the third part and in the forth part, we combine the frame-based parallel speech feature detection system and the segment-based hierarchical phoneme classification system to improve the overall phoneme classification performance and the speech feature detection performance.

The main contribution of this thesis is the creation of a phoneme recognizer that overcomes the disadvantages of pure statistical or knowledge-based systems, and provides a way to incorporate acoustic/phonetic/linguistic knowledge into an existing (HMM-based) automatic speech recognition system.

Acknowledgements

Firstly, I would like to express my deep and sincere gratitude to my advisor, Professor Lawrence Rabiner. This thesis wouldn't have been possible without his guidance, support and patience. His experienced advice and sharp insight of this research field often enlightened me and led to constructive discussions. I have learned a lot from him and enjoyed very much working under his supervision.

Special thanks to Dr. Chin-Hui Lee, our project leader, for providing me with such a great opportunity to join a great research group and for all the help and insightful discussions. Special thanks to Dr. Sorin Dusan for the discussions and comments when we worked together in the same research group.

My sincere thanks are due to Dr. Ivan Marsic, Dr. Joseph Wilder, Dr. Aaron Rosenberg and Dr. Chin-Hui Lee, for taking their valuable time to serve in my dissertation committee and providing valuable comments. I am especially grateful to Dr. Rosenberg for the detailed review of the thesis.

During the course of this research effort I collaborated with many professors and students, and I am grateful to Dr. Biing Hwang Juang, Dr. Mark Clements, Dr. Eric Fosler-Lussier, Dr. Keith Johnson, Dr. Jinyu Li, Mr. Brett Matthews, Ms. Ilana Bromberg, etc., for all the collaborations in our project.

I owe my loving thanks to my parents, Mr. Yanhui Hou and Ms. Baoyu Liu, for the support and encouragement during the years of my study. I also wish to thank my husband and my daughter, for their understanding and support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Knowledge-based and statistics-based approaches are two current directions in Automatic Speech Recognition (ASR), and both have evolved over time [54]. Traditional statistical methods, like Hidden Markov Models (HMM), have achieved great success for controlled tasks [53]. One characteristic of such methods is that the recognizer is blindly trained (using an extensive training set of labeled data) without incorporating the knowledge of how humans actually produce and understand speech. When we require improved (closer to human) accuracy and robustness, the HMM algorithms gradually fail. Rule-based methods, on the other hand, define as many rules as needed to cover the anticipated range of scenarios in the recognition task domain. This kind of approach lacks the flexibility of the statistical methods to effectively and properly handle different scenarios. Hence a need has emerged to incorporate acoustic, phonetic, and linguistic information into ASR systems in order to further discriminate between speech classes or phonemes with high confusion rates, especially in adverse environments (e.g., noise, transmission distortion, signal fading, etc.).

In this thesis, we study how the incorporation of static and dynamic distinctive acoustic, phonetic, and linguistic features can improve the recognition accuracy of traditional statistical-based ASR systems. The thesis research contributes directly to the front-end processing of the Automatic Speech Attribute Transcription (ASAT) project ([39], [40]), whose broad goal is to improve the performance of ASR systems by utilizing linguistically-based speech attributes and speech events in an architecture that integrates knowledge sources, models, data, and tools, ultimately combining the results with state-of-the-art HMM systems.

1.1 Major Issues

There are some major issues with front-end processing design of ASR systems, as is illustrated below.

1.1.1 Statistical data driven models or rule-based models

The first issue is the choice of statistical data driven models or rule-based models. In the statistical model approach to speech recognition, acoustic models are built based on statistical distributions and concentrations of the speech parameters, and speech recognition is a pattern matching process that maps a set of input speech parameters to a set of concatenated patterns corresponding to a set of sound/word trained models. Hidden Markov Models (HMM) and Artificial Neural Networks (ANN) are representatives of the statistical data driven approach to automatic speech recognition.

On the other hand, it is believed that the human brain recognizes and understands sounds by doing a distinctive feature analysis from the information going up the neural pathways and to the brain [66]. The set of distinctive features are believed to be relatively insensitive to noise, background, reverberation; thus they are robust and reliable. Knowledge-based ASR systems generally rely on knowledge gained from auditory models and attempt to detect distinctive features that enable the recognition of all of the phonemes of the English language. The ASAT system utilizes such a set of distinctive features to assist the traditional statistical recognition system, as explained in the next section.

As the traditional statistical ASR systems come to a bottleneck to improve the performance of distinguishing highly confusable sounds, several recent research efforts (e.g. [23], [36], [44], [46]) tried to incorporate other knowledge sources, e.g. linguistic features, into the statistical ASR systems, in an effort to provide additional knowledge in distinguishing them. In this work, we evaluate different sets of linguistic features and their role in the ASR process.

1.1.2 Detection vs. classification

The second issue is the choice of detection or classification in ASR. In detection-based methods, the speech waveform is processed frame-by-frame in a sequenced manner, and whenever the likelihood of a particular feature or phoneme is above a predetermined threshold, it is detected as present. The detection-based methods have no prior knowledge of the sentence. The detection based method needs the entire training data to train the detector, and the detector can be used to process the sequence of features in the entire sentence.

In classification-based methods, the speech waveform is segmented first, and the task is to classify the segment within a certain class of features or phonemes. The segmentation process is to identify word/syllable/phoneme boundaries usually using statistical methods such as Hidden Markov Models. In this thesis, we do not investigate on segmentation, but use hand-labels from the TIMIT database whenever we need pre-segmented training and testing data. In the classification process, the speech segments are classified within certain group of speech features or phonemes. In this type of method, we have prior knowledge of what group of features or phonemes the current segment belongs to. The training of the classification-based method only needs the speech parameters relevant to a group of features or phonemes within a class.

In our research, we investigate both feature detection and phoneme classification, and finally we combine the two approaches to improve both feature detection and phoneme classification performance.

1.1.3 Frame-based methods vs. segment-based methods

Frame-based methods and segment-based methods are two typical approaches to automatic speech recognition. The ASAT (Automatic Speech Attribute Transcription) methodology uses a detection method based on frame-wise speech attributes for phoneme detection ([39], [40]). Whenever the likelihood of a particular feature or phoneme is above a predetermined threshold, the feature or phoneme is detected as present. In classification-based methods, the task is to classify a segment within a given class of features or phonemes. Classification can be performed after segmentation (the so-called segmentation and labeling approach), or segmentation and classification can be performed jointly and suitably optimized.

In this thesis, we investigate frame-based methods for parallel speech feature detection and segment-based methods for hierarchical phoneme classification. Ultimately we combine frame-based and segment-based classification methods to enable a complete speech sound recognition system with improved overall classification and detection performance.

1.2 The Automatic Speech Attribute Transcription (ASAT) system

The ASAT system ([39], [40]) aims at using basic acoustic/linguistic units in a probabilistic event detection model to determine likelihoods of phonemes, words, and sentences, as shown in Figure 1.1 Bottom-up knowledge integration.

The goal of the ASAT front end processing is to estimate a set of attribute probability lattices, $P_0(A/F(t))$, which can be combined with information from higher level knowledge sources (e.g., a word lexicon) to create a phone lattice $P_1(P/F(t))$, a

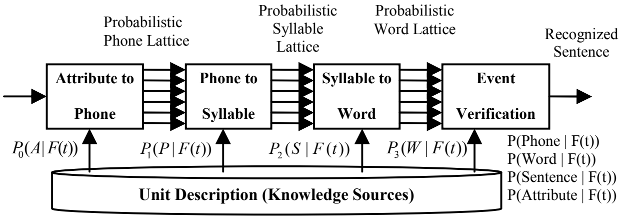


Figure 1.1 Bottom-up knowledge integration

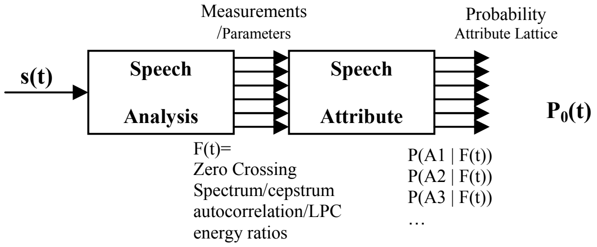


Figure 1.2 Front end processing

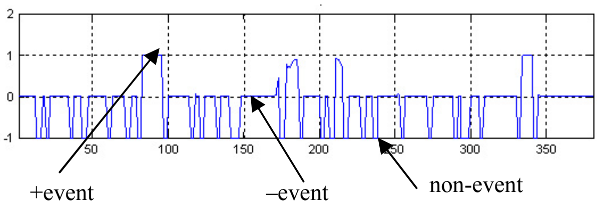


Figure 1.3 Illustration of events

syllable lattice $P_2(S/F(t))$ and a word lattice $P_3(W/F(t))$, which ultimately are used in a set of event verification modules to make a final recognition decision. Here $P_0(A/F(t))$ is the posterior probability of an attribute A given the speech parameters $F(t)$, $P_1(P/F(t))$ is the posterior probability of a phoneme P given $F(t)$, $P_2(S/F(t))$ is the posterior probability of a syllable S given $F(t)$, and $P_3(W/F(t))$ is the posterior probability of a word W given $F(t)$. Figure 1.2 shows the general front end processing system. Figure 1.3 illustrates the ASAT detection process. In this system, each speech parameter $F(t)$ is a direct measurement from the speech waveform, such as zero crossing rate or energy ratio. A speech attribute (also called a speech evidence, and called a speech feature in this thesis), A_i, is a piece of acoustic, phonetic or linguistic information that is estimated from the speech parameters. The speech attributes, e.g., voicing, nasality etc., distinguish the phonemes. An event, $e(t)$, is a stochastic process corresponding to each attribute that is used to make the decision that either the attribute is present (+) or absent (–) at time *t*, as shown in Figure 1.3. Such decisions can also be deferred to higher levels such as phones, syllables, words, and ultimately sentences, thereby mitigating the curse of error propagation that has plagued linguistically-based ASR systems over time.

In this thesis research, we estimate the likelihoods of both distinctive features and phonemes from speech, calculated over both single frames of short time spans, or segments of longer time spans.

1.3 Contribution

In this thesis, we aim at building the statistical and knowledge-based front-ends utilizing linguistic speech features in a bottom-up architecture. Variable lengths of segments are investigated, from single frames that cover a few milliseconds to segments that cover the entire phonemes and can be hundreds of milliseconds in duration. The major contributions of this work are:

(1) It is a bottom-up architecture that includes both feature detection and phoneme classification. Unlike the traditional ASR systems that directly calculate the likelihoods of phonemes given the speech parameters as input, in our system, we first detect the distinctive features, and then we classify phonemes according to the values of the features.

(2) Our work provides a method for combining statistical data-driven methods and knowledge-based approaches.

(3) The front-end processing of the ASAT system provides information in various levels of the knowledge hierarchy, which can be used to directly detect phonemes, words and sentences, or can be combined with the state-of-the-art HMM systems to improve speech recognition accuracy.

(4) We designed signal processing methods that integrate both single frames and segments for feature extraction and phoneme classification.

(5) We investigated the combination of frame-based and segment-based methods, and found that the performance of the combined systems was better than either pure frame-based attribute detection or pure segment-based phoneme classification.

1.4 Dissertation Outline

The remainder of the dissertation is organized as follows. Chapter 2 provides the background material and related research work done in this area.

In Chapter 3, we present an overall design of the front-end processing system. We first briefly illustrate the ASAT system, and then the present the overall architecture of attribute detection and phoneme classification. After that, a few major issues are discussed in the design process.

Chapter 4 mainly concentrates on frame-based methods in linguistic feature detection. We first test the phoneme boundary effect on voiced/unvoiced/silence classification performance. Then a set of Multi-Layer Perceptrons (MLPs) are used to estimate the likelihoods of a set of 14 Sound Pattern of English (SPE) features [9] using balanced training data. The SPE feature likelihoods are incorporated into the ASAT system to improve the phoneme recognition performance using Conditional Random Fields [38].

In Chapter 5 we investigate segment-based approaches to phoneme classification. A Time-Delay Neural Network (TDNN) toolkit is developed from scratch and is used in stop consonant classification. We show that transformation of the TDNN input parameters can improve the classification performance.

In Chapter 6, we combine the frame-based methods and segment-based methods in a phoneme classification task. Frame-based speech attribute detection using MLP is first converted to segment-based speech attribute and phoneme classification. Then the classification results from MLP and TDNN are linearly interpolated. Results are given to show that the combined system outperforms the pure segment-based TDNN classification system.

In Chapter 7, we combine the frame-based methods and segment-based methods in a feature detection task. We use segment-based TDNN to detect frame-wise parallel SPE features. We show that the linear combination of results from MLP detection and TDNN detection improved speech attribute detection performance.

Finally, Chapter 8 presents a summary of results and discusses future work.

Chapter 2

Background

Automatic Speech Recognition technology has evolved for more than five decades [54]. In the 1960's and 1970's, most speech recognition systems were based on acoustic/phonetic methods (a knowledge-based approach based on segmentation and labeling of the speech signal). In the 1980's the Hidden Markov Model (HMM) approach was introduced (a statistical, data-driven, approach to speech recognition) and rapidly became the method of choice for the past 20 years. Recently the trend is toward using a combination of statistical and knowledge-based systems. In this work, our proposed approach to incorporating acoustic/phonetic/linguistic information into ASR systems is based on the existing models of speech production and perception.

In this chapter, we review both the statistical, data-driven approach and the knowledge-based approach to speech recognition. Section 2.1 presents an overview of human speech production and acoustic/phonetic features. Section 2.2 reviews the underlying auditory models that are the basis for the frontend signal processing in most modern speech recognition systems. Section 2.3 reviews the statistical data-driven approaches that are most commonly used today. Section 2.4 presents some knowledge-based speech recognition systems. Section 2.5 summarizes the key findings from the background review.

2.1 Human speech production and acoustic/phonetic features

Humans produce speech by forcing air from the lungs through the vocal tract or nasal tract, where the air flow is modulated by the (time-varying) locations of one or more articulators (e.g., the tongue, jaw, teeth, lips, velum) thus producing various speech sounds [66]. The vocal tract consists of the glottis, the pharynx, and the oral cavity. The nasal tract consists of the velum and the nasal cavity.

The speech production process for creating a range of speech sounds can be viewed from a "manner of articulation" and "place of articulation" point of view.

"Manner of articulation" of a sound refers to the way that an articulator is used to generate a sound, with fixed vocal tract shape or changing vocal track shape, with either vibrating vocal cords or noise-like air flow, etc. Voiced speech sounds (e.g., vowels, diphthongs, nasals, stop consonants) are generated by chopping the air flow from the lungs into puffs of air, caused by the vibrating vocal cords, resulting in quasi-periodic speech waveforms. Unvoiced speech sounds (e.g., unvoiced fricatives, unvoiced stop consonants) are generated from turbulent air flow (i.e., without vocal cord vibration), resulting in a speech waveform with noise-like characteristics.

"Place of articulation" refers to the location of the narrowest constriction along the vocal tract during the course of sound creation. Consonants are generally associated with high degrees of vocal tract constriction at some point along the vocal tract, hence the term Place of Articulation. For example, the stop consonants /P/ and /B/ are labial stops sounds (point of maximum constriction is at the lips), /T/ and /D/ are alveolar stops (point of maximum constriction behind the teeth, at the alveolar ridge), and /K/ and /G/ are velar stops (point of maximum constriction at the velum). Vowels, diphthongs and semivowels are formed with mild constrictions caused by the tongue hump location. For example, /IY/ is produced by a high position of the tongue hump, /EH/ is formed by a medium tongue hump position, and /AE/ is formed by a low tongue hump position.

Acoustic/phonetic features for various speech sounds are based on the different manners and places of articulation, and provide information for accurate classification of a broad range of phoneme classes. In our work, we use both manner of articulation and place of articulation along with other linguistic features to help improve the ASR performance.

The sounds of the English language can be characterized by a set of distinctive features, many of which are motivated by the speech production model. Stevens' acoustic phonetic theory [66] provides detailed descriptions and analyses of human speech production.

Depending on how many different values each distinctive feature can have, the feature sets are categorized into binary valued features (the feature is present or absent) and multi-valued features (the feature has several subclasses). Jakobson and Halle [32] defined 12 distinctive features for each phoneme, including: (1) vocalic/non-vocalic; (2) consonantal/non-consonantal; (3) interrupted/continuant; (4) checked/unchecked; (5) strident/mellow; (6) voiced/unvoiced; (7) compact/diffuse; (8) grave/acute; (9) flat/plain; (10) sharp/plain; (11) tense/lax; (12) nasal/oral.

This set of distinctive features is a universal binary set of speech production features, and they can characterize up to 2^{12}=4096 phonemes, but in any existing language there are significantly fewer phones and phonemes. The English language can be represented by 9 pairs of these features, but the feature representation is not efficient, since 2^9=512, which is far larger than the number of phonemes in the English language (39-61 phonemes).

The Sound Pattern of English (SPE) set of distinctive features provide another way for defining phoneme classes [9], and the SPE distinctive features for each of the 61 TIMIT phonemes include: (1) vocalic; (2) consonantal; (3) high; (4) back; (5) low; (6) anterior; (7) coronal; (8) round; (9) tense; (10) voice; (11) continuant; (12) nasal; (13) strident; (14) silence.

(See Appendix A for a detailed description of the SPE features for the 61 phonemes).

The multi-valued feature sets are defined by several broad features, and each feature takes on more than one value, as shown in Table 2.1.

Table 2.1 Multi-valued features ([34], [35])

Feature	**Possible values**
centrality	central, full, nil
continuant	continuant, noncontinuant
frontback	back, front
manner	vowel, fricative, approximant, nasal, occlusive
phonation	voiced, unvoiced
place	low, mid, high, labial, coronal, palatal, coronal-dental, labio-dental, velar, glottal
roundness	round, non-round
tenseness	lax, tense

2.2 Speech Perception and Auditory Models

2.2.1 Concept

Speech is understood by human beings via listening comprehension and visual comprehension. Visual effects, like lip reading, gesture, etc., can greatly facilitate the understanding of speech, but such side information is beyond the scope of this thesis research. In this section we describe our limited knowledge of how humans recognize/perceive/understand speech via signal processing in the ear and the associated neural pathways. State-of-the-art Automatic Speech Recognition systems are still trying to achieve comparable performance to Human Speech Recognition (HSR) [43].

Auditory models have been created that attempt to mimic human speech perception. When the speech pressure wave reaches the listener, it is first processed by the basilar membrane which performs a spectral analysis (on a highly non-uniform frequency scale) on the speech waveform. The spectral analysis performed by the basilar membrane uses a non-linear frequency scale. The mel frequency scale and Bark frequency scale were created to match this non-linearity of the spectral processing mechanism. The mel frequency scale is calculated using the relation:

$$\text{Pitch in mels} = 3323 * \log10(1+F/1000) \qquad (1\text{-}1)$$

whre F is the frequency in Hz. The Bark frequency scale is similarly defined.

The use of Mel Frequency Cepstral Coefficients (MFCC) is the most popular speech parameter set in modern speech recognition systems, especially in HMM based speech recognition systems [12]. The use of Perceptual Linear Prediction (PLP) coefficients by Hermanski et al [24] enabled an improvement in performance over that obtained using MFCC feature vectors. The PLP system used a Bark scale critical band with non-linear frequency resolution, and it incorporated unequal sensitivity of human hearing versus frequency, used cubic root compression to mimic the intensity-loudness non-linearity and also performed some simple spectral smoothing. The RASTA-PLP method, as proposed by Hermanski and Morgan [26], is the most important variant of the original PLP [24]. RASTA stands for RelAtive SpecTrAl Technique, and it used a special filtering of different frequency bands. Hönig et al compared MFCC and PLP and proposed a modification of PLP that took advantage of the relative strengths of both MFCC and PLP feature vectors [30].

The human ear has a set of characteristic ways of responding to sound waves. The human ear doesn’t respond equally to loudness in different frequency ranges. The effect whereby loud tones (or noise) mask adjacent frequency signals in such critical frequency bands is called auditory masking. A strong masking signal changes the hearing threshold in the vicinity of the masker, and effectively masks signals that fall below the masked threshold and are within a critical bandwidth of the masking signal. The masking effect also exists in the time domain as well as in frequency, where a strong temporal masking signal can pre-mask up to 30 msec. of signal, and post-mask up to 200 msec. of signal [52].

From the analysis of how human auditory systems perceive speech and other acoustic signals, it is clear that in speech recognition we need both short (20 msec for phoneme-duration signals) and long (200 msec for syllable-duration signals) segments of speech for reliable recognition and perception. We also see that the temporal structure of speech is important for some sounds, and the spectral structure is important for other sounds (especially vowels with a well defined formant/resonance structure). We also find that we need dynamic (e.g., first and second order derivative) features of speech for reliable detection of speech sounds in noise, or in reverberant locations.

2.2.2 Auditory models

Auditory models provide various ways to transform the speech wave into acoustic parameter vectors and build up the front-end processing in ASR. There have been several recent research efforts that designed various auditory models.

Seneff modeled the hair cell synapse behavior in an auditory model that captured features of transformation from basilar membrane vibration to probabilistic response properties of the auditory nerve fibers [60]. The Seneff model was able to model short-time adaptation but not long term adaptation. The model had two outputs: the mean rate output that measured the firing rates of auditory nerve fibers and the spectral energy in each channel; the synchrony spectrum output that measured the synchrony of fine temporal structure of each channel, as shown in Figure 2.1. In this model, the generalized synchrony detector implemented the "phase locking" property of nerve fibers and enhanced spectral peaks due to vocal tract resonances. The synchrony detector also provided spectral distinctness in high frequency regions that was useful for classification

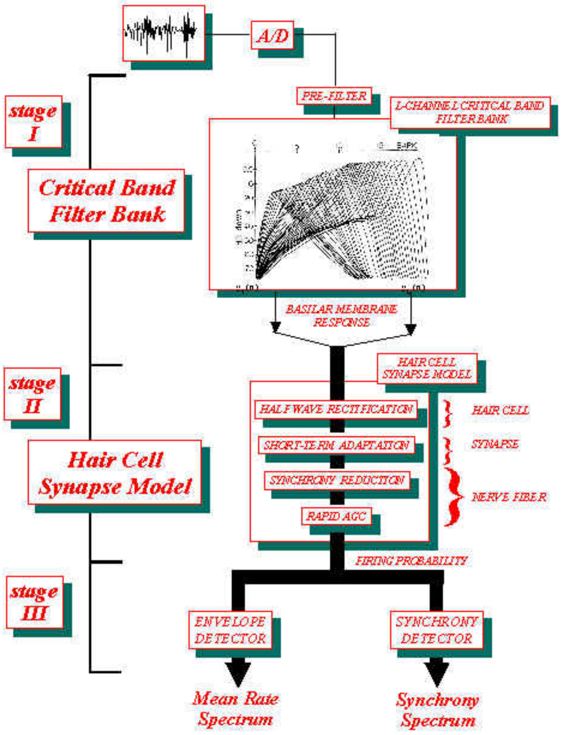


Figure 2.1 Seneff's auditory system [60]

of fricatives and stops, which was later on used in Ali's auditory system for classification of fricatives and stops ([1], [2], [3], [4], [5]).

In another auditory model as shown in Figure 2.2, Lyon modeled the behavior of the cochlea as a non-linear, compressive, cascaded filter bank ([45], [61], [62]). In this model, the outer and inner ear provided pre-emphasis, and the inner hair cells provided half-wave rectification. There were four active and passive Automatic Gain Control phases that modeled the masking effect, and the value of each gain depended on the time constant from preceding output samples of adjacent channels. The output of the model approximated the neural firing rates. The model first computed the cochleagram that depicted the cochlea place's output over time in different frequency channels. The correlogram was then computed to show the short-time autocorrelation of each output channel, and was a 3-D representation of time, frequency and the autocorrelation lag. The correlogram showed concentration of energy in frequency as well as periodicity in the cochlea channels in the autocorrelation lag.

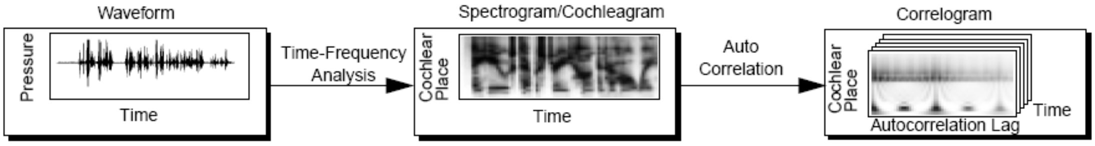


Figure 2.2 Lyon's auditory system [62]

Lyon's auditory model was used to classify vowels and stop consonants for the TIMIT database [69]. The recognition results were approximately 60% accuracy for vowels and 70% accuracy for stops.

Using auditory models in speech recognition, the input parameters can be transformed and combined to improve recognition performance ([63], [64]), using unsupervised transformations such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA) and when using supervised transformations such as Linear Discriminant Analysis (LDA) and Multi-Layer Perceptrons (MLP). Extensive experimentation has shown that all the transformed parameters outperformed the baseline systems which used MFCC and first order deltas in a phoneme recognition task, but couldn't outperform the system with pure PLP parameters and their deltas in a number recognition task. Further improvements were achieved by concatenating the transformations together or combining the transformations with the original PLP parameters [64]. In our study, because of the enormous type and number of speech parameters available, we also needed to transform the original speech parameter vectors into more compact sets via transformations of the type mentioned above.

2.3 Statistical data driven approaches

The most widely used statistical approach to Automatic Speech Recognition (ASR) systems is basically a pattern recognition process. Assume the speech signal, *S,* is characterized by a spoken sequence of words, *W*, and *S* is represented by the acoustic feature vector *X*. In order to decode the speech signal *S* into the sequence of words *W*, a *maximum a posteriori* Bayes formulation is used, giving:

$$\hat{W} = \arg\max_{W} P(W/X) = \arg\max_{W} \frac{P(X/W)P(W)}{P(X)} = \arg\max_{W} P(X/W)P(W) \qquad (2\text{-}1)$$

where $P(W/X)$ is the probability that the word sequence *W* was spoken, given the observed acoustic feature vector *X, P*(*X/W*) is the probability of the feature vector *X,* given the spoken word sequence *W,* (we call this the acoustic model) and *P*(*W*) is the apriori probability that the word sequence *W* was spoken (we call this the language model). We omit the term *P*(*X*) because it doesn't affect the argmax calculation on *W*.

Hidden Markov Models (HMM) [53], Dynamic Bayesian Networks (DBN) (e.g., [44], [76]) and Artificial Neural Networks (ANN) (e.g., [7], [55]) are representative statistical pattern recognition approaches that have been applied to speech recognition problems.

2.3.1 Hidden Markov Models

A Hidden Markov Model (as used for speech recognition) is a sequence of states that correspond (in a statistical sense) to a sequence of sounds used in the production of a spoken input to a machine [53]. Generally an HMM for speech recognition is a left-to-right process where each basic speech unit (e.g., a phoneme or a word) is represented by a sequence of states. The topology of a 3-state left-to-right HMM (as might be used to represent a phoneme in the language) is illustrated in Figure 2.3.

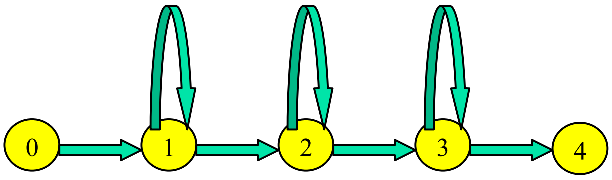


Figure 2.3 HMM diagram

Assume there are *N* states in an HMM and each state has *M* distinct observation symbols. We observe that the HMM system is in one of the *N* states at every time instant *t,* and we use the notation q_t to denote the state at time *t*. Further we use the notation o_t to denote the observation at time *t,* recalling that it must be one of the *M* symbols observed in each state. Finally we denote the observed symbols as v_k for the *v*-th symbol in state *k.*

The HMM consists of three sets of probabilities, denoted as:

$$\lambda = (\boldsymbol{A}, \boldsymbol{B}, \pi) \tag{2-2}$$

where $\boldsymbol{A}$ is the state-transition probability distribution $\boldsymbol{A}=\{a_{ij}\}$ where

$$a_{ij} = P[q_{t+1}=j|q_t=i], \qquad 1\leq i, j\leq N \tag{2-3}$$

$\boldsymbol{B}$ is the observation symbol probability distribution, $\boldsymbol{B}=\{b_j(k)\}$, where

$$b_j(k) = P[\boldsymbol{o}_t = \mathbf{v}_k \mid q_t = j], \qquad 1\leq k\leq M \tag{2-4}$$

π is an initial state distribution $\pi = \{\pi_i\}$ in which

$$\pi_i = P[q_1 = i], \qquad 1\leq i\leq N \tag{2-5}$$

By way of example, to use such an HMM system for isolated word recognition [53] of a *V*-word vocabulary, a unique HMM model can be built for each individual word. Given a temporal sequence of observations (the acoustic feature vectors) $O=\{O_1,O_2,\ldots,O_T\}$ (where *T* is the number of frames in the spoken word), corresponding to one of the spoken words in the vocabulary, the recognition task is to find the word *v* that gives the highest likelihood score, i.e.,

$$v^* = \arg\max_{1\leq v\leq V}\left[P(O/\lambda_v)P(w_v)\right] \tag{2-6}$$

where $P(w_v)$ is the *a priori* probability of the word w_v.

In finding the highest likelihood score a Viterbi search [54] is normally used. This search procedure finds the best alignment path between the observation sequence of acoustic feature vectors and the HMM state model. This optimal alignment path is obtained by sequentially finding the best score along each path ending in state *i* at time *t.*

We denote this best score at time *t* and ending at state *i* as $\delta_t(i)$ and we recursively compute it as:

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] \cdot b_j(o_{t+1}) \tag{2-7}$$

where

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_{t-1}, q_t = i, o_1 o_2 \ldots o_t \mid \lambda] \tag{2-8}$$

is the best state sequence **q** = $(q_1\ q_2\ \ldots q_t)$ ending in state *i* for the observation sequence *O* = $(o_1, o_2, \ldots o_t)$.

Equation (2-7) is applied recursively until the end of the observation sequence, *t=T*. The best path is obtained by back tracking from *t=T* to *t=1*.

The training of an HMM can use the Baum-Welch algorithm [53], in which the likelihood of $P(\boldsymbol{O} \mid \lambda)$ is iteratively and locally refined using the method of Expectation Maximization (EM) [13].

Since the concept of using HMMs was introduced into ASR, there have been enormous efforts in building speech recognition systems using HMMs. Reference [72] gives a good review of the use of HMMs in large vocabulary continuous speech recognition systems.

In our ASAT project, the baseline system is an HMM speech recognizer. We aim to improve the HMM recognizer performance by incorporating information provided by distinctive features.

2.3.2 Artificial Neural Networks

An Artificial Neural Network (ANN) is also called a connectionist model. Figure 2.4 shows the computation of a neuron [43], where there are *N* inputs of $x_1, \ldots x_N$, *N* weights $W_1, \ldots W_N$, and offset Φ, where $x_1, \ldots x_N$ and $W_1, \ldots W_N$ can be any real number, and Φ is a constant. The nonlinear function *f* is used to limit the output to a specified range, typically [-1, +1]. *f* is usually the sigmoid function:

$$f(x) = \frac{1}{1+e^{-\beta x}}, \qquad \beta > 0 \tag{2-9}$$

A neural network can be used to estimate the posterior probabilities of speech classes [55]. It allows direct estimation of probabilities of various phoneme classes based on a large number of input speech parameters. When training ANNs to learn the input patterns in speech, the patterns are represented via a series of hidden nodes in each layer. There are a range of neural network toolboxes available, like the NICO toolkit [49], the CSLU HMM/NN toolkit [11], the Netlab toolbox [48], etc.

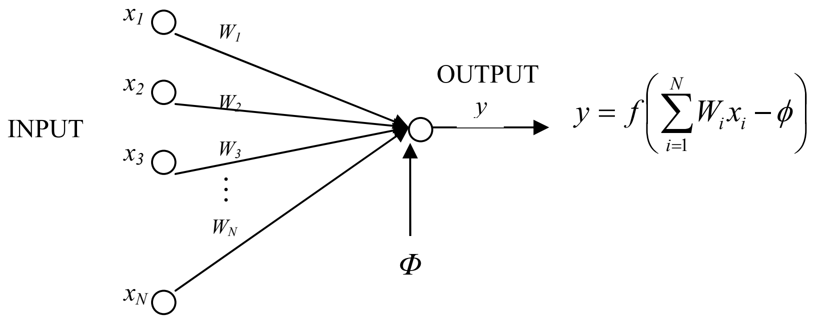


Figure 2.4 Simple computation of a neuron

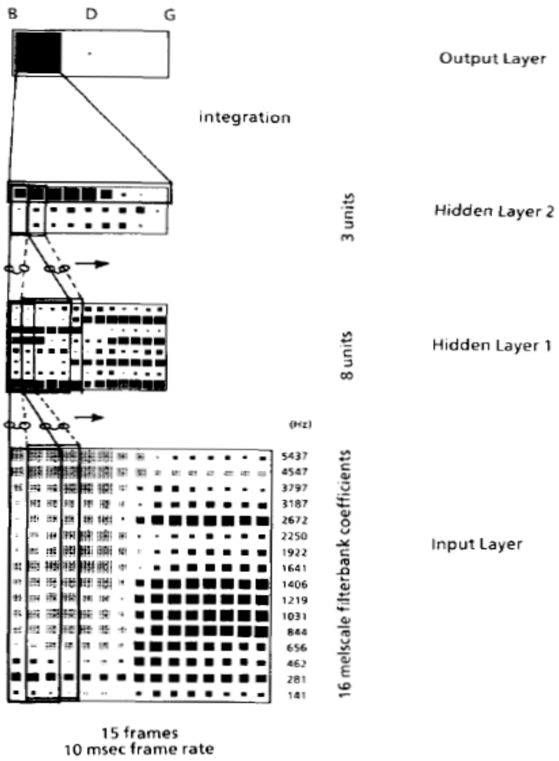


Figure 2.5 TDNN architecture [70]

A special class of neural networks, called Time-Delay Neural Networks (TDNN), was proposed by Waibel et al [70], and has been shown to be effective for classifying dynamic sounds such as voiced stop consonants in Japanese. The TDNN network introduces delays into the input of each layer of a regular Multi-Layer Perceptron (MLP), and relates the current input to the past history of events in the segment-based input feature set. Short duration features are formed at the lower layer(s), while higher levels tend to integrate longer input time intervals and form more complex and longer duration features. Figure 2.5 shows the architecture of a typical TDNN (as defined by Waibel et al). In Waibel's work on voiced stop sounds (/B/, /D/ and /G/) classification, the TDNN looks for a stop event during a 30 ms time span within a 150 ms segment; the stop event can occur anywhere within the 150 ms time span of the input signal. In this sense, the TDNN is shift invariant and doesn't require precise segmentation or alignment of the input.

In [71], several TDNNs were connected together to recognize the complete set of Japanese phonemes. Many variations of the original TDNN have been proposed and studied, including a Frequency-time-shift-invariant TDNN (FTDNN) [59], an Adaptive Time-Delay Neural Network (ATNN) [42], etc.

2.3.3 Dynamic Bayesian Networks

The method of Dynamic Bayesian Networks (DBN) [65] is gaining popularity in automatic speech recognition. A DBN can be seen as a generalization of an HMM in the way that at a given time, there can be multiple state variables with arbitrary dependencies. Suppose we want to model a set of variables $\boldsymbol{X}$=(x_1, x_2, ...x_N), and the variables are represented by a directed acyclic graph with each node denoting one variable, and the joint distribution of $\boldsymbol{X}$ can be written as a Bayesian Network (BN) of the form:

$$P(x_1, x_2 ... x_N) = \prod_{i=1}^{N} p(x_i \mid parents(x_i)) \qquad (2\text{-}10)$$

where $parents(x_i)$ are the parent nodes of the current node x_i. A Dynamic Bayesian Network depicts a process in which each frame is represented by a Bayesian Network.

In distinctive feature detection, a DBN provides a way to represent phonemes or words using multiple features as multiple states.

Zhang, Diao, et al discussed applying DBN on synchronous and asynchronous multi-stream models [73]. The multiple streams include a range of acoustic features such as MFCC, PLP, RASTA, JRASTA, and Wide-band MFCC. The recognition was done via whole word models consisting of a fixed number of states. The synchronous multi-stream model combination required that each of the 5 streams be strictly aligned, and all the 5 streams share the same state. The asynchronous model relaxed the strict alignment by allowing different streams to have different states. The synchronous multi-stream model showed significant improvement on the Aurora 2.0 noisy speech task [28] over single stream models.

2.3.4 Hybrid systems

Several types of hybrid systems have been proposed in order to improve speech recognition accuracy, such as hybrid HMM/NN systems and hybrid HMM/BN systems.

The purpose of hybrid HMM/NN systems is to take advantage of the time-alignment capability of HMM systems and the discrimination power of neural network systems. Bourlard and Morgan described a hybrid connectionist-HMM system in [7]. The NN/HMM system used neural networks to estimate the posterior probabilities of each phonetic class instead of using the Gaussian Mixture Models (GMM) that is typically used in HMM systems. The hybrid system was shown to achieve comparable performance to HMM systems based on GMM but with a simpler system implementation.

The tandem acoustic model [25] is another modification of the traditional NN/HMM hybrid. The tandem system consisted of a neural network and a HTK (Hidden Markov Model Took Kit) decoder [31]. The input to the neural network was 9 frames of MFCCs at a 10 ms frame rate (i.e., a segment of duration 90 ms). The output of the tandem system was the posterior probability of a set of context independent phonemes. The neural network was trained using a minimum cross entropy criterion. A logarithmic compression was applied to the outputs to achieve better contrast between the target phoneme and the competing phonemes. A Karhunen-Loeve transform was applied to the output to reduce feature vector dimensionality. The output was then fed to a Gaussian mixture-based HTK system. The resulting system performance had a 35% error reduction as compared with a baseline HTK (HMM) system. In [16] the combination of a tandem acoustic model and a Gaussian likelihood space using PLP features was applied to the problem of recognition of the TIMIT database. The resulting system was reported to have 18% error reduction as compared with using the HTK decoder based on MFCC coefficients only.

A combination of an HMM recognizer and a Recurrent Neural Network (RNN) [57] was used to capture context information in speech, since the articulation of a phoneme is always affected by many contextual variables, such as coarticulation, speaking rate, etc. In this study, the RNN was used to estimate the phone posterior probabilities. The MLP had two sets of output: one set for the output vector *y(t)*, another for the state *x(t)*. The state output was fed back to the input. In this way the RNN related the history of events into an MLP structure. Cross-entropy was used as the objective function and the resulting system achieved significantly faster training than using least mean square as the objective function. The output of the RNN was fed to a Markov model that had one state for every phone and a transition between any pair of phones was allowed. The RNN/HMM approach performed better than the monophone HMM system but the performance was still not comparable to a conventional context-dependent triphone HMM system.

The use of Hidden Neural Networks (HNN) was introduced in [56]. In this work, the neural networks were used to estimate the emission probabilities in a state and the transition probabilities from one state to another. The training of the HNN maximized the Conditional Maximum Likelihood, a process which was similar to maximizing the Mutual Information for a fixed language model. When used to recognize the 5 broad phoneme classes of vowels, consonants, nasals, liquids and silence, the HNN achieved 84% class accuracy. When using the HNN to recognize the 39-phoneme alphabet, the HNN achieved 69% phoneme accuracy.

The hybrid combination of TDNN and Dynamic Time Warping (DTW) for speech recognition was discussed in [21] and [27]. The architecture was called a Multi-state TDNN (MS-TDNN). This system was an extension of the original TDNN system that recognized individual phonemes and the TDNN/DTW architecture performed recognition on whole sentences. The MS-TDNN had five layers, namely the input layer, the hidden layer, the phoneme layer (second hidden layer), the DTW layer and the word layer. The system was trained in two stages. The phoneme level training used the first three layers, and provided frame-based outputs. The word level training aligned the phoneme path and got the correct path of phonemes to form words. Their results showed that the MS-TDNN performed better than HMM, mixed TDNN/HMM and linear predictive NN for a specific task domain recognizer.

The dynamic Bayesian networks were very popular in estimating acoustic features from speech. The hybrid of DBN and other algorithms were introduced in the feature based systems discussed in Section 2.3.

2.4 Knowledge based systems

Our proposed approach to incorporating acoustic/phonetic/linguistic information into ASR systems is based on the existing models of speech production and perception. In this section, we briefly introduce some of the present knowledge based systems, mainly about distinctive feature detection in recognition of phonemes.

There have been many research efforts that tried to utilize knowledge features into speech recognition. Morgan et al summarized the state-of-the-art in this area [46]. In these studies the investigators used long windows (up to 500 ms) to capture temporal trajectories, and conventional short windows and features to create a multi-stream multi-rate system that characterized both phoneme and syllable structures. They also used all-pole models to represent temporal features and spectral trajectories.

Dynamic Bayesian Networks is another knowledge representation that is gaining popularity in feature-based speech recognition systems. Livescu et al applied the DBN to speech feature recognition [44]. The features were not directly observable by the listener (i.e., they were hidden), and these features corresponded to the states in the DBN. The investigators didn't allow dependencies between features, but assumed the feature value independently changed over time according to the phoneme and the previous feature value, as shown in Figure 2.6, where the output *O* corresponded with the *N* acoustic features $A_1, \ldots A_N$, which was generated from a specific state *S*. Their feature set consisted of 8 multi-valued features: voicing, velum, manner, place, retroflex, tongueBodyLowHigh, tongueBodyBackFront, and rounding. The Aurora noisy speech corpus of connected digit utterances was used for training and testing. The combination of the deterministic hidden feature model and the phone model performed better than when using the phone model HMM alone, across a range of noise levels. However the non-deterministic hidden feature model didn't show any significant improvement over the HMM model.

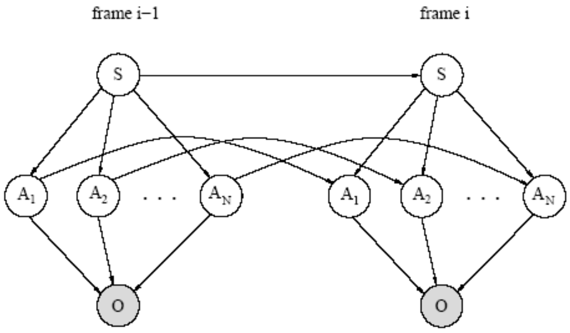


Figure 2.6 Hidden feature model [44]

Frankel and King proposed a hybrid ANN/DBN approach to recognize articulatory features [17]. They explicitly modeled the mutual dependencies of features on each other (the presence of one feature depends on other features), and used a DBN to model the inter-dependencies of the features, as shown in Figure 2.7. The observation was generated by feature states, and each feature value was generated by a set of templates, as shown in Figure 2.8. They used six features, as shown in Table 2.2. Recurrent Neural Networks were used to estimate the prior probabilities of each feature. The system was trained in a synchronous mode and in an asynchronous mode in order to estimate the conditional probabilities between parent and child in the Bayesian network.

The ANN/DBN system performed better than the ANN/HMM system for estimating the features (87.8% vs. 83.5%). The synchronous mode performed comparably with the asynchronous mode. The ANNs were used to provide Virtual Evidence (VE) [51], and the VE and the asynchronous DBNs were used to realign the training labels. Their results showed that after realigning for one iteration, the system performances increased from 88.2% to 93.5%, a significant improvement after refining the model.

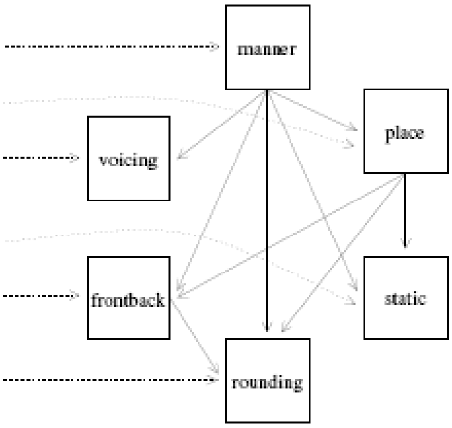


Figure 2.7 Inter-feature dependencies [17]

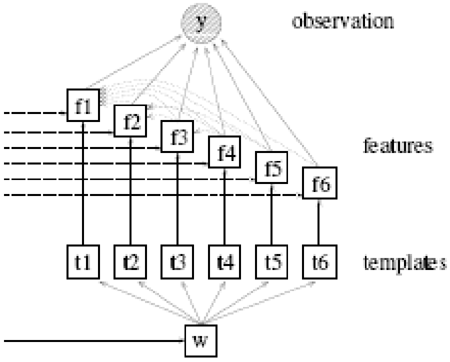


Figure 2.8 One time-slice of the articulatory feature based recognition system [17]

Table 2.2 Multi-valued articulatory features [17]

feature	values	cardinality
manner	approximant, fricative, nasal, stop, vowel, silence	6
place	labiodental, dental, alveolar, velar, high, mid, low, silence	8
voicing	voiced, voiceless, silence	3
rounding	rounded, unrounded, nil, silence	4
front-back	front, central, back, nil, silence	5
Static	static, dynamic, silence	3

Another recent trend in feature-based ASR systems is the use of landmark-based speech feature detection algorithms as summarized in [23]. A landmark refers to the acoustic cues of abrupt changes in articulation, including consonant closures and releases, syllable peaks and dips, etc. [66]. The level changes in different energy bands are possible landmarks. A special dictionary was created to represent words and phonemes using landmarks [67]. The landmark based systems all used Support Vector Machines (SVM) [10] to transform the acoustic parameters into landmarks [33]. The acoustic parameters included MFCC, spectral shape and a range of acoustic-phonetic parameters. A Dynamic Programming (DP) algorithm [58] was used for aligning the detected words with the landmark lexicon. A Dynamic Bayesian Network (DBN) was used to represent the transitions between acoustic-phonetic features.

In another study, Kirchhoff used Hidden Markov Models and Artificial Neural Networks to detect articulatory features [36]. This work showed that incorporating the distinctive features into traditional phone-based systems improved recognition robustness in noisy environments.

King and Taylor used Artificial Neural Networks to detect phonological features in continuous speech [35]. They compared the binary SPE features with Multi-Valued

(MV) features and with the Government Phonology (GP) primes [22]. Their results showed that the accuracy of SPE features was better than the MV features (92% vs. 86%), but the phoneme accuracy were basically the same (59% vs. 60%). This was due to the fact that there were fewer features used to decide a phoneme using the MV system. In [34], King et al. compared HMM and ANN for detection of the SPE and MV features. They showed that the ANN performed better than the HMM for both SPE and MV feature detection.

Ali et al used auditory models with specific measurements for detection of specific phoneme classes ([1], [2], [3], [4], [5]). Feature parameters from Seneff's auditory model were used as the front end processing for these systems. The average localized synchrony detection (ALSD) [1] proved to be robust for detection of formants, and it also reduced occurrences of spurious spectral peaks. For recognition of 4 vowels (/ae/, /iy/, /aa/ and /uw/) in the TIMIT database, ALSD obtained better performance than when using the mean rate detector and the Generalized Synchrony Detector (GSD) as the speech feature set. The accuracy was approximately 81% for clean speech, 79% for SNR=10 dB, which represented an improvement in performance of approximately 14% above that obtained using GSD methods. The ALSD also outperformed the mean rate and GSD in estimating the place of articulation detection for vowels, fricatives, and stops [2].

Ali's fricative detection [3] consisted of 2 parts: a voicing detection system and a place of articulation detection system. The duration of the unvoiced portion (DUP) was used for voicing detection, and provided about 93% accuracy. The place of articulation detection system used Maximum Normalized Spectral Slope (MNSS), the location of the most spectral slope (MDS), the location of the most dominant peak (MDP), Spectral Center of Gravity (SCG), and the dominance relative to the highest filter (DRHF). The performance for place or articulation detection was 91%. The overall performance was 87% fricative detection accuracy. MNSS discriminated much better on /f, v, th, dh/ than Relative Amplitude (RA), but not for /sh/ and /zh/. This showed that RA and MNSS provided information useful for different fricative groups.

For stop consonant detection [4], the voicing detector used 3 features of voicing during closure (prevoicing), voicing onset time (VOT), and closure duration, and obtained 96% accuracy. The place of articulation detection used the burst frequency (BF), the second formant of the following vowel, the MNSS, the burst frequency prominence, the formant transitions before and after the stop, and the voicing decision, and obtained 90% accuracy. The overall performance on stop consonant detection was 86% accuracy. The experiments were only performed on initial and medial stops in the TIMIT database.

Ali's overall phonetic feature detection system is depicted in Figure 2.9 [5].

The goal of the Automatic Speech Attribute Transcription (ASAT) project ([39], [40]) is to capture and utilize information that is readily measured from the speech waveform, including linguistic, temporal and spectral information with the goal of improving the accuracy of existing speech recognition systems, especially in noisy and reverberant environments. The information is collectively called the set of *Speech Attributes*, and such information has been shown to be useful in speech recognition and speaker identification systems. The ASAT system tries to integrate various knowledge

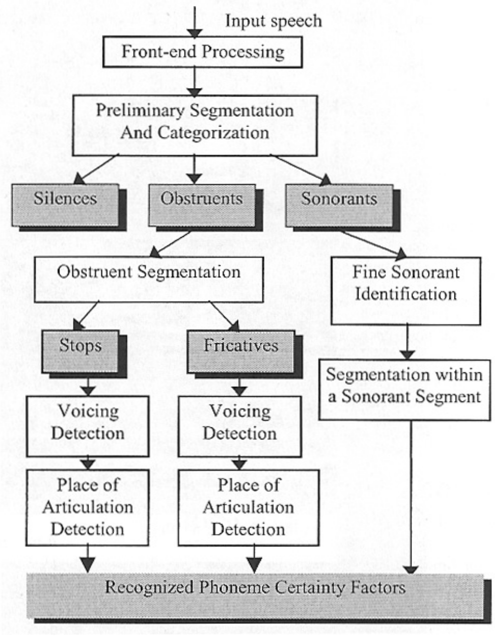

Input speech
Front-end Processing
Preliminary Segmentation And Categorization
Silences
Obstruents
Sonorants
Obstruent Segmentation
Fine Sonorant Identification
Stops
Fricatives
Segmentation within a Sonorant Segment
Voicing Detection
Voicing Detection
Place of Articulation Detection
Place of Articulation Detection
Recognized Phoneme Certainty Factors

Figure 2.9 Ali's General phoneme detection system [5]

sources in a bottom-up architecture. Initial studies at using such speech attributes were performed by Li et al (who used the speech attributes to rescore the HMM detection output [41]), and Dusan (who estimated speaker–specific vocal tract lengths from the speech sound [14]). Our goal is to create a system that outputs reliable and useful information at various levels of the speech knowledge hierarchy.

2.5 Conclusion

While the HMM-based systems have been highly successful, the rate of progress in speech recognition systems has slowed down in recent years as researchers have tried to find ways to make recognition systems be robust to noise and reverberation that exists in real speaking environments. The results presented in this section demonstrate that the use of acoustic/phonetic/linguistic features provides useful cues for speech recognition, and the combination of statistical and knowledge-based systems should be able to utilize the additional information and improve performance of the statistical model for speech recognition. This thesis aims at building knowledge-based front ends that utilize information in speech segments of varying lengths (from a single frame to tens of frames), and detects speech attributes and phonemes in a specified knowledge hierarchy. The remainder of this thesis discusses the methods we have investigated for detecting the speech attributes and utilizing that knowledge to estimate the probabilities of speech sounds (primarily phonemes).

Chapter 3

Overview of Distinctive Feature Detection and Classification Using Frames and Segments

The goal of this thesis research is to detect and recognize phonemes based on speech attributes that contain information at different levels of the speech production/perception hierarchy. The resulting attribute detection system contributes to the frontend processing of the ASAT project. The architecture of the complete ASAT attribute detection system is shown in Figure 3.1. The input to this system is the speech signal $s(t)$, $t=1,2,\ldots,T$, which is passed through a set of M signal processing modules giving a set of processed speech signals $s_i(t)$, $i=1,2,\ldots,M$. Subsequently the M processed speech signals are passed through N speech parameter detectors to get speech parameters $SP_j(t)$, $j=1,2,\ldots,N$ (e.g. MFCC, VOT), and the speech parameters are combined to form L speech features (attributes) $A_k(t)$, $k=1,2,\ldots,L$. The speech attributes are then used to calculate the likelihoods of phonemes or to help to rescore the phoneme lattice obtained from an HMM-based decoder.

3.1 The Major Issues

Using this framework we attempted to answer the following key questions, namely:

(1) *What parameters to measure*

There exists numerous temporal and spectral speech parameter sets that have been extensively studied for use in speech recognition systems, and our goal is to choose the parameter sets that would be most effective (and efficient) for estimating the speech attributes of interest. The range of speech parameters that we have investigated (or that we are in the process of investigating) include a range of short-time and long-time parameters, based both on temporal processing methods as well as spectral processing methods. Table 3.1 shows the range of speech parameters that are being investigated at this time—many of which are already included in the ASAT front end processing system.

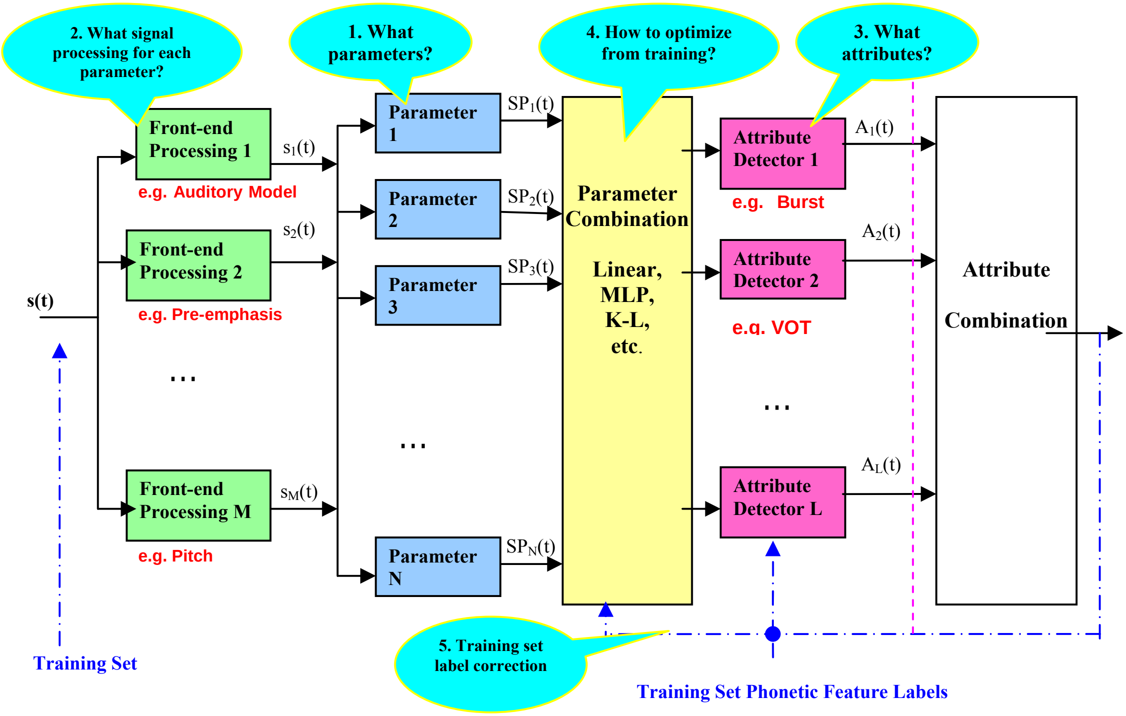


Figure 3.1 Overall diagram of the front-end processing

Table 3.1 Speech parameter groups

	Short-time	Long-time
Temporal	voiced/unvoiced/silence Pitch Segmental SNR	VOT burst duration unvoiced duration syllable duration

Spectral	MFCC
	Spectral flatness
	Relative band energies
		delta(MFCC)
		delta-delta(MFCC)

If all parameters are linearly independent (highly unlikely) and contain useful speech information we can combine them in a concatenative manner, as shown in Figure 3.2, thereby giving a super-parameter vector from which we could estimate a wide range of speech attributes.

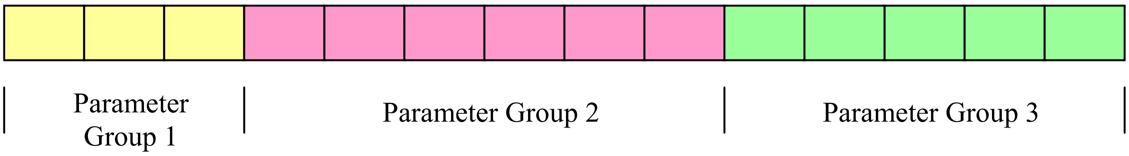


Figure 3.2 Parameter group concatenation

One of the problems with combining parameters using the method of Figure 3.2 is that any parameter with a large dynamic range can numerically swamp out the contributions of all other parameters; hence some type of normalization must be applied to the super-parameter vector. Assume $x_i(t)=SP_i(t)$ is one parameter of the super-vector, with average value, $\bar{x}_i$, and standard deviation, σ_i, computed as:

$$\bar{x}_i = \frac{1}{T}\sum_{t=1}^{T} x_i(t) \tag{3-1}$$

$$\sigma_i = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\left(x_i(t)-\bar{x}_i\right)^2} \tag{3-2}$$

where T is the total number of frames of all input sentences. Thus a simple form of parameter normalization is to subtract out the mean and divide by the standard deviation, giving a zero mean, unity standard deviation parameter of the form:

$$\hat{x}_i = \left(\frac{x_i - \bar{x}_i}{\sigma_{i_i}}\right) \tag{3-3}$$

If any such normalized parameter is highly correlated with any other normalized parameter we are doing unnecessary computation. Therefore the dimensionality can be reduced by using standard statistical methods (for each group), e.g., PCA, K-L, etc. The dimensionality can also be reduced by preprocessing groups of parameters to minimize correlation among coefficients of each group.

(2) *What signal processing algorithm should be used for each parameter*

Often there exist multiple signal processing algorithms for different speech parameters, e.g., to calculate formants or pitch periods we could use cepstral or LPC methods. Thus we have to make choices for each parameter set.

(3) *What attributes to estimate*

When defining a feature set for detecting speech attributes, we need to pay attention to the following criteria:

- the feature set should be capable of uniquely determining a phoneme;
- the feature set should be meaningful in terms of the relations to a range of acoustic/phonetic/linguistic attributes;
- the feature organization should be compact (efficient and comprehensive).

Depending on the speech representation, different attribute sets are meaningful, including the 14 Sound Pattern of English (SPE) binary attributes, phonological attributes such as nasality, frication, etc. The organization of the attributes can be parallel, hierarchical or combined. A parallel structure is a flat representation of all speech attributes, and it assumes that all the attributes are independent whereas in reality they are not. The flat representation avoids the problem of error propagation from different levels, but suffers from the problem of not utilizing layers of information about the sound so as to reduce the uncertainty as to sound class. A hierarchical structure is more efficient in representing all the sounds of a language (the set of all phones), but suffers from the problem of error propagation from higher levels to lower levels (i.e., errors made at a high level of the hierarchy propagate to lower levels with no clear correction mechanism). In this study we investigate both the parallel attribute organization and the hierarchical organization and compare their performance.

(4) *How to optimize attribute calculation from training*

Attribute events are usually obtained by some type of probability estimation process, e.g., Multi-Layer Perceptron (MLP) or Karhunen-Loeve (K-L) expansion.

(5) *Training set label correction*

The TIMIT phone labels are known to have errors in both labeling and alignment so that careful use of the TIMIT data is essential for reliable attribute estimation methods. When utilizing an attribute combination method to calculate the likelihoods of phonemes (clusters of attributes), the training set labels and alignments might have to be modified based on the confidence and time span of the resulting phonemes.

In order to estimate a range of speech attributes from the various speech parameters and measurements, we need both a way of estimating attribute probabilities as well as a training set of data that will enable us to optimize the estimation method and maximize the reliability of the resulting attribute probabilities. To that end we use the training set and the test set in the TIMIT database for both training the various classifiers of Figure 3.1, and to test the resulting estimation methods on unseen data. We do not investigate on segmentation in this thesis, and when we need pre-segmentation, we use the TIMIT hand labels as the segment points.

3.2 Frame-Based and Segment-Based Methods

Speech frames are flexible and convenient representations of the spectral/temporal properties of the speech at a given time, since the length of the window can be short or long when calculating speech parameters in a frame, and the total number of frames when calculating a speech feature or phoneme can also vary. They generally are easy to implement, and they characterize static, short-time, unchanging properties of speech. Segments normally cover longer time spans (order of 10 frames or longer) and they characterize speech dynamics. A segment generally contains a variable number of frames. The static sounds, including long vowels, fricatives and silence, can be determined via a single frame, whereas the dynamic sounds, including diphthongs, semivowels, nasals, stops, affricates and whisper, need segments to classify. In this work, we investigate both frame-based and segment-based methods in the detection and classification of distinctive speech features and phonemes, and at last combine the two methods to form a complete speech recognition system.

Chapter 4

Frame-Based Methods for Speech Attribute Detection

In this chapter, we estimate the likelihood of the Sound Pattern of English (SPE) features using frame-based methods and then we detect the phoneme classes using combinations of attributes. First, the mathematical framework of frame-based phoneme and attribute detection is presented. Then, we investigate the phoneme boundary effect on the accuracy of speech attribute detection. In Section 4.3, we use Artificial Neural Networks to detect the 14 Sound Pattern of English features. In Section 4.4, we integrate the SPE feature detection into the ASAT system. We found that single frames do not contain sufficient information to reliably detect dynamic phonemes (e.g., diphthongs, stops, affricates). In later chapters, we will integrate frame-based methods with segment-based methods, thereby giving a complete attribute-based frontend processing for detecting and classifying phonemes of the English language.

4.1 Mathematical Formulations

In frame-based phoneme recognition, given a speech parameter vector x of the input frame, the process can be viewed as trying to find the best phoneme $\hat{a}$ within the phoneme alphabet A using the *maximum a posteriori* (MAP) probability rule

$$\hat{a} = \arg\max_{a \in A} P(a \mid x) \tag{4-1}$$

where a is a phoneme hypothesis for a given input speech frame $x,$ and $\boldsymbol{A}$ is the alphabet consisting of M phonemes of the English language. A phoneme is represented by a set of N parallel binary speech features, i.e., $e_1,\ldots,e_N$ where

$$a \overset{def}{=} (e_1, e_2, \cdots, e_N), \quad e_j = 0,1, \quad j = 1, \cdots, N \tag{4-2}$$

If we assume that the features are independent of each other, $P(a|x)$ in equation (1) can be written as

$$P(a \mid x) = P(e_1, e_2, \cdots e_N \mid x) = \prod_{j=1}^{N} P(e_j \mid x) \tag{4-3}$$

Note that since there are less than 2^N phones in the alphabet, not all feature value combinations are possible, and some feature value combinations do not correspond to any phone and hence will be discarded in the optimization of equation (4-1).

The posterior probabilities $P(e_j \mid x), j = 1, \cdots, N$ denote the probability of detection of features $e_j, j = 1, \cdots, N$, and can be readily estimated using artificial neural networks.

4.2 The accuracy of phoneme boundaries

Due to the effects of phoneme co-articulation and TIMIT labeling errors, we need to determine how much the boundary frames (which are often grossly in error) affect phonetic feature detection accuracy. The issues with phoneme co-articulation and TIMIT labeling accuracy are the following:

- Co-articulation--When people speak syllables, words, or sentences, sound co-articulation occurs as a natural process of speaking. This means that the articulation for the current phoneme is highly influenced by the articulation of the preceding and succeeding phonemes, and similarly influences the articulation of the following phoneme, thus creating overlapping of phoneme articulation of sequences of phonemes. For this reason, the (hand labeled) phoneme borders (from the TIMIT database) are not exact, and the acoustic/phonetic/linguistic features for frames at or near the phoneme boundaries are imprecise at best and wrong at worst.

- TIMIT labeling accuracy--The TIMIT labels are known to have errors in alignment, and timestamps for each phoneme are not exact. Hence a frame labeled as the beginning of one phoneme can, in fact, belong to the previous phoneme, or a frame labeled as the ending of one phoneme can, in fact, belong to the succeeding phoneme. Because the phonetic features are associated with each phoneme, inexact phoneme boundaries can affect the accuracy of phonetic features.

4.2.1 Atal & Rabiner Algorithm

To estimate the accuracy of the TIMIT-labeled phoneme boundaries, we classified TIMIT frames into the short-time temporal categories of Voiced/Unvoiced/Silence (V/U/S), using the Atal and Rabiner statistical estimation algorithm [6]. The algorithm used the statistical distributions of five acoustic parameters to make this VUS decision, namely:

(1) normalized zero crossing rate;

(2) log energy (relative to 0 db peak);

$$E_s = 10 * \log_{10}\left(\varepsilon + \frac{1}{N}\sum_{n=1}^{N} s^2(n)\right) \tag{4-4}$$

(3) normalized autocorrelation coefficient at unit sample delay, C_1;

$$C_1 = \frac{\sum_{n=1}^{N} s(n)s(n-1)}{\sqrt{\left(\sum_{n=1}^{N} s^2(n)\right)\left(\sum_{n=0}^{N-1} s^2(n)\right)}} \tag{4-5}$$

(4) first predictor coefficient, α_1, of a 12-pole LPC analysis using the covariance method;

(5) normalized prediction error, E_p

$$E_p = E_s - 10 * \log_{10}\left(10^{-6} + \left|\sum_{k=1}^{p} \alpha_k \phi(0,k) + \phi(0,0)\right|\right) \tag{4-6}$$

$$\phi(i,k) = \frac{1}{N}\sum_{n=1}^{N} s(n-i)s(n-k) \tag{4-7}$$

The VUS detection procedure assumed joint Gaussian distributions for the five parameters, and a training set of TIMIT utterances was used to estimate means and covariance of the 5 acoustic parameters listed above. The decision rule was a Bayesian classification decision, namely:

$$p_i g_i(x) \ge p_j g_j(x), \text{ for all } i \ne j \tag{4-8}$$

choose class i that maximized the class aposteriori probability, $p_i g_i(x)$, where $p_i(x)$ was the *a priori* probability that x belongs to the i^{th} class, and $g_i(x)$ was the joint L-dimensional Gaussian density function (L=5) with mean vector m_i and covariance matrix W_i:

$$g_i(x) = (2\pi)^{-L/2}|W_i|^{-1/2}\exp\left[-\frac{1}{2}(x-m_i)^t W_i^{-1}(x-m_i)\right] \tag{4-9}$$

4.2.2 Experiments

We use the TIMIT database for training and testing. The training set was the TIMIT training set consisting of 4620 speech files, and the test set was the separate TIMIT test set, consisting of 1680 speech files. We used a Hamming window with frame length of 32 ms and with a 10 ms frame overlap.

We found that by restricting the training and testing sets to TIMIT frames within a subset of the phonemes (we call this the ***stable phoneme set***, as shown in Table 4.1), and by avoiding phone boundary frames (which often are impossible to be accurately classified) we achieved classification accuracies of 99% for voiced frames, 87% for unvoiced frames and 96% for silence/background frames on the independent test set. When all phonemes were included in the training and test sets (still omitting the phone boundary frames), the classification accuracy fell to 96% for voiced frames, 72% for unvoiced frames and 93% for silence/background frames. Finally when all phonemes and all frames were used for training and testing, the classification error fell further to 93% for voiced frames, 60% for unvoiced frames, and 86% for silence/background frames. Results of these experiments are shown in Table 4.2-Table 4.4 below. The results showed that when the phoneme boundaries were included in the feature calculation, the overall performance degraded to some extent (from 96% to 84%). Clearly we have to tread lightly when training Neural Network or Bayesian classifiers using the TIMIT training set.

Another way of measuring the accuracy of phoneme boundaries is to measure how many boundaries occurred within a certain time period. According to Dusan and Rabiner's work in [15], about 97% of the phoneme boundaries occurred within 40 ms of hand determined range, and the accuracy of annotated boundaries is about plus or minus 13 ms on average.

Table 4.1 Stable phoneme set

Voiced	l, r, w, y, el, iy, ih, eh, ey, ae, aa, aw, ay, ah, ao, oy, ow, uh, uw, ux, er, ax, ix, axr, ax-h
Unvoiced	hh, hv, ch, s, sh, f, th
Silence	pau, epi, h#

Table 4.2 Results on V/U/S detection for stable phonemes and without boundary frames (in number of frames and in percentage)

	Silence	**Unvoiced**	**Voiced**	**Overall**
Silence	56194 95.84%	1504 2.57%	936 1.60 %	96.26%
Unvoiced	3319 7.49%	38468 86.86%	2502 5.65%	
Voiced	684 0. 43%	876 0. 55%	158298 99.02%	

Table 4.3 Results on V/U/S detection for all phonemes, without boundary frames (in number of frames and in percentage)

	Silence	**Unvoiced**	**Voiced**	**Overall**
Silence	76787 92.61%	2348 2.83%	3784 4.56%	90.63%
Unvoiced	4209 6.78%	44798 72.18%	13058 21.04%	
Voiced	6522 3.64%	470 0. 26%	172253 96.10%	

Table 4.4 Results on V/U/S detection for all phonemes, all frames (in number of frames and in percentage)

	Silence	**Unvoiced**	**Voiced**	**Overall**
Silence	93602 86.32%	5720 5.27%	9117 8.41%	84.35%
Unvoiced	13386 12.01%	67115 60.22%	30945 27.77%	
Voiced	18317 6.26%	2738 0. 94%	271596 92.81%	

4.3 Use of Multi-Layer Perceptrons to Detect SPE Features

We trained 14 frame-based multi-layer perceptrons (MLPs) in parallel to detect each of the 14 binary valued sound pattern of English (SPE) speech features. Each MLP estimates one *a posteriori* probability $P(e_j \mid x), j = 1, \cdots, N$ as in equation (4-3), where *N*=14.

4.3.1 Use 2-layer and 3-layer MLPs to detect SPE features

Using the 61 phoneme set, we built and optimized a set of Multi-Layer Perceptrons (MLP), one for each of the 14 Sound Patterns of English features, using the Netlab [48] and Matlab toolboxes. The parallel organization of the 14 SPE features for the 61 TIMIT phonemes is depicted in Appendix A Sound Pattern of English Feature Values for the TIMIT 61-Phoneme Alphabet. Each feature is represented as +1 for feature present, 0 for feature absent, −1 for feature unavailable.

We tested this architecture using Multi-Layer Perceptrons with fully connected 2 hidden layers and 3 hidden layers. A separate frame-based MLP was trained for detection of each of the 14 SPE features. The input layer had 13 nodes corresponding to 13 MFCCs, the output layer contained one node corresponding to one of the 14 SPE features. Figure 4.1 and Figure 4.2 show a 2-layer MLP and a 3-layer MLP.

The problems with utilizing the MLP architecture for detecting the SPE features are:

(1) the convergence of the MLP is not guaranteed;

(2) the training of the MLP is slow;

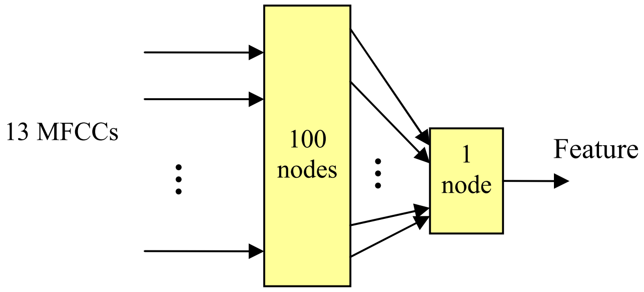


Figure 4.1 Two-Layer MLP

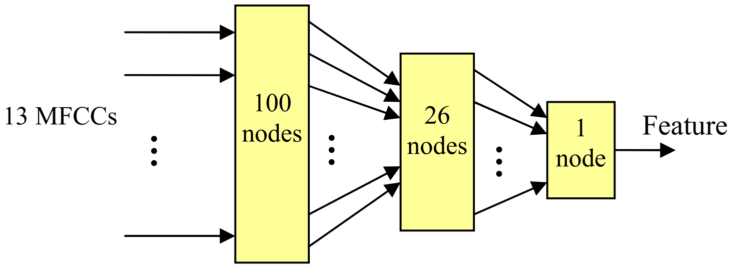


Figure 4.2 Three-Layer MLP

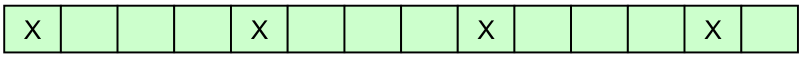


Figure 4.3 1-in-4 sampling of frames

(3) information contained in consecutive frames is highly correlated.

Based on the above problems, we determined that sampling frames of the input (i.e., not using all training set frames) was a good way to reduce computation with little or no loss in performance, as shown in Figure 4.3. In the following experiments, we sampled 1 out of 4 consecutive frames to reduce computation by a factor of 4.

4.3.1.1 Two-Layer MLP results

We first optimized the number of nodes in the hidden layer. Using nasal detection as the test attribute, we varied the size of the hidden layer and re-trained the

MLP for each size hidden layer. We found that about 100 nodes in the first hidden layer was adequate.

Next we tried to determine how many files were needed for training a particular speech attribute. For example, to detect nasals, we found that the classifier mean-squared error was relatively insensitive over a range of 100-4000 training files, as shown in Figure 4.4.

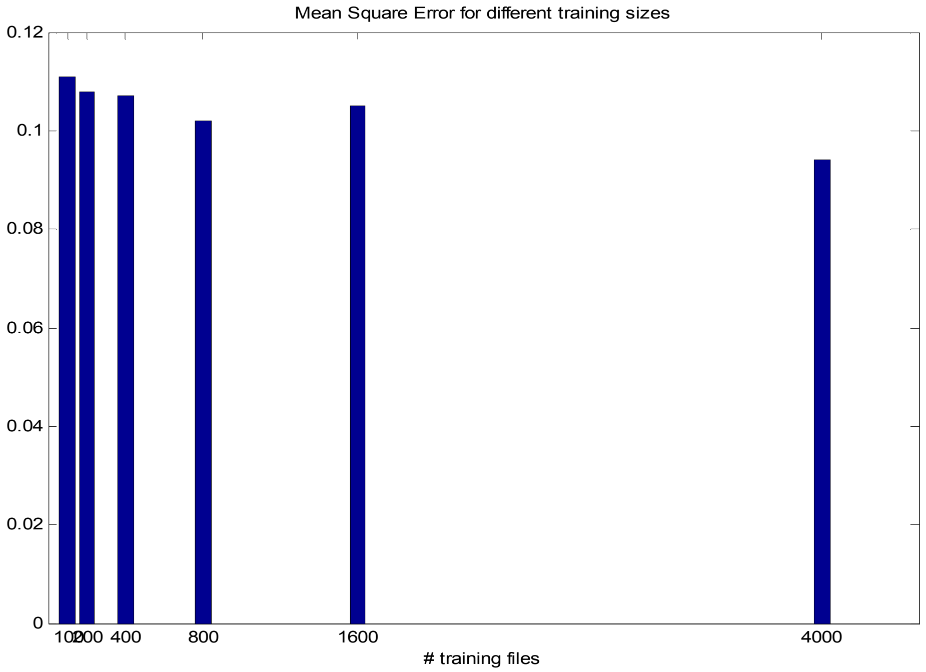


Figure 4.4 Mean square error vs. number of training files for 2-layer MLP

An example of nasal classification on an independent test utterance, using the ANN Nasal Classifier that resulted by optimizing the 2-layer MLP, is shown in Figure 4.5. Finally, Figure 4.6 shows the ROC curve for nasal detection using the 2-layer MLP classifier. The MLP output range is in the interval [0, 1]. If the output is above a threshold, the feature is classified as present; otherwise it is classified as absent. Using a threshold of about 0.05 (on the MLP output) keeps both the number of false rejections and false alarms at very low levels.

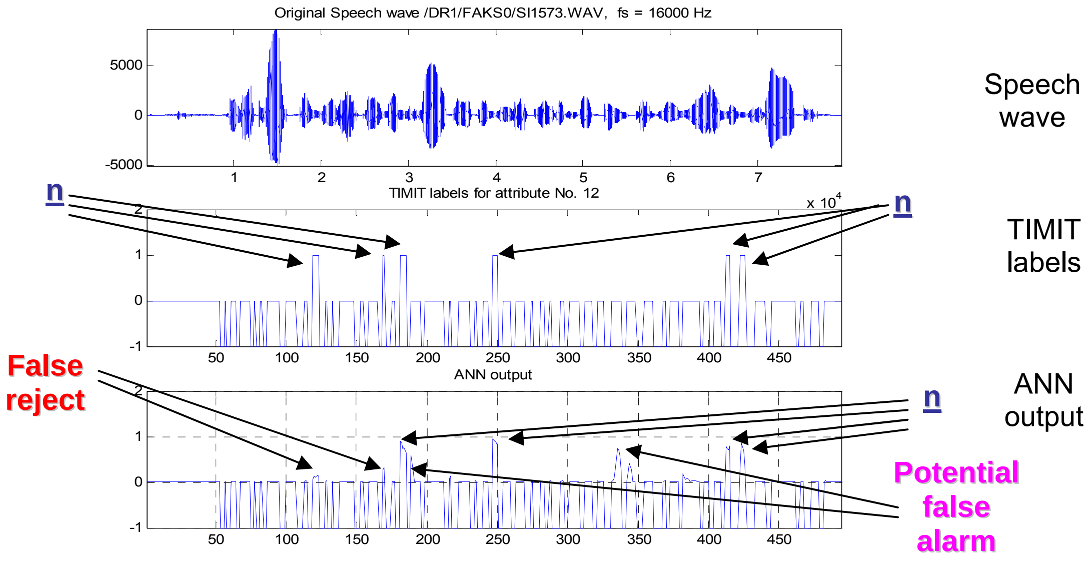


Figure 4.5 Two-layer MLP output including false alarms and false rejects. TIMIT sentence: “His captain was thin and haggard and his beautiful boots were worn and shabby”. 1 ~ nasal; 0 ~ non-nasal; -1 ~ boundary frames, ignored.

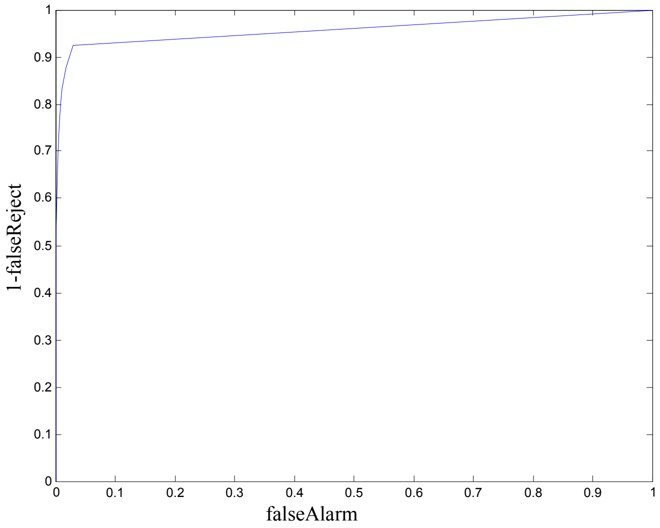


Figure 4.6 ROC curve for MLP output threshold. Y-axis: 1-falseReject, X-axis: falseAlarm. The area under the curve is 0.9571.

4.3.1.2 Three-Layer MLP results

The 2-layer MLP developed using the Netlab toolbox performed well but there was a persistent convergence problem. The MLP training sometimes didn't converge. Hence we next used the Matlab Neural Network toolbox and developed and trained a 3-layer MLP. Due to computer memory limits, we again had to sample the frames in the training set. Due to the high correlation between adjacent frames, we choose 1 out of every four consecutive frames for training and testing. Thus, the training set size was 48,000 frames and the test set size was 33,020 frames. The phoneme boundary frames and the immediately adjacent frames were also discarded for reasons explained previously.

For the 3-layer ANN we found that having 100 nodes for the first hidden layer, with 26 nodes for the second hidden layer gave the best classification accuracy for the 14 SPE features. Figure 4.7 shows an example of selecting the best MLP topology.

We classified the attribute detection performance as "good" when the detection accuracy was above 90% for ***both*** + feature and – feature detection; we classified it as "acceptable" when both + and – feature detection rates were above 80% but at least one was below 90 %; and we classified it as "poor" when at least one of the feature detection rates was below 80%.

By using a random sample of both the features (+ classification correct) and the anti-features (- classification correct), we obtained the following results on the 14 SPE attribute detectors:

- 4 detectors provided good performance, namely the features of Voice (10), Continuant (11), Strident (13), and Silence (14).

- 2 detectors provided acceptable performance, namely the features of Vocalic (1), and Tense (9).
- All other 8 feature detectors ((2)Consonantal, (3) High, (4) Back, (5) Low, (6) Anterior, (7) Coronal, (8) Round and (12) Nasal) gave poor performance for the + features (range of from 43% for the +ROUND feature to 75% for the +LOW feature), but good for the – features (all between 94% and 99% correct)

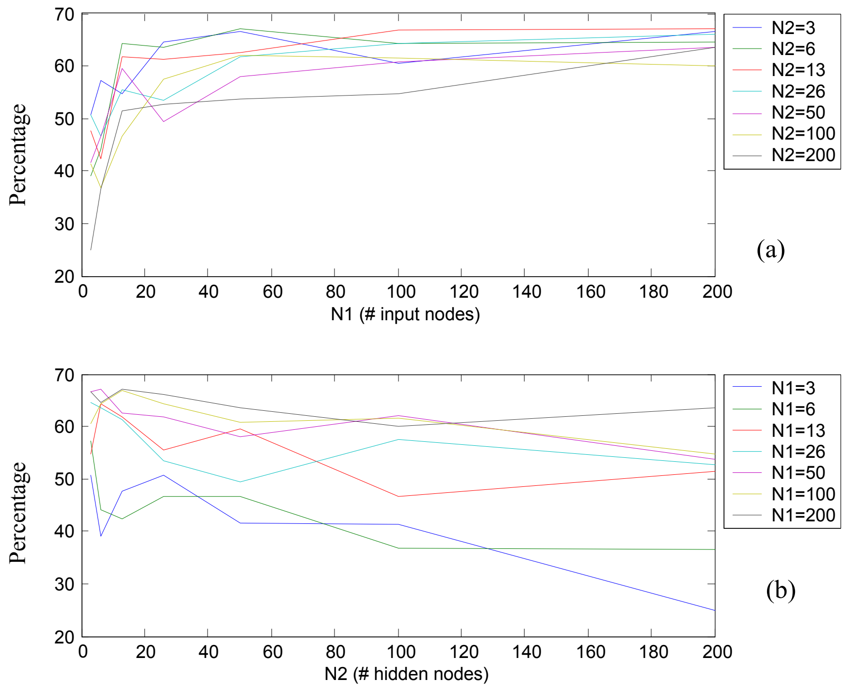


Figure 4.7 Nasal detected as nasal. (a) for fixed N2, vary N1; (b) for fixed N1, vary N2.

4.3.2 Balancing the training set

Initially we trained the 3-layer MLP classifiers using randomly selected frames for each feature where there generally were far more occurrences of frames with the "–

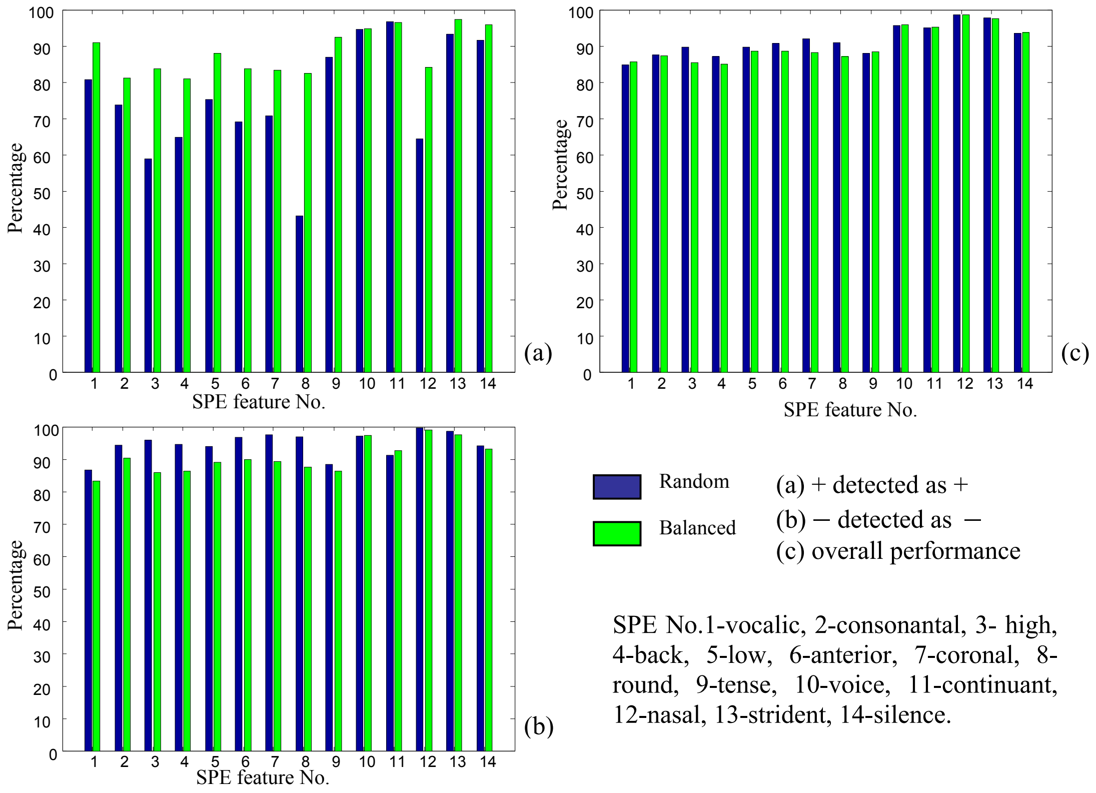
feature" present than frames with the "+ feature" present. We called this training set the "unbalanced" set. For example, the ratio between frames with + nasal features and frames with – nasal features in the training set was about 3:100—i.e., there were 33 times more frames with the negative feature than with the feature that was being trained.

On average there are 4 "+" features for a TIMIT phoneme. Since we are mostly interested in detecting the + features accurately and reliably, we devised a way to carefully balance the training set so that the number of training samples with the + features was comparable to the number of training samples with the – feature. By carefully balancing the training set of features against the set of anti-features (rather than using the natural imbalance that occurs in the TIMIT set), we were able to significantly improve the correct feature classification scores without seriously lowering the correct anti-feature rejection capabilities of the 3-layer ANN. Based on a balanced training set, the + feature detection performance significantly improved without seriously affecting the – feature detection accuracy. The new set of results (using the balanced training set) showed that the MLP feature detectors for 6 SPE features achieved "good" detection performance (as compared to 4 for the unbalanced training set), and the remaining 8 SPE feature detectors achieved "acceptable" performance (as compared to 2 for the unbalanced training set). For all the 14 SPE features (taken as a whole) the average frame correctness for + features detected correctly as + features was 90%; similarly the overall rate of – features correctly detected as – features was 90.5%; and the overall rate of frames being correctly classified was 90.4% (as compared to 81.9%, 95.1% and 91.5% respectively for the unbalanced training set). Table 4.5 and Figure 4.8 show the comparison of detection performance on unbalanced and balanced training sets. King *et* *al.* ([34], [35]) achieved similar results for the SPE feature detection, but they did not consider the importance of balancing the "+" and "–" features.

Although there are many ways to explain the power of training with equal representations of the feature being detected with the absence of that feature, perhaps the best (and most analytical) way of showing why this method of training was optimal is via measurement of the ROC curve relating false rejections (classifying frames with a given feature as lacking that feature) to false alarms (classifying frames without a given feature

Table 4.5 Compare results using unbalanced training sets and balanced training sets

	# good performances	# acceptable performance	# poor performance
Unbalanced	4	2	8
Balanced	6	8	0



SPE No.1-vocalic, 2-consonantal, 3- high, 4-back, 5-low, 6-anterior, 7-coronal, 8-round, 9-tense, 10-voice, 11-continuant, 12-nasal, 13-strident, 14-silence.

Figure 4.8 Comparison of random training and balanced training for 14 SPE features

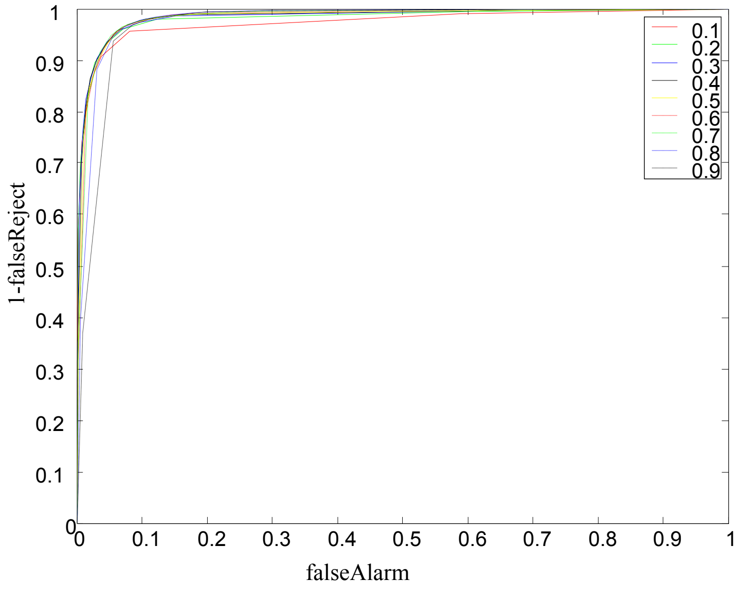
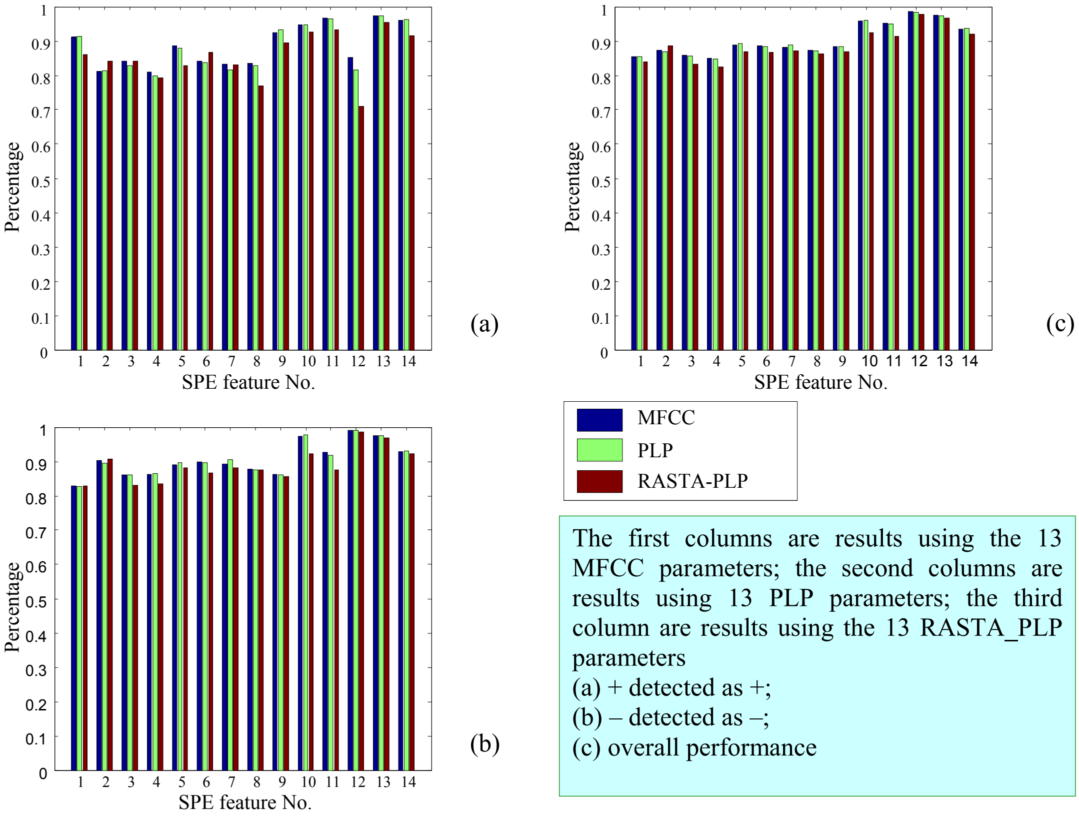


Figure 4.9 ROC curve for different training data balance ratios for SPE feature "continuant"

as having that feature). Thus we created a complete set of ROC curves for the 14 SPE features and Figure 4.9 shows an example of a typical ROC curve for SPE feature No.11: "continuant" for a variety of training data 'balance ratios' ranging from 0.1 (heavily biased towards examples lacking the selected feature) to 0.9 (heavily biased towards examples having the selected feature). Although the performance of the systems with almost any training ratio was reasonably good, the optimum performance (corresponding to the maximum area under the ROC curve) for all of the SPE feature detectors occurred when using balanced training data (i.e., an equal number of training examples with (+) and without (–) the desired feature).

4.3.3 Comparison of MFCC, PLP and RASTA-PLP Features

We also compared MFCC, PLP and RASTA-PLP speech parameters for use in detection of the 14 SPE features, as shown in Figure 4.10, and found that MFCC coefficients gave the highest classification accuracies for 9 of the 14 SPE features, while PLP parameters gave the highest classification accuracy for 3 features, and finally RASTA-PLP gave the highest classification accuracy for the remaining 2 features.



The first columns are results using the 13 MFCC parameters; the second columns are results using 13 PLP parameters; the third column are results using the 13 RASTA_PLP parameters
(a) + detected as +;
(b) – detected as –;
(c) overall performance

Figure 4.10 Comparing performances of MFCC, PLP and RASTA-PLP

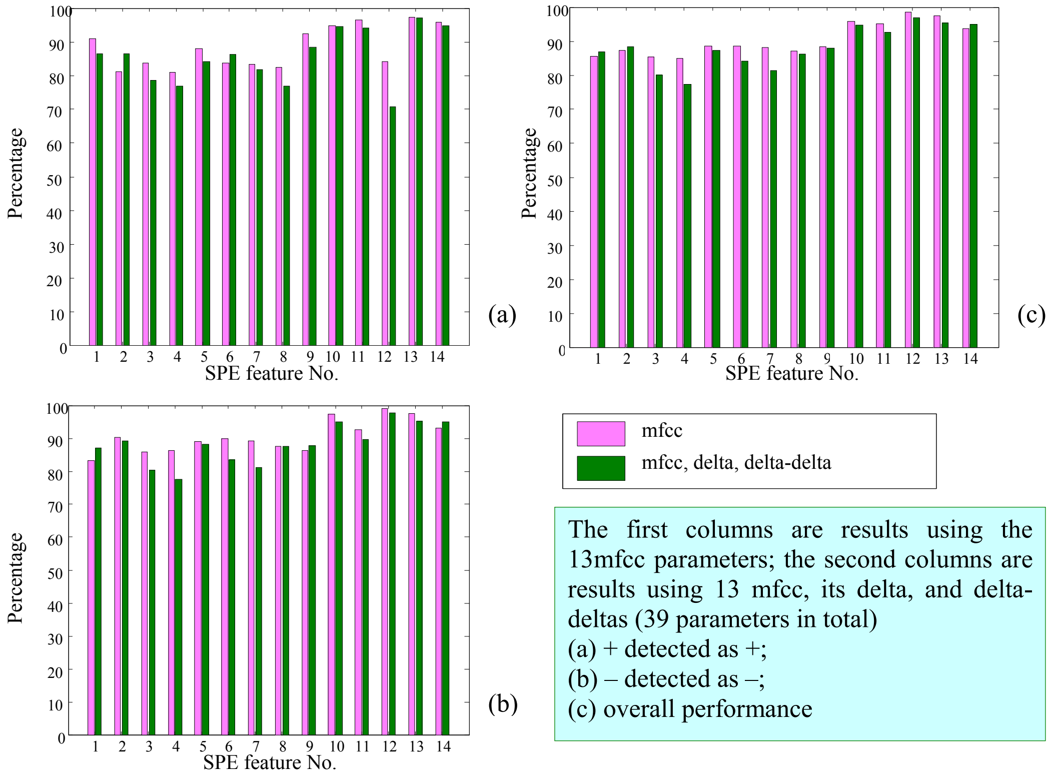
4.3.4 Comparison of MFCC and MFCC and its delta, and delta-delta derived features

We compared performance of the MLP detectors of the 14 SPE features using the following parameter sets:

- 13 mfcc coefficients per speech frame
- 13 mfcc, 13 delta mfcc, and 13 delta-delta mfcc coefficients for a total of 39 input coefficients per speech frame

For MFCC features alone, 48000 frames were used for training and 33020 frames for testing. When using mfcc, delta, and delta deltas, due to memory limits, we used 24000 frames for training, and 33020 frames for testing.

Figure 4.11 compares detection accuracy for the 14 SPE features using either the set of 13 mfcc coefficients, or the set of 39 mfcc plus delta mfcc plus delta-delta mfcc coefficients, for correct acceptance (+ feature detected as + feature), correct rejection (- feature detected as – feature), and for overall performance. From the results shown in Figure 4.11, we found that only 3 of the 14 SPE feature detectors showed slightly better detection accuracy when using the 39 input parameters than when using only the initial set of 13 mfcc coefficents, while for the remaining 11 SPE features, the results were somewhat lower in accuracy. Because frame-based detectors mainly capture static information about a sound, performance of a feature detector using only the set of 13 mfcc coefficients was generally better than that obtained by combining the static features with dynamic information – i.e., the delta and delta-delta coefficients did not provide any reliable information for recognizing static sounds. The place where use of dynamic information about the sound should help is for reliable frame-based detection of dynamic sounds (e.g., diphthongs, semivowels) and also when using segment-based methods.



The first columns are results using the 13mfcc parameters; the second columns are results using 13 mfcc, its delta, and delta-deltas (39 parameters in total)
(a) + detected as +;
(b) – detected as –;
(c) overall performance

Figure 4.11 Comparing mfcc and mfcc plus deltas and delta-deltas

4.3.5 Error Pattern Analysis

In order to understand the root cause of detection failures of the various SPE features, we built an error pattern analysis tool and used it to diagnose the MLP detection outputs for the 14 SPE features to find which phoneme or phoneme combinations errors were most likely to have occurred. Using this error pattern analysis tool, we uncovered the major modes of detection failure for the various sounds of English. For example, for detection of the “consonantal” feature, as shown in Figure 4.12, we have found that high false alarms occured for the sounds /hh/ (as in the word “hay” /hh ey/) and /hv/ (as in the word “ahead” /ax hvs eh dcl d/) and the softly pronounced vowel /ax-h/ (as in the word “suspect” /s ax-h s pcl p eh kcl k tcl t/).

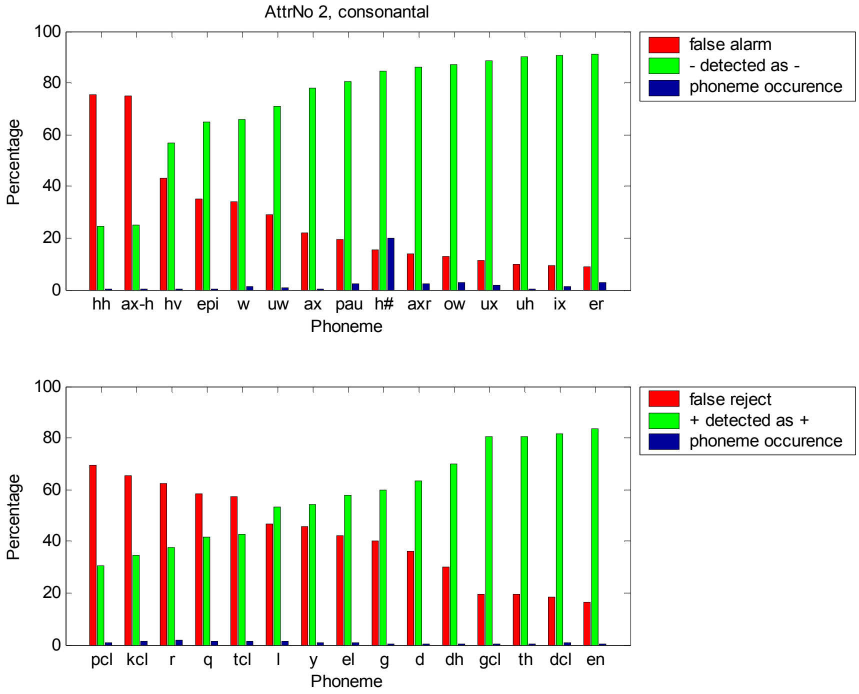


Figure 4.12 “Consonantal” feature detection errors. The plots show the top 15 errors leading to phoneme classification errors based on the presence or absence of the consonantal feature.

From an analysis of the error patterns from all SPE features and for all sounds of English, we found that the major failure modes were the following:

I. The features for diphthongs and stops were difficult to detect based on single frame measurements, since these sounds are inherently dynamic and thus need temporal sequence (segment) information.

II. The semivowels were prone to detection errors because of the strong influence of adjacent sounds on the spectral properties of the sound; it was anticipated that dynamic (segment) models would alleviate these problems.

III. Some (weak) fricatives were not easy to detect using frames, especially /th/.

IV. The nasalized vowels /em/, /en/, almost always performed worse than the nasal consonants /m/ and /n/. This was again due to the influence of the nasal on the vowel quality.

V. The stop gaps in speech should be classified as silence; and the stop gaps are important features for reliable recognition of stop consonants.

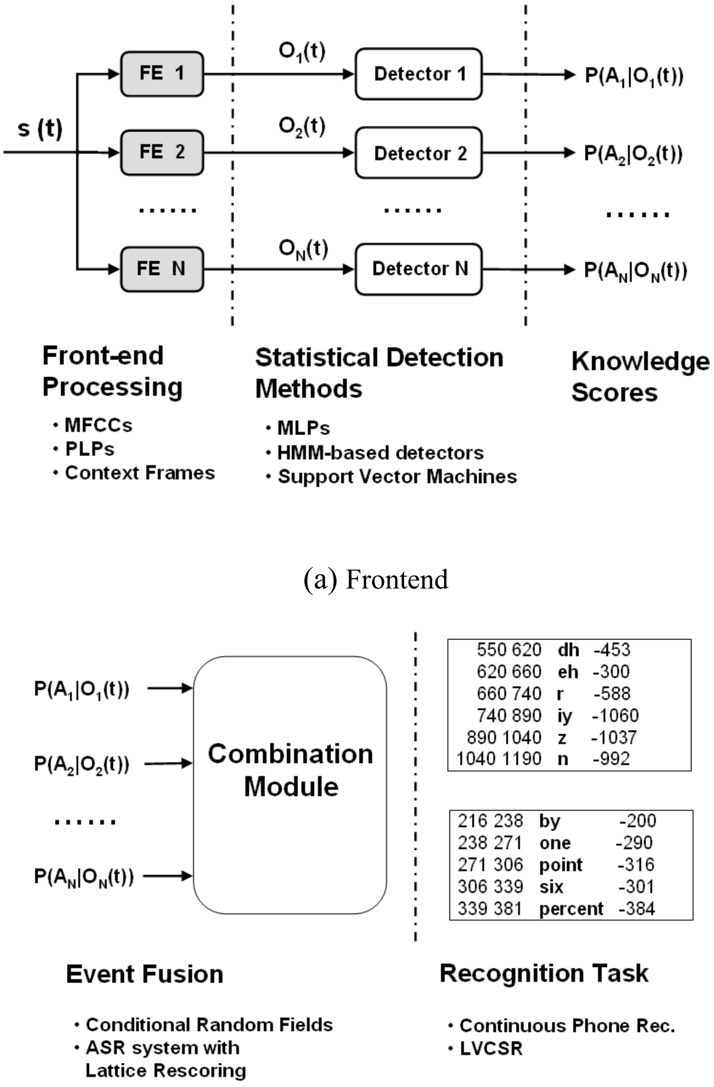
VI. Features of some vowels were incorrectly detected, and it was indicative that the MFCC parameters were not appropriate for detection of those features, or the MLP was not a good classifier for detection of those features, or perhaps most importantly, the SPE features were not good measures for detection of these vowels.

VII. The TIMIT labels have errors in alignment (e.g., stop gaps were classified as voiced).

VIII. Some phonemes in the 61 phoneme set were not properly segmented and marked, e.g., the softly pronounced /ax-h/.

4.4 Integrate the SPE feature detection in the ASAT system

The Automatic Speech Attribute Detection system aims at using basic acoustic/linguistic units in a probabilistic event detection model to determine likelihoods of phonemes, words, and sentences. There are two basic modules in the current system, as shown in Figure 4.13 [8]: (1) the front-end processing module computes the frame-wise likelihood scores for several predefined sets of speech feature classes (speech attributes). Our SPE feature set is one of the speech attribute classes being used in the ASAT system. (2) The decoding module combines the different sets of attribute detection scores with results from HMM based decoders using Conditional Random Fields (CRF) [47] and knowledge-based lattice rescoring of phoneme lattices for continuous phoneme recognition on the TIMIT database.



(a) Frontend

550	620	**dh**	-453
620	660	**eh**	-300
660	740	**r**	-588
740	890	**iy**	-1060
890	1040	**z**	-1037
1040	1190	**n**	-992

216	238	**by**	-200
238	271	**one**	-290
271	306	**point**	-316
306	339	**six**	-301
339	381	**percent**	-384

(b) Decoding

Figure 4.13 ASAT Detection based ASR [8]

The speech attribute detectors used are the following, as shown in Table 4.6:

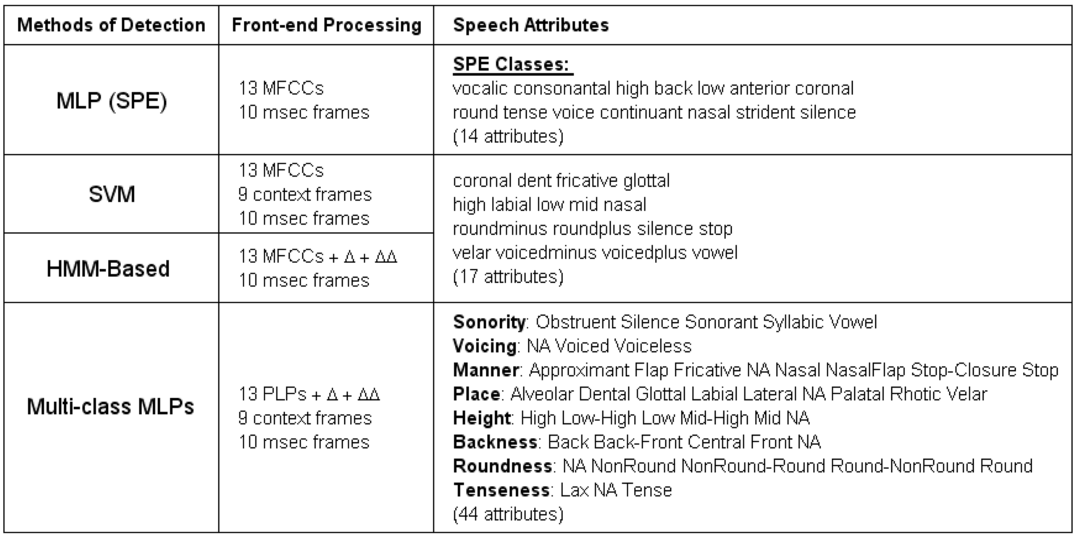
(1) MLP detectors for the SPE features (the second row in Table 4.6);

(2) Multiclass MLPs for Intl. Phonetic Assoc. (IPA) classes (the last row in Table 4.6);

(3) HMM based detectors for 17 phonetic attribute classes (the fourth row in Table 4.6);

(4) Support Vector Machine based detectors for the same 17 phonetic attribute classes in (3) (the third row in Table 4.6).

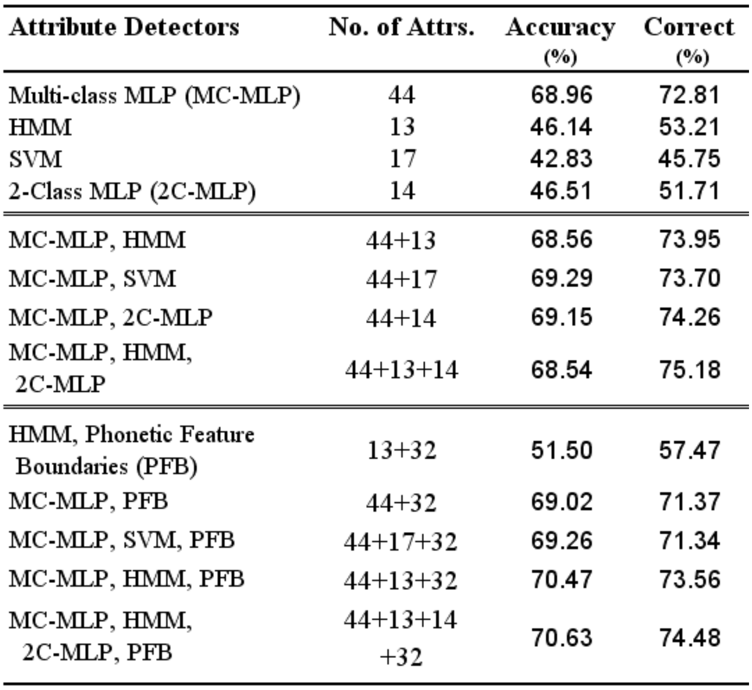
Table 4.6 Summary of detectors, front-end processing methods and speech attributes [8]



Methods of Detection	Front-end Processing	Speech Attributes
MLP (SPE)	13 MFCCs 10 msec frames	**SPE Classes:** vocalic consonantal high back low anterior coronal round tense voice continuant nasal strident silence (14 attributes)
SVM	13 MFCCs 9 context frames 10 msec frames	coronal dent fricative glottal high labial low mid nasal roundminus roundplus silence stop velar voicedminus voicedplus vowel (17 attributes)
HMM-Based	13 MFCCs + Δ + ΔΔ 10 msec frames	
Multi-class MLPs	13 PLPs + Δ + ΔΔ 9 context frames 10 msec frames	**Sonority**: Obstruent Silence Sonorant Syllabic Vowel **Voicing**: NA Voiced Voiceless **Manner**: Approximant Flap Fricative NA Nasal NasalFlap Stop-Closure Stop **Place**: Alveolar Dental Glottal Labial Lateral NA Palatal Rhotic Velar **Height**: High Low-High Low Mid-High Mid NA **Backness**: Back Back-Front Central Front NA **Roundness**: NA NonRound NonRound-Round Round-NonRound Round **Tenseness**: Lax NA Tense (44 attributes)

We also utilized information in the phone and phonological feature boundaries, as the rapid changing character of boundaries often carry a lot of important information for speech recognition. Using the above sets of speech attributes and the Conditional Random Fields, and considering the phonetic feature boundary (PFB) detection, the phoneme detection results are showed in Table 4.7. In this table, the first column lists the different attribute detectors, the second column lists the number of attributes used in the detector or combination of detectors. From this table, we can see that, broadly speaking, the phoneme recognition accuracy improved as the number of attributes increased in each speech attribute detectors (or combination of detectors). When using the largest number of different sets of speech attributes (44+13+14+32=103 attributes in total), we achieved the highest phoneme recognition accuracy of 70.63%. The results of our first set of experiments are very encouraging, and we hope to improve the phoneme recognition performance even further as we add more speech attributes to the ASAT system.

Table 4.7 Continuous phone recognition experiments with CRFs on the TIMIT database [8]



Attribute Detectors	No. of Attrs.	Accuracy (%)	Correct (%)
Multi-class MLP (MC-MLP)	44	68.96	72.81
HMM	13	46.14	53.21
SVM	17	42.83	45.75
2-Class MLP (2C-MLP)	14	46.51	51.71
MC-MLP, HMM	44+13	68.56	73.95
MC-MLP, SVM	44+17	69.29	73.70
MC-MLP, 2C-MLP	44+14	69.15	74.26
MC-MLP, HMM, 2C-MLP	44+13+14	68.54	75.18
HMM, Phonetic Feature Boundaries (PFB)	13+32	51.50	57.47
MC-MLP, PFB	44+32	69.02	71.37
MC-MLP, SVM, PFB	44+17+32	69.26	71.34
MC-MLP, HMM, PFB	44+13+32	70.47	73.56
MC-MLP, HMM, 2C-MLP, PFB	44+13+14 +32	70.63	74.48

4.5 Summary

In this chapter, we measured a range of spectral and temporal, short term and long term parameters, and included them in the ASAT parameter set. We extensively tested ANN's of different types and provided linguistic/distinctive feature labels with varying degrees of success. Training on balanced training sets showed significant improvements over standard ANN training methods which use randomly selected training data. Due to the TIMIT labeling errors, boundary frames were discarded for training purposes. We also found that different auditory models were of benefit for different speech features. When various sets of speech attribute detector results are combined together in the ASAT system, the overall performance increased as more features were added into the system.

In the next chapter, we investigate the use of segment-based methods for classification of dynamic sounds.

Chapter 5

Segment-Based Approaches to Sound Classification

A segment covers a much longer time span (usually more than 100 ms) than a single frame (10~30 ms) and carries information that is necessary for dynamic phoneme detection. A static sound's feature values nominally don't change much during the period of articulation, whereas a dynamic sound's feature values change dramatically over the duration of the sound. It is the changing nature of the feature values that characterize a dynamic sound. In this chapter, we investigate the use of segment-based methods for the classification of stop consonants, a class of sounds that are typical of dynamic sounds. We examine the use of segment-based detection methods for all phoneme classes later in this thesis research.

5.1 Hierarchical phoneme detection structure

In order to investigate the use of segment-based approaches to the classification of phonemes, we need to create some type of organizational structure for characterizing the various sound classes. A flat structure, where all sounds are equally likely, has some advantages, but suffers from the problem of not utilizing layers of information about the sound that reduce the uncertainty as to sound class. A hierarchical structure is more efficient in representing all the sounds of the language (the set of spoken phonemes), but suffers from the problem of error propagation from higher levels to lower levels. In the case of a hierarchical structure, we must have extremely accurate classification for the higher levels so that the detection algorithms can provide good performance at the lower levels.

5.1.1 Drawback of the 14 SPE features for detection of the 61 TIMIT phonemes

In the previous chapter, MLPs were used to detect the 14 Sound Pattern of English features for the 61 phoneme TIMIT alphabet using single frames of MFCC's coefficients. The 14 SPE features were detected in parallel, and each feature was represented as +1 for present, 0 for absent, −1 for unavailable. Based on this flat representation of individual speech frames, we were able to measure likelihood of a class of 61 sounds of the language (the so-called TIMIT phonemes). The drawbacks of this structure were that:

(1) It assumed all the features of each sound were independent, whereas in reality they are not.

(2) The 61 phonemes did not have unique feature values, e.g., /bcl/ and /pcl/ shared the same feature values, and /aw/ and /ao/ also shared the same feature values.

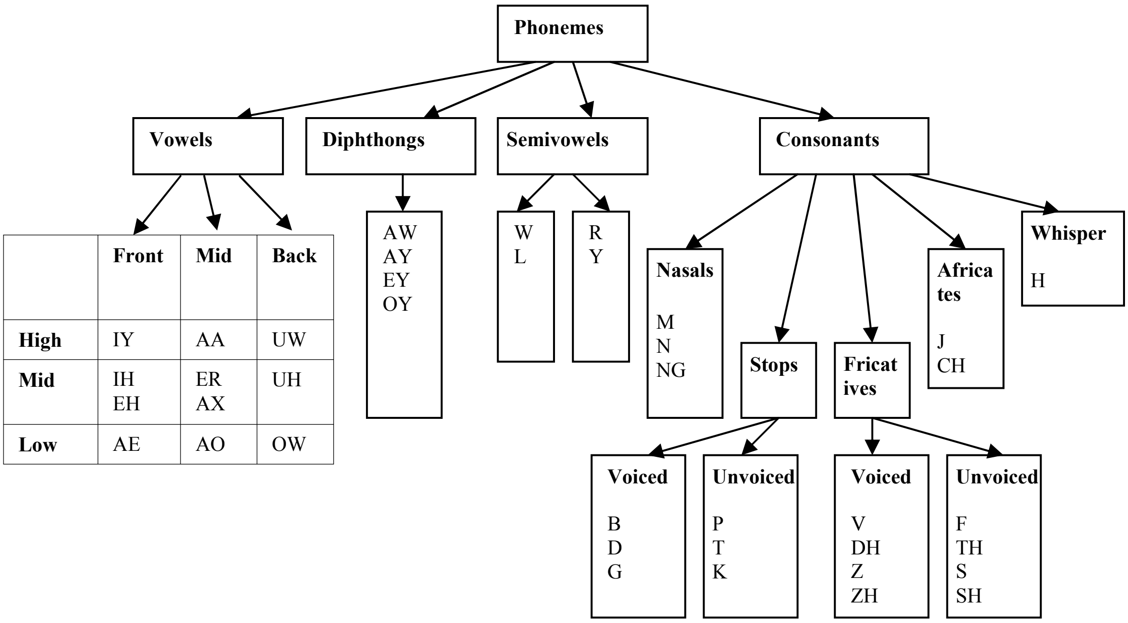
(3) Some diphthongs could only be distinguished from certain vowels by setting the vocalic feature to 0, and the ANN outputs for those phonemes showed high confusion rates.

(4) Diphthongs, stops and affricates are dynamic phonemes, and they needed dynamic features to detect them reliably.

(5) Phonemes were difficult to detect reliably from single frame measures, especially dynamic phonemes. Segment-based methods were needed.

5.1.2 Hierarchical phoneme organization using 40 phonemes

Instead of using the 61 TIMIT phonemes as the base set of sounds of the English language, we decided to use the reduced set of 39 phonemes plus silence (40 phonemes all together) that was widely used in DARPA speech recognition and natural language understanding tasks. We also decided to represent the set of 40 sounds using a hierarchical structure of the type shown in Figure 5.1.



	Front	**Mid**	**Back**
High	IY	AA	UW
Mid	IH EH	ER AX	UH
Low	AE	AO	OW

Figure 5.1 Hierarchical phoneme organization structure

5.2 Segment-Based Methods for Phoneme Classification

5.2.1 The Variability of Speech Sounds

Phonemes in the English language can be partitioned into several broad classes according to various linguistic or acoustic criteria, e.g., vowels, unvoiced fricatives, nasals. Individual phonemes can further be classified as either static or dynamic sounds. Static sounds are those whose temporal/spectral properties are relatively steady during the central part of phoneme articulation, e.g., long vowels and fricatives. A dynamic phoneme's feature values change significantly over the duration of the sound, either as an essential part of the phoneme itself (e.g., diphthongs, affricates) or as a result of context from the previous and/or succeeding sounds (e.g., semivowels, stops).

The basic implication for dynamic sounds is that their inherent characteristics (which enable humans to recognize the sounds) are time-varying and highly context sensitive. As such, there arises the need for dynamic features (i.e., features measured over segment-length sections of speech) for characterizing the time-varying properties of the sound. Such segments normally cover relatively long time spans (order of 10-15 frames or longer) and attempt to characterize speech dynamics (both inherent and context-dependent).

The above analysis is a bit simplistic since multiple occurrences of a common sound (especially highly dynamic sounds) are not all of equal duration, or even all with the same set of time-varying features. Thus, for example, the stop gap presence and duration of a voiced stop sound (followed by a vowel) are highly variable and highly dependent on speaking rate, amount of sound co-articulation, etc. Due to this type of pronunciation variability, the same phoneme (in the same context) can be pronounced differently and be of varying duration. In order for segment-based classification methods to cover the full range of phoneme pronunciations, segments must contain a variable number of frames. However, for ease of processing, it is best if segments contain a fixed number of frames. There exist various techniques to time warp and align variable length segments with a fixed length prototype, including the use of Dynamic Time Warping [54], and Viterbi alignment algorithms as illustrated in Figure 5.2, or even use of a Hidden Markov Model. The process, illustrated in Figure 5.2 shows, for each of a set of 40 phonemes, the various tokens of variable length being aligned with the ‘average’ token of length N1 frames via a simple process of clustering (to create the prototype token) and time aligning to the cluster center. To give some idea as to the degree of variability of the duration of each of the 40 phonemes of English, Figure 5.3 shows the average duration of the set of TIMIT phonemes, calculated from the TIMIT training set transcriptions.

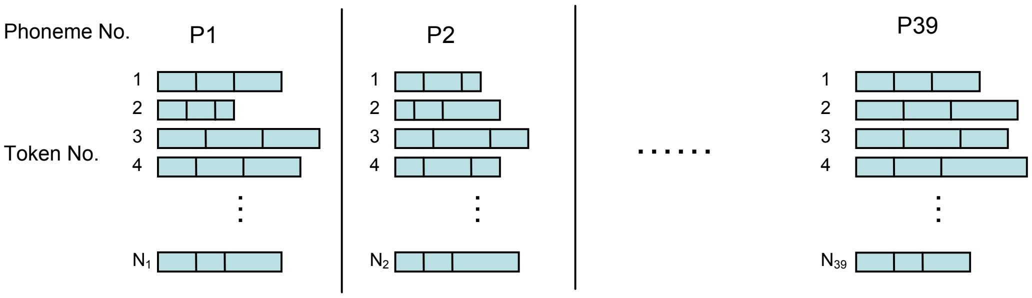


Figure 5.2 Use of DTW alignment of variable length segments with a fixed length prototype

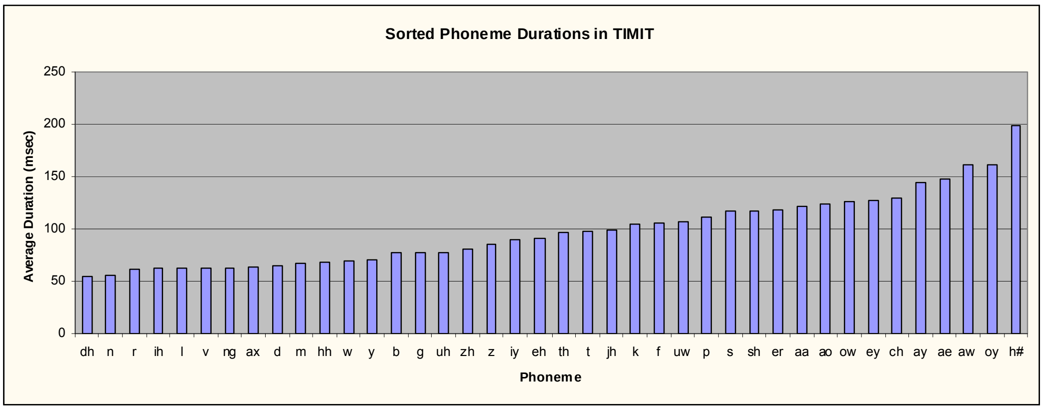


Figure 5.3 Phoneme durations in TIMIT

5.2.2 Mathematical Formulation

For hierarchical speech feature classification, given an input speech segment $\underline{y}$ (represented by a sequence of speech parameter vectors), the goal is to find the class $\hat{c}$ within the range of H classes with the maximum *a posteriori* probability

$$\hat{c} = \arg\max_{c \in C} P(c \mid \underline{y}) \qquad (5\text{-}1)$$

where $C = \{c_k, k = 1, \cdots, H\}$ is the set of classes that the current segment is to be classified within, $P(c \mid \underline{y})$ is the posterior probability of segment $\underline{y}$ belonging to class *c*, where we have used the result that for all *C* classes $\sum_{c \in C} P(c \mid \underline{y}) = 1$.

In this chapter, we used the Time-Delay Neural Networks to estimate the *a posteriori* probability $P(c \mid \underline{y})$.

5.3 Time-Delay Neural Networks

The use of a special class of neural networks, called Time-Delay Neural Networks (TDNN), as originally proposed by Waibel et al. in 1989, has been shown to be a good method for classification of dynamic sounds [70]. The TDNN in [70] detected stop burst events occurring within 30 ms sliding windows in a fixed 150 ms segment, and classified the segment based on the match to a trained TDNN network.

There have been some research efforts that utilized TDNN in phoneme classification. In [71], several TDNNs were connected together to recognize the complete set of Japanese phonemes. Many variations of the original TDNN have been proposed and studied, including an αβ-TDNN [20], an Adaptive Time-Delay Neural Network (ATNN) [42], a Frequency-time-shift-invariant TDNN (FTDNN) [59], etc.

Similar to the work of Waibel et al, we first used the TDNN network to distinguish among the voiced stop consonants /b/, /d/, /g/ and unvoiced stop consonants /p/, /t/, /k/, and then we generalized the processing to other phoneme classes in the hierarchy of Figure 5.1.

5.3.1 Hidden Layer Processing of TDNN

Unlike MLP neural networks, the inputs of a TDNN unit are multiplied by the un-delayed weights and *D* sets of delayed weights, then summed and passed through a nonlinear function, e.g., a sigmoid function. In this manner, a TDNN encodes temporal relationships within the range of *D* delays, and the values of *D* are different for each layer in the TDNN. Figure 5.4 shows a typical hidden layer of a TDNN.

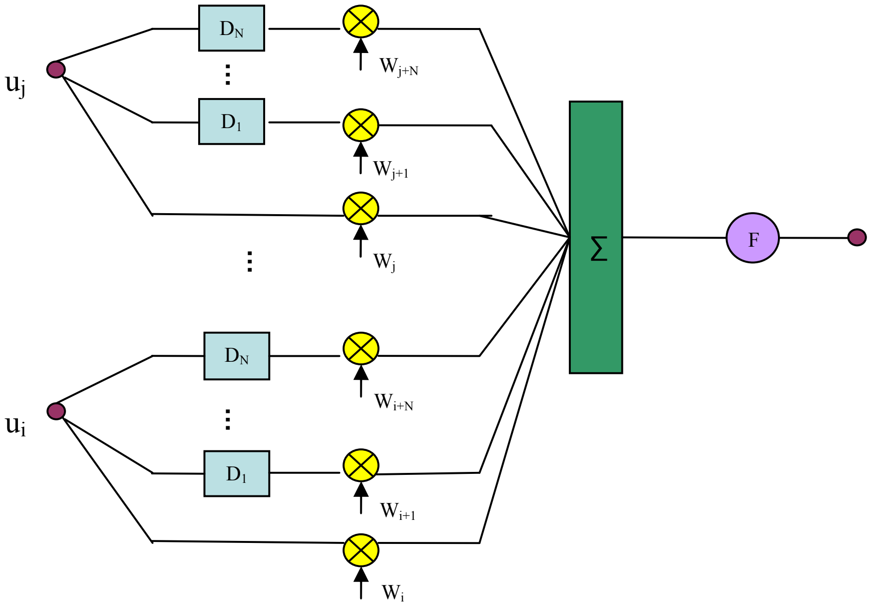


Figure 5.4 Hidden Layer Processing of TDNN [70]

5.3.2 Implementation of Our Own TDNN Toolbox

Due to an inability to find working code that implemented a standard TDNN network, we chose to design and implement a TDNN toolbox from scratch, using Netlab [48] as a reference and a starting point. The TDNN toolbox included routines for TDNN training and evaluation, and contained the following piece parts:

- **Spatial expansion of inputs and weights –** In TDNN training, the inputs and weights are spatially expanded, resulting in a spectrogram-like input pattern of the type seen at each of the layers of Figure 2.5.
- **Forward pass –** The forward pass of a TDNN is similar to that used in a standard MLP implementation, where outputs and errors are calculated.
- **Back propagation of mean squared error –** We tried both Gradient Descent (GD) and Scaled Conjugate Gradient Descent (SCGD) algorithms for back propagation of the mean squared errors [48]. For SCGD, convergence is achieved for a small number of tokens in adaptive training, in which the weights are adjusted after each sample is fed into the network. For GD, the TDNN converges for both batch training and adaptive training. The GD algorithm is simpler, although its convergence is slower than SCGD and needs more iterations. The overall training time is similar to SCGD. We used GD in the later experiments.
- **Batch training** – The weights are adjusted after all the training tokens have been calculated.
- **Staged training strategy –** The training tokens are gradually added and convergence of the network is obtained before new training tokens are added.

5.3.3 Incorporating Dynamic Features into TDNN Input

The time-delayed nature of the TDNN takes into account fine temporal information when looking for stop burst events within 3 consecutive frames of the input by using duplicated weights. Suppose the number of delayed nodes in each layer is $D^{(l)}, l=1,\cdots,L$. Each frame in the spectrogram-like input of each layer is $u_{n,m}^{(l)}, n=1,\cdots,N^{(l)}, m=1,\cdots,M^{(l)}, l=1,\cdots,L$, where $N^{(l)}$ is the dimension of each frame in layer *l*, $M^{(l)}$ is the number of frames in layer *l*, and *L* is the total number of hidden layers. At each node we use the sigmoid function

$$F(a) = \frac{1}{1+\exp(-a)} \tag{5-2}$$

For hidden layers *l*=1 and *l*=2, the activation of each node $a_{i,j}^{(l)}$ can be written as

$$a_{i,j}^{(l)} = \sum_{m=j}^{j+D^{(l)}} \sum_{n=1}^{N^{(l)}} u_{n,m}^{(l)} \cdot w_{i,j,n,m}^{(l)} + b_{i,j}^{(l)} \tag{5-3}$$

where $w_{i,j,n,m}^{(l)}$ are the weights and $b_{i,j}^{(l)}$ are the biases. Eq. (5-3) shows that the output of each node in the two hidden layers of the TDNN computes the weighted average of the hidden layer inputs using the delayed weights. From Eq. (5-3) we see that the TDNN has the ability to utilize temporal information within $D^{(l)}+1$ frames in layer *l*. Thus when we add additional temporal information to the input layer of the TDNN, if the total time span of the new parameters covers more than $D^{(1)}+1$ frames, then it potentially contains new information which can be used to improve classification performance of the TDNN.

The set of mel frequency cepstral coefficients (MFCC) and their first and second order derivatives (usually computed as simple delta and delta-delta features) are the most commonly used speech parameters in modern automatic speech recognition systems. In an effort to add additional temporal information to the TDNN we included the first and second order deltas of the MFCC coefficients to the input feature vector, using a range of

5 frames for the calculation of the first (and second) order delta (and delta-delta) MFCC features.

5.3.4 Transformation of TDNN input

In statistical pattern classification, we assume that there are *C* classes and we train a set of *C* models, one for each class. On the other hand, standard neural networks simply use the speech parameters as the input, without any knowledge of the models trained for each class. In order to take into account knowledge of the statistical models, we can define a function to transform the input vector using the parameters in each model and train one neural network based on the transformed vectors for each of the *C* classes. When evaluating all *C* neural networks, the test token is transformed *C* times according to each model, the transformed token is fed into each net, and then the maximum score is chosen as the final classification.

It is well known that neural network training is affected by the dynamic range of the input parameters. The most rapid convergence in training occurs when each input parameter has a common dynamic range. Suppose the input frame, *X*, is *N*-dimensional, i.e. $X = (x_1, x_2, ..., x_N)$. Assume that each parameter in the frame is independent of all other parameters, and approximately modeled by a Gaussian distribution. In this case each parameter, x_j, can be normalized to a zero-mean, unity variance Gaussian, $\hat{x}_j$, by the transformation.

$$\hat{x}_i = \frac{x_i - \mu_i}{\sigma_i}, i = 1, \cdots, N \qquad (5\text{-}4)$$

where $\mu_i, i = 1, \cdots, N$ is the mean, and $\sigma_i, i = 1, \cdots, N$ is the standard deviation for each parameter. In this manner, all of the training data are normalized by one mean vector and one standard deviation vector calculated from all training tokens.

Using the normalization as the transformation function, and using the mean and variance as the model parameters, we calculate $(\mu_j^{(i)}, \sigma_j^{(i)}), j = 1,..., N$ for each class $i = 1,..., C$, normalize all the input tokens on $(\mu_j^{(i)}, \sigma_j^{(i)}), j = 1,..., N$ for each class separately, and train a set of C TDNNs with each TDNN trained on the data normalized on $(\mu_j^{(i)}, \sigma_j^{(i)}), j = 1,..., N$ for one class. Finally when evaluating the TDNNs, we select the maximum score as the final classification.

5.4 Experiments

In this section, we present the results on the training and testing of TDNN for segment-based phoneme speech feature and phoneme classification. We first test the TDNN on voiced stop sounds, and then generalize the procedure to other phoneme classes in later chapters.

5.4.1 Use of TDNN for the classification of the stop consonants

The training set for /b, d, g/ and /p, t, k/ classification consisted of all the stop sounds in the TIMIT database (except for tokens from the Speaker Adaptation (SA) sentences) that were preceded by any phoneme and followed by a vowel or a diphthong (i.e., utterances of the form *CV, where C is in the set of /b, d, g/ or /p, t, k/, V is any vowel or diphthong in Figure 5.1, and * means any preceding phoneme). Other structures (e.g., VC sequences) could be used but, generally, these sequences have a more complex structure and set of dependencies to the preceding vowel, and will be investigated later. Figure 5.5 shows an example of a voiced stop /b/, with short stop gap, medium stop gap and long stop gap. The independent test set was the TIMIT TEST set of *CV utterances (without any tokens from the set of SA sentences) from the 8 dialect regions. We list the number of tokens used in the training and testing sets for stop sounds in Table 5.1.

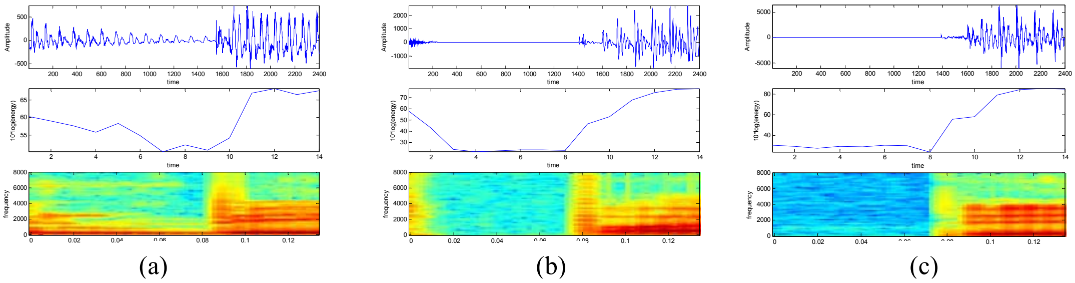


Figure 5.5 Example /b/ sounds with (a) short stop gap, (b) medium stop gap and (c) long stop gap. The first panel is the speech waveform, the second is log energy, and the third is the spectrogram.

Table 5.1 Number of tokens used in training and testing for both voiced and z stops

	Voiced stops			Unvoiced stops		
	/b/	/d/	/g/	/p/	/t/	/k/
Train	1567	1460	658	1311	2176	1900
Total	3685			5387		
Test	638	537	243	510	711	587
Total	1418			1808		

Thus we see that inherently the training set for /b, d, g/ classification via TDNN methods is highly skewed, with significantly more/b/ tokens (1567) and /d/ tokens (1460) than /g/ tokens (658). Similarly skewing of test tokens occurs in the test set. The data of unvoiced stop sounds is less skewed than that of voiced stops.

The TDNN had 2 hidden layers. The first hidden layer had 8 nodes, with delay $D^{(1)} = 2$. The second hidden layer had 3 nodes, with delay $D^{(2)} = 4$. The output layer had 3 nodes, one for each of /b/, /d/, and /g/ or /p/, /t/, and /k/. The maximum output from the 3 output nodes of the TDNN was selected as the classification for the current segment.

5.4.1.1 Staged batch training

The training procedure was a staged batch training method. Because of the time-invariant nature of the TDNN, its convergence was slow and sometimes the network didn't converge at all. Thus we trained the TDNN on a small number of tokens initially, and after it converged, gradually added more training tokens. The procedure for this type of batch training is an essential part of the work, and will be described in detail in a later section of this thesis.

Similar to Waibel [70] we used staged batch training. The numbers of training tokens used, at each new training level, were: 3, 6, 9, 24, 99, 249, 780 and finally all the *CV tokens in the training set. All the tokens were randomly selected from the training set. The first 7 training sets were balanced in the number of occurrences of /b/, /d/, and /g/ or /p/, /t/, and /k/; the last training set was unbalanced, using all the tokens of voiced stops or unvoiced stops from Table 5.1.

Figure 5.6 shows the curve of Mean Squared Error (MSE) versus the number of training epochs. It can be seen that the MSE starts each training set iteration at a relatively high value and then falls rapidly to a very low level, indicating convergence of the TDNN to a highly accurate network. As each new training set is introduced, the MSE rises to a relatively high level, but then settles back down to a low level after convergence. The last three training sets (with 249, 780 and 3685 tokens) were most problematic, showing a fair degree of jitter and ultimate convergence to a slightly high level of MSE (for the 3685 token training set), reflecting the number of outlier tokens that appeared in the large training set.

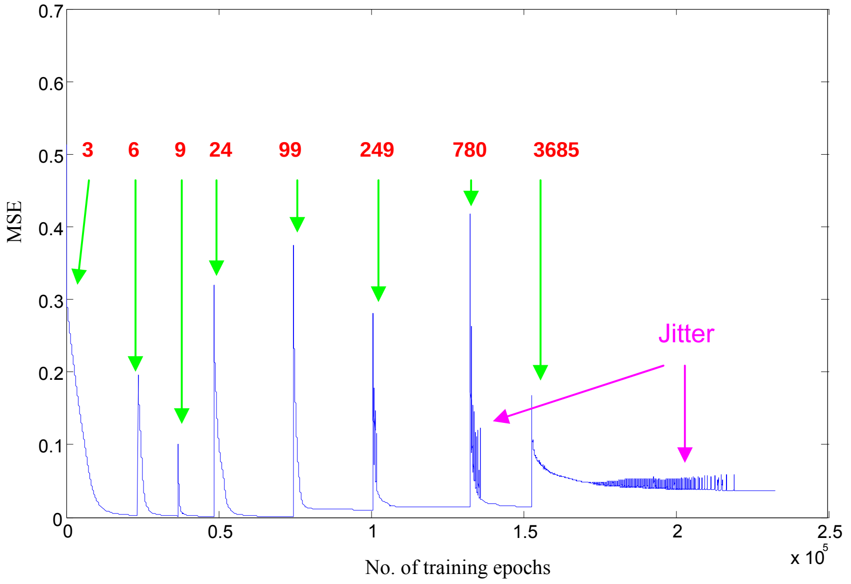


Figure 5.6 TDNN training curve showing the variation of MSE with training epoch for a sequence of training set sizes from 3 to 3685 tokens

5.4.2 Finding the Optimal Input Segment Configuration

Using the converged TDNN, we investigated the effect of varying both analysis window length and the resulting segment length on performance of the TDNN. We show the phoneme detection rates as a function of the window length (in ms) in Figure 5.7 and as a function of segment length (in frames) in Figure 5.8. We see that 10 ms windows and 15-frame segments were locally optimum in performance.

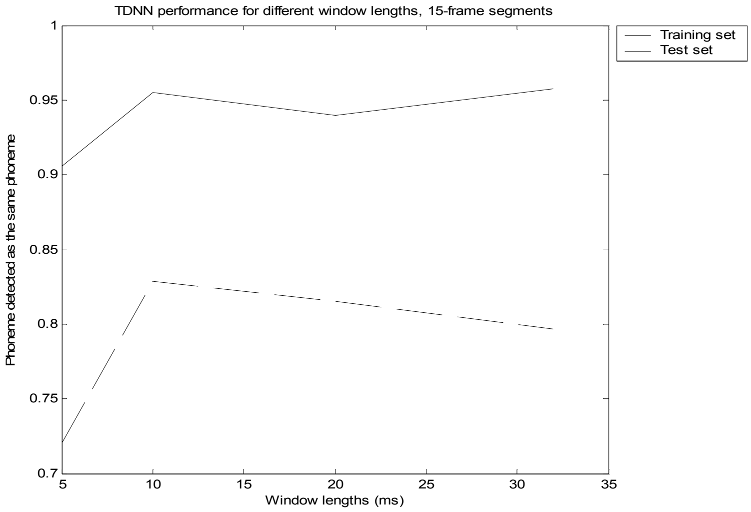


Figure 5.7 The effect of window length on phoneme detection rate using a converged TDNN network

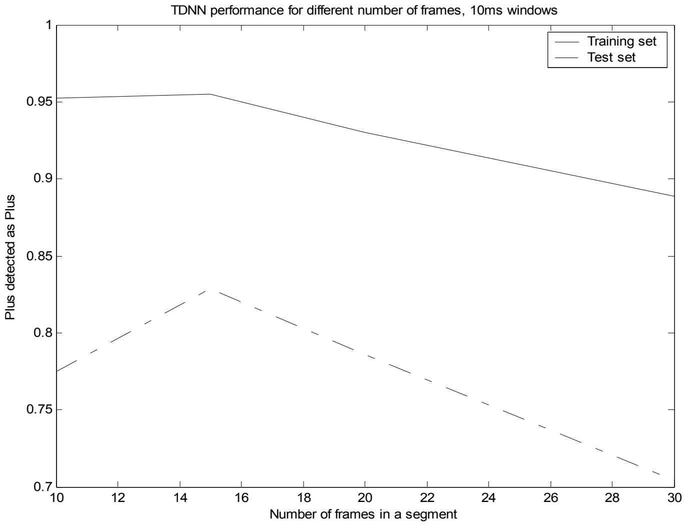


Figure 5.8 The effect of segment length on phoneme detection rate for a converged TDNN network.

5.4.3 Comparison of TDNN with a 3-Layer MLP

To determine the inherent advantage of a TDNN over a conventional MLP, we trained an MLP with an input layer that had the same segment length as that used for the TDNN, namely 150 msec, with 15 frames each having 13 MFCC coefficients, with 195 input nodes in total. The first hidden layer had 8 nodes, the second hidden layer had 3 nodes, and the output layer had 3 nodes, one each for /b/, /d/, and /g/. We measured a single mean and standard deviation for each of the 13 MFCC parameters using all the input tokens, and then we normalized each of the inputs appropriately. Table 5.2 shows the classification accuracy of the TDNN and the MLP networks on the training set and the test set when all 3685 tokens were used for training for this overall normalization method. We see that the MLP performed better than the TDNN on the training set, but it performed considerably worse on the test set.

Table 5.2 Classification accuracy for training and test sets for /b, d, g/ when using MLP and TDNN classifiers.

	Training set	Test set
MLP	96.3	82.3
TDNN	95.3	86.7

In order to compare the results of Table 5.2 with Waibel's work [70], we need to recall that Waibel's TDNN was trained on Japanese CV utterances, where there were only 5 vowels; hence there were only 15 possible CV combinations. Also, Waibel's TDNNs were speaker dependent, with one TDNN trained for each speaker. The English language has 11 vowels and 4 diphthongs (in the 40-phoneme alphabet), so there are (11+4)*3=45 possible CV combinations. Further, the TDNN that we created was trained on multiple speakers and therefore was speaker independent. Hence there is no simple direct comparison of results; however we see that the performance obtained using the TDNN on the test set is quite high, considering the major differences in system specifications.

5.4.3.1 Results when using MFCC and its delta and delta-delta features

The delta and delta-delta parameters were calculated using standard definitions over a 5-frame window. A comparison of TDNN network classification performance (in terms of percentage accuracy of classification) is given in Table 5.3.

Table 5.3 TDNN classification accuracy (%) for /b,d,g/ on training and test sets using MFCC and MFCC+Δ+ΔΔ feature sets.

Features	Training set	Test set
MFCC	95.3	86.7
MFCC+Δ+ΔΔ	98.8	87.7

From Table 5.3 we see that the TDNN performance is improved by a small amount by incorporating the dynamic features.

5.4.4 Results with transformed TDNN inputs

As a second normalization method we calculated one mean and one standard deviation for each of the 39 parameters for each of the 3 voiced stop consonants, ($\mu^{(i)}$, $\sigma^{(i)}$, i=1,…,3). Then we trained the TDNN on the individually normalized training tokens. The resulting classification accuracy was 100% on the training set. Then we normalized each test token using the correct mean and standard deviation, as calculated from the training set for this phoneme class, and we achieved 100% classification accuracy on the test set. This procedure is, of course, invalid because when we do real world testing, we do not know which $\mu^{(i)}$ and $\sigma^{(i)}$ to use for feature normalization; otherwise we would have known the phoneme class. This test was performed just to determine whether the phonemes could be clearly separated when normalizing each phoneme class separately to the appropriate standard normal distribution.

As a valid evaluation we normalized the test token three times, using each of (μ_b, σ_b), (μ_d, σ_d) and (μ_g, σ_g), and evaluated the TDNN outputs using the TDNN trained above for each of the separately normalized tokens, and selected the maximum score for classification, but only achieved 80% classification accuracy on the test set.

Using the transformation method described in the previous section, we trained one TDNN using tokens normalized on (μ_b, σ_b), another TDNN using tokens normalized on (μ_d, σ_d) and still another one using (μ_g, σ_g). When we did the testing, we normalized the test token using each of (μ_b, σ_b), (μ_d, σ_d) and (μ_g, σ_g) separately and calculated the outputs from the three TDNNs. Then we selected the maximum score as the final classification result. Using this method, we achieved 90.9% accuracy for voiced stop classification on the 1418-token test set, i.e., an improvement of 3.2% in accuracy.

Similarly, for unvoiced stop consonant classification, we calculated (μ_p, σ_p) on the /p/ training tokens, (μ_t, σ_t) on the /t/ training tokens, and (μ_k, σ_k) on the /k/ training tokens. Then we trained three separate TDNN's with all the input tokens normalized on each set of mean and standard deviation. We again tested the performance of the resulting TDNN's by selecting the maximum output from the 3 TDNNs as the final classification. We achieved 98.6% classification accuracy on the training set and 91.9% accuracy on the 1808-token test set. The results are shown in Table 5.4, where N01-all denotes normalizing the training tokens using one global mean and one global standard deviation, N01-3 denotes using three means and standard deviations to normalize the inputs and train three TDNN's.

Table 5.4 Classification accuracies (%) of different feature normalization methods on both voiced and unvoiced stop consonants.

		Training set	Test set
Voiced stops /b, d, g/	N01-all	98.8	87.7
	N01-3	97.9	***90.9***
Unvoiced stops /p, t, k/	N01-all	99.1	88.6
	N01-3	98.6	***91.9***

5.4.5 Discussion

To compare our results with other research efforts for the classification of stop consonants, we first need to note the statistical divergence between different classification systems. In Ali's work on stop consonant classification [4], the test was done on 7 dialect regions consisting of 1200 stops, and he achieved 90% accuracy for place of articulation detection of the 6 stop consonants. In Suchato's work on stop consonant classification [68], the database was only 2 male speakers and 2 female speakers, and he achieved 92.1% accuracy on 4007 stops. Zhang et al. achieved 88.1% accuracy on the place of articulation detection of all the stops in the TIMIT test set of 5725 stops [73]. All of the above research efforts were conducted using a number of hand selected acoustic (spectral and/or temporal) features such as formant tracks.

In our work, the classification of place of articulation detection was partitioned into voiced stops and unvoiced stops, and we achieved 90.9% classification accuracy on 1418 voiced stops and 91.9% accuracy on 1808 unvoiced stop tokens of the *CV form from the 8 dialect regions in the TIMIT database. The average place of articulation classification accuracy would be 91.5%. This performance was achieved using only MFCC and its delta and delta-delta features, without any specific acoustic phonetic measurements especially tailored to the problem of stop consonant classification. The results presented here represent state-of-the-art performance for stop consonants place classification, on a large database, using algorithmic methods.

5.5 Conclusion

In this chanper we studied the class of Time-Delay Neural Networks and measured its performance for classification of voiced and unvoiced stop consonants in English. We showed that TDNNs, trained on the TIMIT database, generalized much better on an unknown test set than a traditional MLP network without the delay features. We found that the use of MFCC and its delta features provided additional, and highly reliable information for stop consonant classification. We used a simple normalization procedure as the transformation function of the neural net feature inputs, and found that by training the classifier based on different normalization methods, the classification accuracy improved above that obtained from uniformly normalized input features. Overall we achieved 90.9% classification accuracy on voiced stops and 91.9% classification accuracy on unvoiced stops. Our experiments were conducted without any specific acoustic information specially tailored for the classification of stop consonants.

Chapter 6

Combined Frame and Segment-Based Methods for Speech Feature and Phoneme Classification

Phonemes in the English language can be represented using either parallel or hierarchical distinctive speech features. There have been a number of efforts to integrate multiple information sources but none of these efforts addressed the issue of combining multiple sets of articulatory/linguistic features with different organization topologies. In this chapter, we combine frame-based methods for parallel speech feature detection and segment-based methods for hierarchical phoneme classification to improve the overall phoneme classification performance, in which different feature organization topologies are merged at each level of the phoneme classification hierarchy including the broad class level and the narrow phone level. We first present a mathematical framework that combines the parallel feature detection and hierarchical phoneme classification, and then show that this results in an improvement in the overall classification performance in the hierarchical phoneme classification task.

6.1 Background

Automatic speech recognition (ASR) based on the use of acoustic-phonetic features has been gaining popularity in recent years ([17], [23], [35], [36], [40]). When representing all the phonemes in the English language, the organization of the speech features can be either parallel or hierarchical. A parallel structure, as used in ([9], [35]), is a flat representation of all speech attributes, and it assumes that all speech features are independent, whereas in reality they are not. The flat representation avoids the problem of error propagation from different levels of a hierarchy, but suffers from the problem of not utilizing layers of information about the sound so as to reduce the uncertainty as to sound class. A hierarchical structure, as used in [23], is more efficient in representing all the sounds of a language (the set of spoken phonemes), but suffers from the problem of error propagation from higher levels to lower levels (i.e., errors made at a high level of the hierarchy propagate to lower levels with no clear correction mechanism). In this chapter, we combine the parallel speech feature organization and the hierarchical organization and show that this results in an improvement in the overall classification performance.

There have been several research efforts that tried to combine different features in order to improve speech recognition accuracy. Most of these efforts combined articulatory features with standard acoustic features based on MFCC, PLP, etc. The MIT SUMMIT system [19] integrated landmark-based and segment-based feature streams and the recognizer performed segmentation and classification jointly. The combination of features can be performed at different levels (e.g., frame, phone, word) in the classification system, and Kirchhoff et al. discussed a few rules for combination at the phone level [37]. In this chapter, we tried to merge different feature organization topologies at each level of the phoneme classification hierarchy including the broad class level and the narrow phone level.

Frame-based methods and segment-based methods are two typical approaches to automatic speech recognition. The ASAT (Automatic Speech Attribute Transcription) methodology uses a detection method based on frame-wise speech attributes for phoneme detection. Whenever the likelihood of a particular feature or phoneme is above a predetermined threshold, the feature or phoneme is detected as present. In classification-based methods, the task is to classify a segment within a given class of features or phonemes. Classification can be performed after segmentation (the so-called segmentation-and-labeling approach), or segmentation and classification can be performed jointly and suitably optimized ([19], [29], [50]).

In this chapter, we combine frame-based methods for parallel speech feature detection and segment-based methods for hierarchical phoneme classification to improve the overall classification performance.

6.2 Mathematical Framework

In this section, we present the mathematical framework for the combination of parallel and hierarchical phoneme classification. We first formulate the frame-based parallel feature detection and the hierarchical phoneme classification approaches. Then we discuss the incorporation of the two processes together to form a unified phoneme classification system.

6.2.1 Frame-based parallel speech attribute detection

From Section 4.2 in Chapter 4, we know that in a frame-based parallel speech attribute detection systems, a phoneme can be represented by a set of N parallel binary speech features, i.e., $e_1,\dots,e_N$ where

$$a \overset{def}{=} (e_1, e_2, \cdots, e_N), \quad e_j = 0,1, \quad j = 1, \cdots, N \tag{6-1}$$

When we assume that the features are independent of each other, the *a posteriori* probability $P(a \mid x)$ that denotes the detection of phonemes can be written as

$$P(a \mid x) = P(e_1, e_2, \cdots e_N \mid x) = \prod_{j=1}^{N} P(e_j \mid x) \qquad (6\text{-}2)$$

where a is a phoneme hypothesis for a given input speech frame x. The posterior probabilities $P(e_j \mid x), j = 1, \cdots, N$ denote the probability of detection of features.

In Chapter 4, we trained 14 frame-based multi-layer perceptrons (MLPs) in parallel to detect each of the 14 binary valued sound pattern of English (SPE) speech features.

6.2.2 Segment-based hierarchical phoneme classification

From Section 5.1 in Chapter 5, we know that in segment-based hierarchical speech feature classification, we need to estimate the *a posteriori* probability $P(c \mid \underline{y})$ of segment $\underline{y}$ belonging to class c, where we have used the result that for all the L number of child classes of current class, $\sum_{k=1}^{L} P(c_k \mid \underline{y}) = 1$.

Table 6.1 Hierarchical feature values

Feature name	Feature values
Top class	V, Consonant, Silence
V	Vowel, Diphthong, Semivowel
Vowel-LH	high, mid, low
Vowel-FB	front, mid, back
Diphthong	/aw/, /ay/, /ey/, /oy/
Semivowel	/w/, /l/, /r/, /y/
Consonant	Nasal, Stop, Fricative, Affricate, Whisper (/h/)
Nasal	/m/, /n/, /ng/
Stop-place	Labial (/p/,/b/), Alveolar (/t/,/d/), Velar (/k/,/g/)
Stop-voicing	Voiced (/b/,/d/,/g/), Unvoiced (/p/,/t/,/k/)
Fric-place	Labiodental (/f/,/v/), Dental (/th/,/dh/), Alveolar (/s/,/z/), Palatal (/sh/,/zh/)
Fric-voicing	Voiced (/v/,/dh/,/z/,/zh/), Unvoiced (/f/,/th/,/s/,/sh/)
Affricate	/ch/, /jh/

We used Time-Delay Neural Networks developed in Chapter 5 to estimate $P(c \mid \underline{y})$. The features used are listed in Table 6.1.

6.2.3 Combination of parallel and hierarchical phoneme classifications

The reason for incorporating frame-based feature detection into segment-based phoneme classification is that the TDNN (operating on segment-length utterances) provides much higher classification performance than the MLP (operating on single frames). Hence the goal is to improve, whenever possible, the TDNN classification performance using parallel feature detection. In order to do this we first convert the frame-based feature detection method into a segment-based feature classification method and then combine the two methods.

6.2.3.1 Incorporating frame-based feature detection in segment classification

For the hierarchical classification problem described in Eq. (5-1), since the parallel and hierarchical feature organizations are quite different, it is likely that we don't have a common set of classes in both methods. In order to calculate the *a posteriori* probability $P(c \mid \underline{y})$ from parallel features for segment $\underline{y}$, we must decompose the class c into its phoneme constituents

$$A_c = (a_1, a_2, \cdots a_{N_c}) \tag{6-3}$$

where class c consists of a subset of all phonemes, denoted by $\boldsymbol{A_c}$. Then the posterior probability $P(c \mid \underline{y})$ can be rewritten as

$$P(c \mid \underline{y}) = \sum_{a \in A_c} P(a \mid \underline{y}) \Big/ \sum_{c \in C} \sum_{a \in A_c} P(a \mid \underline{y}) \tag{6-4}$$

In equation (6-4), the term $P(a \mid \underline{y})$ is the probability that the segment is detected as phoneme a within a class c. The denominator is used to ensure that the posterior probabilities for each of the classes in set C, given the segment, sum up to 1.

In frame-based methods, it is commonly assumed that the frames are independent and identically distributed within the same phonemic state. Assuming that there are L frames within the phoneme segment, we use the average of the frame-wise posterior probabilities as the "segment" posterior probability of phoneme a given segment $\underline{y}$

$$P(a \mid \underline{y}) = \frac{1}{L}\sum_{l=1}^{L} P(a \mid x_l) \qquad (6\text{-}5)$$

For one single frame x_l within the phoneme segment $\underline{y}$, using the parallel feature representation of (6-1) and (6-2), then equation (6-5) can be transformed to the form

$$P(a \mid \underline{y}) = \frac{1}{L}\sum_{l=1}^{L}\prod_{j=1}^{N} P(e_j \mid x_l) \qquad l = 1, \cdots, L \qquad (6\text{-}6)$$

In the hierarchical phoneme classification, except for the top class, all other classes simply consist of a fraction of all the phonemes in the alphabet. When the number of phonemes within the C classes to be classified is less than the total number of M phonemes in the alphabet, we do not need all of the N speech features to determine each phoneme. We denote the number of features needed for distinguishing C classes of phonemes as N_C, where $N_C \leq N$. Then (6-6) is further simplified as

$$P(a \mid \underline{y}) = \frac{1}{L}\sum_{l=1}^{L}\prod_{j=1}^{N_C} P(e_j \mid x_l) \qquad l = 1, \cdots, L \qquad (6\text{-}7)$$

For example, the SPE feature values for semivowel classification are listed in Table 6.2. From the table, we can see that only features numbered (1), (3), (4), (6), (7) and (8) are different for the four semivowels, whereas each of the other SPE features has identical values for all four phonemes and they don't provide information for classification, and thus are discarded for the semivowel class. N_C=6 in this case.

Table 6.2 SPE feature values for semivowels

SPE No.		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Values	l	**0**	1	**1**	**0**	0	**1**	**1**	**0**	0	1	1	0	0	0
	r	**1**	1	**0**	**0**	0	**0**	**1**	**0**	0	1	1	0	0	0
	w	**0**	1	**1**	**1**	0	**0**	**0**	**1**	0	1	1	0	0	0
	y	**0**	1	**1**	**0**	0	**0**	**0**	**0**	0	1	1	0	0	0

SPE No.1-vocalic, 2-consonantal, 3- high, 4-back, 5-low, 6-anterior, 7-coronal, 8-round, 9-tense, 10-voice, 11-continuant, 12-nasal, 13-strident, 14-silence.

6.2.3.2 Combination method

Our initial attempt at merging the frame-based results (estimated via MLP methods) and the segment-based results (estimated via TDNN methods) was to linearly combine the two sets of *a posteriori* probabilities giving

$$\begin{aligned} &P(c\,|\,\underline{y}) = \alpha_1 \cdot P_{TDNN}(c\,|\,\underline{y}) + \alpha_2 \cdot P_{MLP}(c\,|\,\underline{y}) \\ &\alpha_1 + \alpha_2 = 1, \qquad \alpha_1 \geq 0, \qquad \alpha_2 \geq 0. \end{aligned} \tag{6-8}$$

where $P_{TDNN}(c\,|\,\underline{y})$ is the *a posteriori* probability of class c given segment $\underline{y}$ in equation (6-2); $P_{MLP}(c\,|\,\underline{y})$ is the *a posteriori* probability of averaged frame-wise scores from equation (6) and α_1 and α_2 are appropriate weights.

Here we sum up the *a posteriori* probabilities but not the log likelihoods that are commonly used in ASR, since we use artificial neural networks that directly estimate *a posteriori* probabilities.

6.3 Experiments

Our experiments consist of 3 parts: the detection of the 14 parallel SPE features using MLPs, the classification of hierarchical speech features and phonemes using TDNNs, and finally the combination of the above two approaches. All experiments were conducted on the TIMIT database. The training set was the TIMIT TRAIN set and the test set was the independent TIMIT TEST set. Both training and testing sets consist of all the sentences of the 8 dialect regions except for the SA sentences. The speech feature vector consisted of 13 MFCCs along with 13 delta MFCCs and 13 delta delta MFCCs for a total feature vector size of 39 components. We used 10 msec Hamming windows for the calculation of the MFCC parameters, and the frame rate was 200 Hz. Adjacent frames were averaged, resulting in a 100 Hz frame rate.

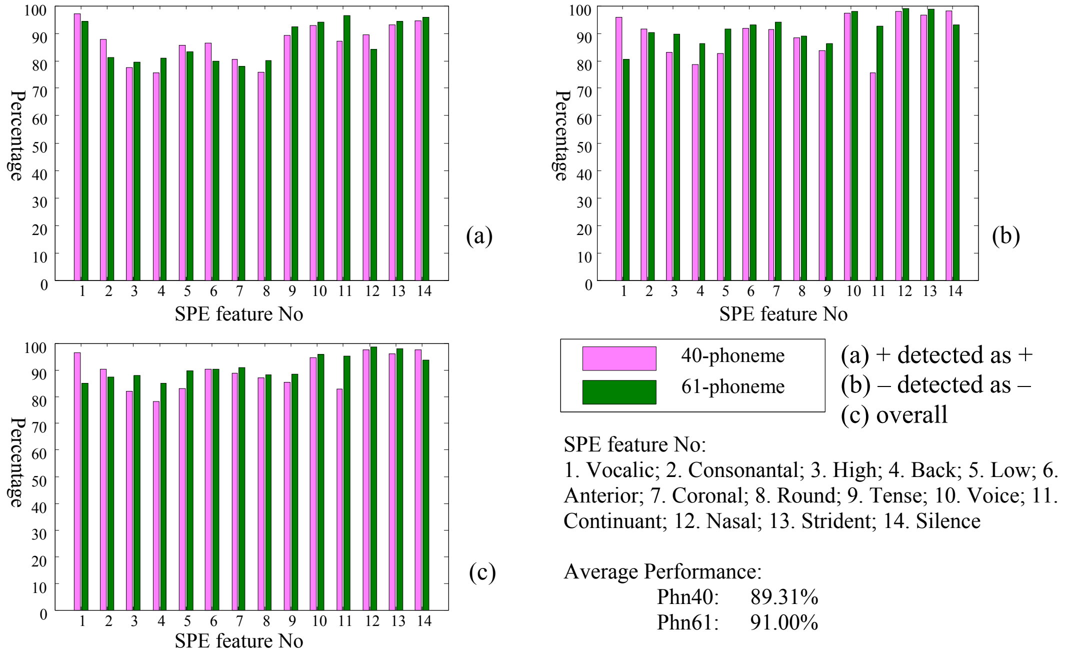
6.3.1 Parallel Speech Feature Detection

We extended our previous work on frame-based feature detection in which 14 MLPs were trained to detect each of the 14 SPE features for the DARPA 61 phoneme alphabet using single frames of 13 MFCCs using a balanced training set. We compared the performance when using 13 MFCCs only and when using MFCC and its delta and delta-delta parameters, and found that single frame 13-component MFCCs performed better in frame-wise feature detection.

In this chapter, we used a reduced phoneme alphabet consisting of 39 phonemes plus silence. Due to memory limits and the high correlation between adjacent frames, we sampled one out of every four consecutive frames for training and testing. Phoneme boundary frames and immediately adjacent boundary frames were excluded from both training and testing sets. We limited the training data size to 48,000 frames and the test data size to 33,020 frames. Using balanced training data, the feature detection performance of the 14 SPE features for the 40-phoneme alphabet and 61-phoneme alphabet are measured and is illustrated in Figure 6.1. The numerical results for SPE feature detection using the 40-phoneme alphabet are given in Table 6.3.

Table 6.3 SPE feature detection performance using MLPs (% correct)

SPE No.	1	2	3	4	5	6	7
Performance	96.5	90.3	82.0	78.2	83.1	90.4	88.8
SPE No.	8	9	10	11	12	13	14
Performance	87.2	85.4	94.6	82.8	97.6	96.0	97.5



(a) + detected as +
(b) – detected as –
(c) overall

SPE feature No:
1. Vocalic; 2. Consonantal; 3. High; 4. Back; 5. Low; 6. Anterior; 7. Coronal; 8. Round; 9. Tense; 10. Voice; 11. Continuant; 12. Nasal; 13. Strident; 14. Silence

Average Performance:
Phn40: 89.31%
Phn61: 91.00%

Figure 6.1 SPE feature detection performance for the 40-phoneme alphabet and 61-phoneme alphabet

From Figure 6.1, we can see that the overall performance of feature detection for the 61-phoneme alphabet is 1.7% higher than the 40-phoneme alphabet. Using the 40-

phoneme alphabet, we can see that there are 7 SPE features whose detection performance was above 90%; 6 SPE features with detection performance between 80 and 90%; and one between 70 and 80%. Although the individual performance is satisfactory for each feature group, when multiplying the *a posteriori* probabilities of all the 14 features together, we get rather poor frame-wise phoneme classification results.

6.3.2 Use of frame-based MLP detectors in segment-based phoneme classification

In this experiment, we chose the phoneme segment consisting of all the frames within the duration of the phoneme. When the total number of phonemes within a class was less than the number of phonemes within the alphabet, not all SPE features were needed to distinguish each phoneme. The minimum set of SPE features is the set that uniquely distinguishes all phonemes within that class. We first chose the minimum number of SPE features to use for classification, and then gradually added more features to see if performance could be improved by adding additional but redundant information. For example, for semivowel classification, there are only 4 phonemes but 6 of the SPE features can be used to distinguish them. The feature detection performances for semivowels for the 6 features numbered 1,3,4,6,7,8 in Table 6.2 were 77.15%, 79.97%, 64.51%, 72.23%, 51.98%, and 59.32% respectively. The 6 detector performance scores were sorted from highest to lowest and the first 4 features could uniquely determine each phoneme. The performance scores obtained by using 4 of 6 features, 5 out of 6 features, and all 6 features were compared, and we determined the best performance was achieved using 6 features. We applied this strategy to each feature class in the hierarchy, and the results are listed in the third column in Table 6.4. If we compare these results with the TDNN classification results, we see that TDNNs always performed better than MLPs.

In our experiments we also noted that the performance was better for certain features when the phoneme boundary frames were discarded, and for some features the performance was better when all frames within the phoneme were included.

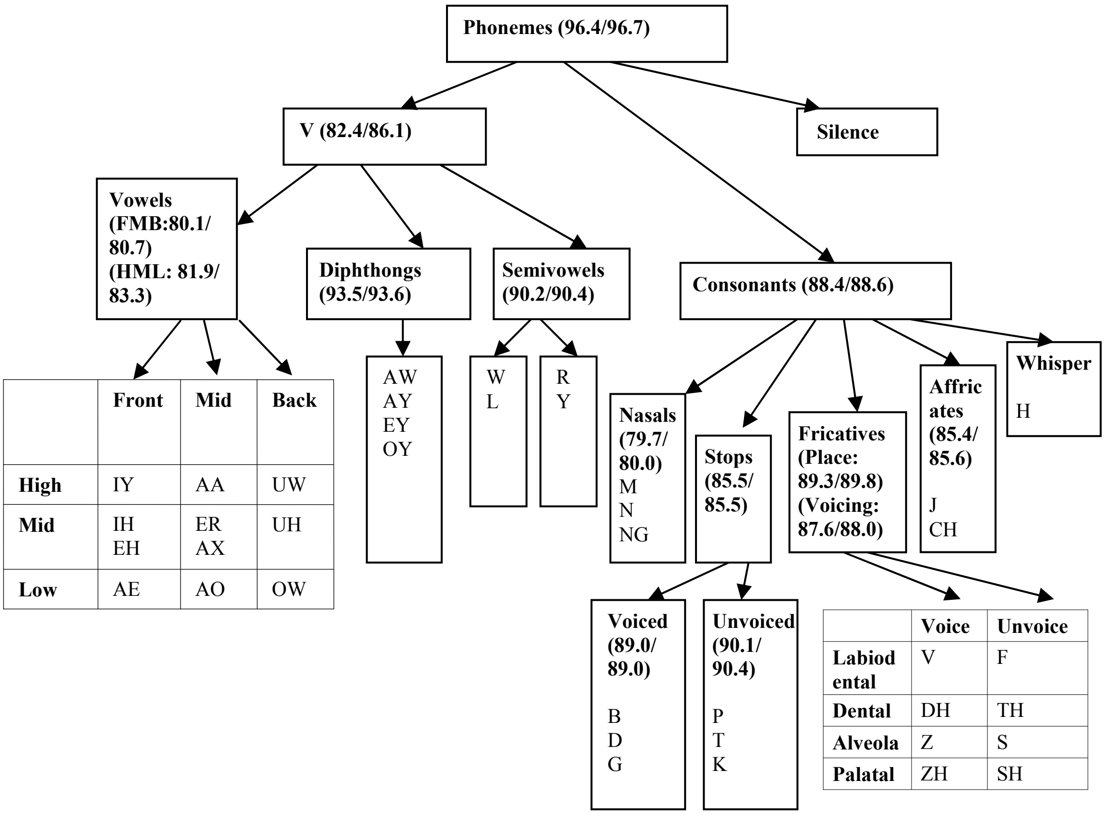
6.3.3 Linear Interpolation of TDNN and MLP Results

After we obtained the segment-based classification scores using TDNN and the frame-based detection scores (using MLP) converted to segment scores, we combined the two sets of scores using a weighted sum. The combination was optimized for each class in the phoneme hierarchy and the weights that gave the best classification performance were chosen. The results are listed in the rightmost column of Table 6.4 and also shown in Figure 6.2.

From these results, we can see that the overall performance scores were improved (sometimes only by very small amounts) by combining the MLP and TDNN classification results. Some classification performance scores improved significantly, including “Top Class” (V-Consonant-Silence), “V” (Vowels-Diphthongs-Semivowels), “Vowel-LH” (low-mid-high feature for vowels) and “Vowel-FB” (front-mid-back feature for vowels). Other classes only improved slightly, and one class didn’t improve at all. Small or no improvement occurred when the MLP performance was very poor by itself or when the MLP didn’t provide complementary information for classification.

Table 6.4 Classification performance using TDNN, MLP, and combination of the two (% correct)

Feature name	TDNN	MLP	TDNN+MLP
Top Class	96.4	78.1	96.7
V	82.4	67.2	86.1
Vowel-LH	81.9	72.2	83.3
Vowel-FB	80.1	47.8	80.7
Diphthong	93.5	66.4	93.6
Semivowel	90.2	70.4	90.4
Consonant	88.4	66.0	88.6
Nasal	79.7	53.2	80.0
Stop-place	89.6	50.9	89.8
Stop-voicing	85.5	60.1	85.5
Fric-place	89.3	77.1	89.8
Fric-voicing	87.6	79.8	88.0
Affricate	85.4	74.4	85.6


Phonemes (96.4/96.7)
V (82.4/86.1)
Silence
Vowels (FMB:80.1/ 80.7) (HML: 81.9/ 83.3)
Diphthongs (93.5/93.6)
Semivowels (90.2/90.4)
Consonants (88.4/88.6)
AW AY EY OY
W L
R Y
Nasals (79.7/ 80.0) M N NG
Stops (85.5/ 85.5)
Fricatives (Place: 89.3/89.8) (Voicing: 87.6/88.0)
Affric ates (85.4/ 85.6) J CH
Whisper H
Voiced (89.0/ 89.0) B D G
Unvoiced (90.1/ 90.4) P T K

	Front	Mid	Back
High	IY	AA	UW
Mid	IH EH	ER AX	UH
Low	AE	AO	OW

	Voice	Unvoice
Labiod ental	V	F
Dental	DH	TH
Alveola	Z	S
Palatal	ZH	SH

Figure 6.2 Classification performances using TDNN and using the combination method (performance of TDNN/performance of TDNN+MLP)

To compare our results with other research efforts in phoneme classification, we first note that feature definition in various research efforts is often quite different from that in this paper. With that limitation it can be shown that the results in Table 6.4 compare favorably with results from other recent studies, e.g. landmark-based speech recognition [23].

6.4 Discussion

In this section we described a method for combining *a posteriori* probabilities from frame-based parallel feature detection and segment-based hierarchical phoneme classification. The mathematical framework for combining the *a posteriori* probability scores was formulated and performance scores were obtained on the TIMIT database. For the frame-based system, single frame MLPs trained for the detection of 14 parallel SPE features were converted to segment-based hierarchical phoneme classification probabilities. The segment-based TDNN classification probabilities were linearly combined (with an optimal set of weighting coefficients) with the MLP probabilities giving improved classification performance scores for all phoneme classes, although the improvement scores were marginal for some classes.

One reason that the performance scores didn't improve much for some classes is that the TDNN segment-based classifier is a much better classifier than the traditional frame-based MLP. Another reason why the segment-based method performed better than frame-based method in classification is that frame-wise feature detectors use no prior knowledge of the sentence and need the entire training data to train the detector, while segment-based methods have prior knowledge of what group of features or phonemes the current segment belongs to and classification can be tailored to fit the specific class. The training of a classification-based method only needs the speech parameters relevant to a group of features or phonemes within a class. In our experiments, we used previously trained parallel speech feature detectors and transformed them to classify segments. In future studies, in order to refine this procedure, we will re-train each SPE feature detector for detection within just the one class that consists of only a subset of the complete phoneme alphabet. Since the classification problem has more flexible solutions we will study the effect of using different acoustic parameters and different algorithms for each class.

Chapter 7

Combined Frame and Segment-Based Approaches to Speech Attribute Detection

In this chapter, we investigate the combination of frame and segment-based methods for speech attribute detection. We use segment-based TDNN classifiers for the detection of 14 SPE features in parallel, and linearly combine the results from TDNN detection and MLP detection. Results show that the combined system provides better detection performance than either TDNN or MLP detectors.

7.1 Background

In frame-based parallel speech feature detection, each attribute can be flexibly estimated using the same or different methods using various sorts of speech parameters. There are various forms of speech attributes, and the ASAT project utilizes several of them to improve the performance of the traditional HMM based system. In Chapter 4, we trained 14 Multi-Layer Perceptrons for the detection of the 14 Sound Pattern of English features. In this chapter, we tried to improve the MLP detection performance by combining it with the TDNN detection results.

In segment-based hierarchical speech feature and phoneme classification, the speech waveform is firstly segmented, and then each segment is classified within a certain range of speech features or phonemes. The ability of segments to capture long term speech features and the transient features is one of the major attractions for this type of method. But one drawback of this approach is that this method needs pre-segmentation, which cannot be very precise. Further, the segmentation error can propagate throughout the following recognition process.

There have been some research efforts that tried to incorporate both short-term and long term features in speech recognition and other tasks. For example, Zhao and Morgan used multi-stream spectro-temporal features for robust speech recognition and achieved 30% error reduction in word error rate compared with the MFCC only recognition system [74]. Fukuda et al used different window lengths to calculate delta cepstrum features for phoneme recognition, and their results were better than pure MFCC features [18]. Most of those efforts use various acoustic parameters other than MFCCs, and none of them addressed the importance of incorporating linguistic attributes using different attribute organization topologies.

In our previous chapters, we have investigated frame-based parallel speech attribute detection and segment-based speech feature and phoneme classification. In this chapter, we study the combination of frame and segment-based approaches to be used in the event detection based ASAT paradigm.

7.2 Mathematical Framework

In this section, we present the mathematical framework for the combination of parallel and hierarchical speech feature detection. We first recall the frame-based parallel feature detection and the hierarchical phoneme classification approaches from previous chapters. Then we discuss the incorporation of the two processes to form a unified parallel speech feature detection system.

To combine the two approaches, we need to convert segment-based phoneme classification into frame-based speech attribute detection. In doing so, we first use the segment-based TDNN classification result as the frame-wise speech feature and phoneme detection score for the frame in the center of the TDNN input segment. To calculate the attribute detection score for that frame using TDNN, we decompose each parallel speech attribute into two phoneme sets, one plus set and one minus set, according to the plus value or minus value of the attribute. Then the TDNN classification result for each phoneme in the plus (or minus) set is summed up to calculate the plus (or minus) detection score of the attribute.

7.2.1 Frame-based parallel speech feature detection

In frame-based speech attribute detection, the task can be viewed as trying to estimate the *a posteriori* probabilities of speech attributes for each frame *x*, i.e.,

$$P(e_j \mid x), j = 1,...,N \tag{7-1}$$

where $\{e_j, j = 1,...,N\}$ is a set of *N* parallel speech attributes, which can be binary-valued or multi-valued. Here we use the 14 binary-valued Sound Pattern of English (SPE) features as speech attributes (*N*=14 in this case); thus we have $e_j = 1$ (denoted as e_j^+) or $e_j = 0$ (denoted as e_j^-). The set $\{P(e_j \mid x), j = 1,...,N\}$ is a set of *a posteriori* probabilities corresponding with the detection of each attribute e_j given frame x.

Using the approach outlined above, we trained a set of 14 Multi-Layer Perceptrons (MLPs) to estimate the posterior probabilities of $\{P(e_j \mid x), j = 1,...,N\}$ (Please refer to Chapter 4).

7.2.2 Segment-Based Hierarchical Speech Feature Classification

In the hierarchical phoneme organization, we assume that each of the entire set of English phonemes can be uniquely represented by hierarchical classes. All the phonemes are at the leaf nodes, and the broad classes are at the internal nodes in the phoneme hierarchy. Each phoneme a_i can be represented by the path from the root node $h_{i,1}$ to the leaf node h_{i,H_i} in the phoneme hierarchy:

$$a_i \quad \propto \quad \{h_{i,1}, h_{i,2}, ..., h_{i,H_i}\} \tag{7-2}$$

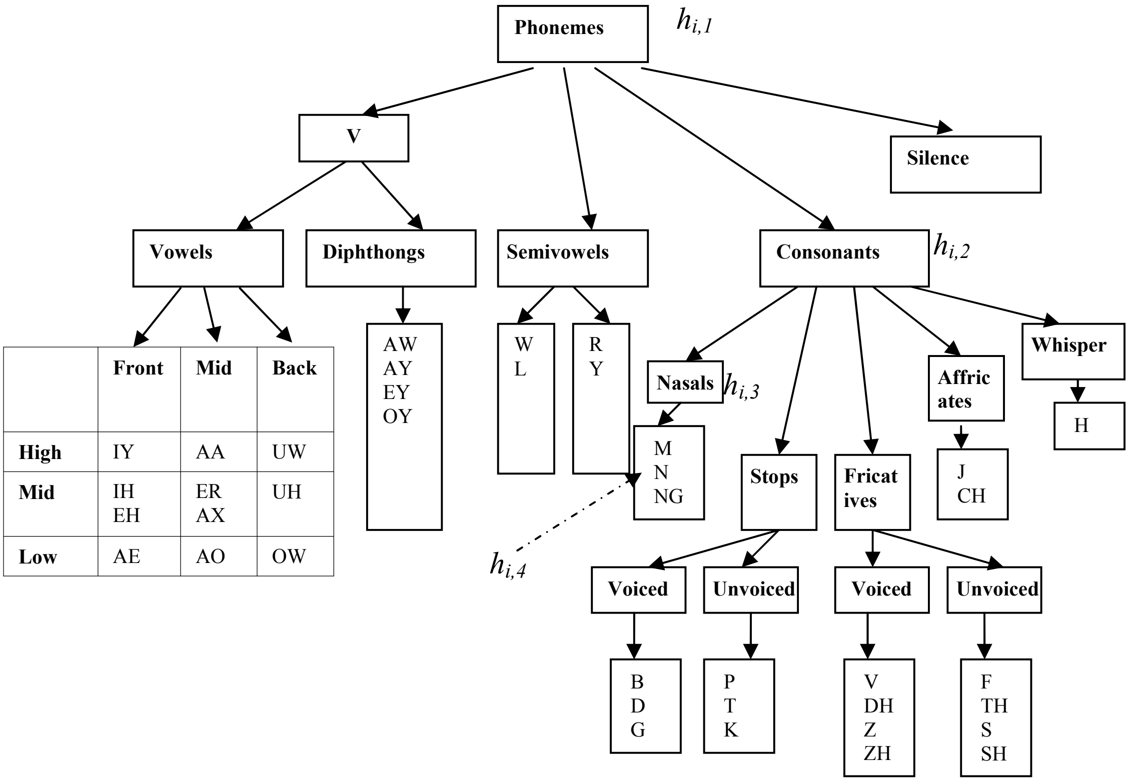
where H_i is the total number of nodes from the root class to the specific phoneme class. For example, in Figure 7.1, the nasal sound 'N' corresponds with a path containing 4 nodes {'Phonemes', 'Consonants', 'Nasals', 'N'}, and here H_i=4.

In segment-based hierarchical phoneme classification, given a segment $\underline{y}$, we denote the posterior probability of segment $\underline{y}$ belonging to class c as $P(c \mid \underline{y})$, where c can be a broad class in the internal node of the hierarchical phoneme organization, or a set of phonemes in the leaf node in the phoneme hierarchy.

Using the phoneme representation of (7-2), the posterior probability of phoneme a_i given the segment $\underline{y}$ for hierarchical phoneme classification algorithm can be written as

$$P(a_i \mid \underline{y}) = P(h_{i,1}, ..., h_{i,H_i} \mid \underline{y}) = P(h_{i,1} \mid \underline{y}) \prod_{k=2}^{H_i} P(h_{i,k} \mid \underline{y}, h_{i,k-1}) \tag{7-3}$$

We used a Time-Delay Neural Network (TDNN) toolbox developed in Chapter 5 to estimate the *a posteriori* probabilities of $P(h_{i,k} \mid \underline{y}, h_{i,k-1})$. The classes used are listed in Table 6.1.



	Front	Mid	Back
High	IY	AA	UW
Mid	IH EH	ER AX	UH
Low	AE	AO	OW

Figure 7.1 Hierarchical phoneme classification example for phoneme /N/

7.2.3 Incorporating segment-based phoneme classification into frame-based speech attribute detection

The reason for incorporating segment-based speech feature and phoneme classification into frame-based speech attribute detection is that the ASAT project is a detection based approach, and since TDNN has been shown to be a highly effective classifier, we are interested in understanding its applicability to the detection domain to improve frame-based speech feature detection.

7.2.3.1 Converting segment-based phoneme classification into frame-based speech feature detection

In order to combine segment and frame based approaches, first we need to convert segment-based TDNN for phoneme classification to frame-based speech attribute detection. The output score of the segment-based phoneme classification is for the entire phoneme that is manifest in the center of the segment. And in doing this conversion, we make an assumption that the center "frame" of the segment corresponds with the target feature/phoneme. When using TDNN for detection, the TDNN is moved frame-by-frame from the first frame to the last frame of the sentence. The TDNN output is seen as the classification score for the class corresponding with the target phoneme at the center of the segment.

We denote the complete set of English phonemes as $S = \{a_i, i = 1,2,...,M\}$, where M is the total number of English phonemes. We assume that each phoneme can be uniquely represented by a set of N binary valued attributes $e_j, j = 1,...,N$. We see that each pair of attribute values, e_j^+ and e_j^-, can partition the entire set of phonemes into two subsets, S_j^+ (corresponding with e_j^+) and S_j^- (corresponding with e_j^-)

$$\begin{aligned} e_j^+ &\propto S_j^+ = \{a_{j,1}, a_{j,2}, ..., a_{j,M_j^+}\} \\ e_j^- &\propto S_j^- = \{a_{j,1}, a_{j,2}, ..., a_{j,M_j^-}\} \end{aligned} \tag{7-4}$$

where $S = S_j^+ \bigcup S_j^-$, for $j = 1,...,N$. M_j^+ is the total number of phonemes for the plus value of e_j and M_j^- is the total number of phonemes for the minus value of e_j. Thus

$$\begin{aligned}&P(e_j^+ \mid x) = P(S_j^+ \mid x),\\ &P(e_j^- \mid x) = P(S_j^- \mid x),\\ &j = 1,\ldots,N\end{aligned} \tag{7-5}$$

And given a parallel speech attribute e_j, we can calculate two posterior probabilities for its plus value and minus value

$$\begin{aligned}&P(e_j^+ \mid x) = P(S_j^+ \mid x) = \sum_{i=1}^{M_j^+} P(a_{j,i} \mid x) \quad \propto \quad \sum_{i=1}^{M_j^+} P(a_{j,i} \mid \underline{y}),\\ &P(e_j^- \mid x) = P(S_j^- \mid x) = \sum_{i=1}^{M_j^-} P(a_{j,i} \mid x) \quad \propto \quad \sum_{i=1}^{M_j^-} P(a_{j,i} \mid \underline{y})\\ &j = 1,\ldots,N\end{aligned} \tag{7-6}$$

where x is the center frame of segment $\underline{y}$.

In this way, we decompose the speech attribute e_j^+ and e_j^- into its phoneme constituents, and by summing up the TDNN detection scores for each of the phonemes in the plus attribute value phoneme subset S_j^+ and minus attribute value phoneme subset S_j^-, we can calculate the posterior probability of an attribute for a given frame $P(e_j \mid x)$ using TDNN.

7.2.4 Linearly combine the two posterior probabilities

We merge the frame-based results estimated via MLP methods and the segment-based results estimated via TDNN methods using the linear combination of the two sets of *a posteriori* probabilities, giving

$$\begin{aligned}&P(e_j \mid x) = \alpha_1 \cdot P_{MLP}(e_j \mid x) + \alpha_2 \cdot P_{TDNN}(e_j \mid x)\\ &\alpha_1 + \alpha_2 = 1, \qquad \alpha_1 \geq 0, \qquad \alpha_2 \geq 0.\end{aligned} \tag{7-7}$$

where $P_{MLP}(e_j \mid x)$ is the *a posteriori* probability of frame-wise attribute detection scores estimated using MLP for equation (7-1); $P_{TDNN}(e_j \mid x)$ is the frame-wise *a posteriori* probability of frame x residing in the center of segment $\underline{y}$, estimated from equation (7-6), and α_1 and α_2 are appropriate weights and should be optimized. In practice, we simply linearly interpolate the plus attribute detection score $P_{MLP}(e_j^+ \mid x)$ from MLP with $P_{TDNN}(e_j^+ \mid x)$ from TDNN to calculate $P(e_j^+ \mid x)$; and linearly interpolate the minus attribute detection score $P_{MLP}(e_j^- \mid x)$ with $P_{TDNN}(e_j^- \mid x)$ to calculate $P(e_j^- \mid x)$, thus get the final numerical result.

7.3 Experiments

Our experiments consist of 3 parts: the detection of the 14 parallel SPE features using MLPs, the classification of hierarchical speech features and phonemes using TDNNs, and finally the combination of the above two approaches to form a unified speech attribute detection system. All of our experiments were conducted on the TIMIT database. The training set was the TIMIT TRAIN set and the test set was the independent TIMIT TEST set. Both training and testing sets consist of all the sentences of the 8 dialect regions except for the SA sentences. Phoneme boundary frames and immediately adjacent boundary frames were excluded from both training and testing sets. We used a reduced phoneme alphabet consisting of 39 phonemes plus silence instead of the 61 TIMIT phones.

The SPE feature detection used single frame parameters of a set of 13 Mel-Frequency Cepstral Coefficients (MFCC). The hierarchical phoneme classification used variable length of segments, i.e. 15-frame segments for most classes, but 20-frame segments for the diphthong class due to the reason that diphthongs are generally much longer than other phonemes. The speech parameters for TDNN classification are 13 MFCCs and its first and second order deltas, for a vector size of 39 parameters per frame, and 585 parameters for most segments except for diphthong segments which have 780 parameters.

We used 10 msec Hamming windows for the calculation of the MFCC parameters, and the frame rate was 200 Hz. Adjacent frames were averaged, resulting in a 100 Hz frame rate.

7.3.1 Frame-based parallel speech feature detection

14 MLPs were trained to detect each of the 14 SPE features for the 40 phoneme alphabet using single frames of 13 MFCCs and a balanced training set. Due to memory limits and the high correlation between adjacent frames, we sampled one out of every four consecutive frames for training. Phoneme boundary frames and immediately adjacent boundary frames were excluded from both training and testing sets. We limited the training data size to 48,000 frames and tested on all the sentences in the independent TIMIT test set except for the SA sentences. Using balanced training data, the feature detection performance of the 14 SPE features for the 40-phoneme alphabet is illustrated in Table 7.1. It can be seen that 5 of the SPE features had detection performance of 94% or higher; 8 additional SPE features had detection performance of between 80 and 90% and only 1 SPE feature had detection performance below 80%.

Table 7.1 SPE feature detection using MLP (% correct)

SPE feature	MLP
vocalic	96.2
consonantal	89.7
high	80.9
back	76.8
low	82.0
anterior	89.8
coronal	88.0
round	86.3
tense	84.4
voice	94.2
continuant	81.7
nasal	97.5
strident	95.8
silence	97.3

7.3.2 Segment-based hierarchical speech feature classification

We used the TDNN toolbox developed in Chapter 5 for the classification of the features listed in Table 7.2. We used 15-frame segments consisting of 13 MFCC plus delta and delta delta parameters as input to train the TDNN for all phonemes, except for the diphthong classification in which the segment length was raised to 20 frames. Both training and testing data were pre-segmented according to the TIMIT hand labels. The center of the phoneme was placed at the center of the segment. The performance for each class is given in Table 7.2. Again we see that 3 hierarchical features had classification performance in the 90-100% range; 9 hierarchical features had classification performance in the 80-90% range and only 1 feature had classification performance below 80%.

Table 7.2 Hierarchical speech feature classification performance using TDNN (% correct)

Feature name	TDNN result
Top Class	96.7
V	84.4
Vowel-LH	81.4
Vowel-FB	80.0
Diphthong	93.5
Semivowel	90.2
Consonant	88.4
Nasal	79.7
Stop-place	89.6
Stop-voicing	85.5
Fric-place	89.3
Fric-voicing	87.6
Affricate	85.4

7.3.3 Convert hierarchical phoneme classification to parallel speech attribute detection

In this experiment, the TDNN was moved frame-by-frame to calculate frame-wise scores for the given utterance. The TDNNs were trained using segment-based methods, with the target phoneme designated at the center of the segment. Directly using the classification TDNNs for detection gave the results are listed in column 3 of Table 7.3. From this table, we can see that even though TDNN is not trained for all the tokens as it is moved frame-by-frame within a sentence, its average SPE feature detection performance (89.5%) is even a little better than that of MLP (88.6%). This result supports the idea that the TDNN is a powerful classifier for the SPE feature set.

7.3.4 Linear interpolation of MLP and TDNN detection results

For the results shown in Table 7.3 we obtained the SPE detection scores directly from MLP and indirectly from TDNN. Next we combined the two sets of scores using a weighted sum. The combination was optimized for each of the SPE features and the weights that gave the best classification performance were chosen. The performance results and the weights are listed in the "Combined" and "α_1, α_2" columns in Table 7.3. The performance of the MLP, TDNN and Combined methods are shown in Figure 7.2.

From the results, we can see that the overall detection performance scores were greatly improved by combining the MLP and TDNN classification results. The combined performance is much better than the MLP detection performance, with 3% absolute and 26% relative error reduction. The best improvement is for the "back" feature, which is 10% better than the original MLP performance. The combined performance is, on average, 2% better than the TDNN detection (18% relative error reduction). The best improvement for the TDNN detection is for the "consonantal" feature, with 6.4% absolute error reduction. In both cases, the best improvements happened when either TDNN or MLP detection performance was relatively poor.

Table 7.3 SPE feature detection performance (% correct)

SPE feature	**MLP**	**TDNN**	**Combined**	**α_1, α_2**
vocalic	96.2	94.9	96.8	0.7, 0.3
consonantal	89.7	84.7	91.1	0.7, 0.3
high	80.9	84.9	86.1	0.6, 0.4
back	76.8	86.8	86.9	0.4, 0.6
low	82.0	85.8	86.5	0.5, 0.5
anterior	89.8	88.3	92.0	0.6, 0.4
coronal	88.0	85.4	90.3	0.6, 0.4
round	86.3	90.4	90.9	0.5, 05
tense	84.4	85.5	87.1	0.6, 0.4
voice	94.2	91.7	94.8	0.6, 0.4
continuant	81.7	83.9	86.0	0.6, 0.4
nasal	97.5	97.0	98.1	0.6, 0.4
strident	95.8	95.9	96.8	0.6, 0.4
silence	97.3	96.8	98.4	0.6, 0.4
Average	**88.6**	**89.5**	**91.6**	**---------**

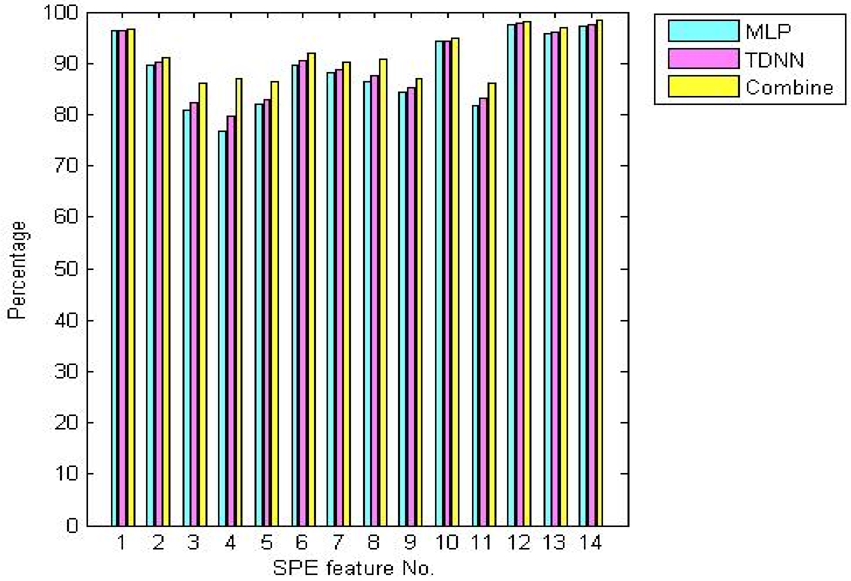


Figure 7.2 SPE feature Detection performances using MLP, TDNN and the combination method (performance of MLP/TDNN/ TDNN+MLP)

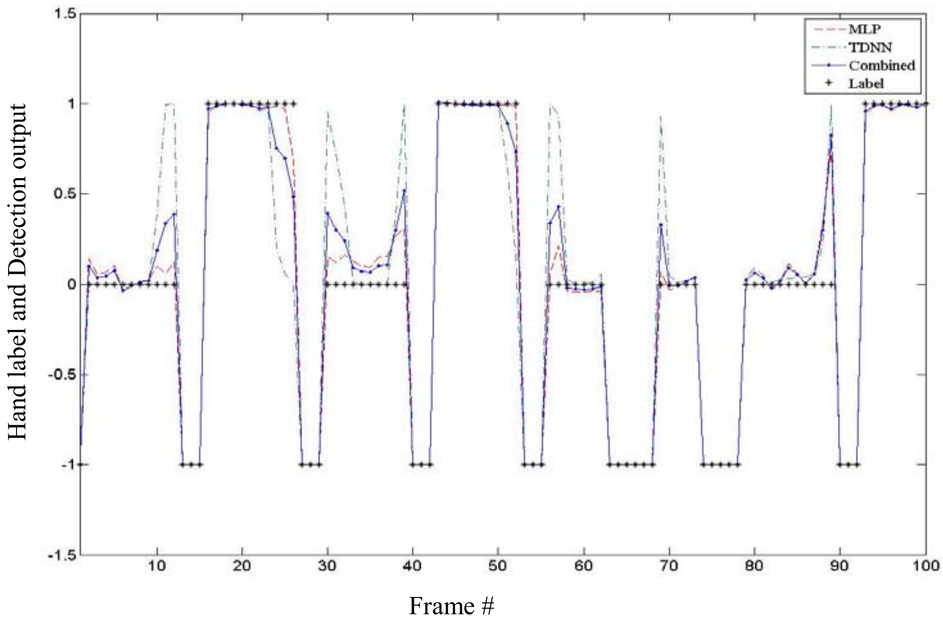


Figure 7.3 Example "Consonantal" feature detection using MLP/TDNN/Combined methods

Figure 7.3 shows an example of the "consonantal" feature detection on a speech segment in TIMIT using MLP, TDNN and the combined method. The TIMIT labels for that speech segment are also depicted in the figure. We can see that the combined method performed better than using MLP or TDNN alone, by generating fewer false alarms and fewer false rejects.

7.4 Discussion

We formulated the mathematical framework and experimented with a method for combining *a posteriori* probabilities from frame-based parallel feature detection and segment-based hierarchical phoneme classification to form a speech attribute detection system. All our experiments were conducted on the TIMIT database. For the segment-based system, TDNNs were trained using segments of various lengths for different classes, and the TDNNs outputs were converted to parallel SPE features detection probabilities. The TDNN detection probabilities were linearly combined with the MLP detection probabilities with weight optimization to provide improved classification performance scores for all the SPE attributes.

The results of phoneme detection using TDNN can be directly incorporated into the ASAT system using Conditional Random Field to improve the phoneme recognition accuracy, or it can be used to improve other speech attribute detection accuracy (as is shown in this chapter) which should then result in better overall phoneme recognition performance.

Although TDNN was trained using segments with the target phonemes at the center of the segment, it can directly be used in the speech feature detection problem and the performance is even better than the frame-based MLP results. In future studies, we can train the TDNN using all tokens within each sentence, and the performance can be expected to improve even further.

Chapter 8

Conclusions and Future Work

Knowledge-based and statistics-based approaches are two current directions in Automatic Speech Recognition (ASR). There have been several research efforts that tried to integrate the two approaches to improve the recognition performance and the Automatic Speech Attribute Transcription project is one of them. The ASAT system utilizes linguistically based speech attributes and speech events in an architecture that integrates knowledge sources, models, data, and tools, ultimately combining the results with state-of-the-art HMM systems. We found that when more knowledge sources are incorporated into the recognition system, the performance improved gradually.

In this thesis, we designed and optimized the front end processing of the ASAT system, in which various acoustic/phonetic speech features are incorporated with traditional acoustic measurements to improve the performance of the statistical automatic speech recognition system.

An automatic speech recognition system can use either detection or classification methods for segmenting and labeling the sounds within a spoken utterance using frame-based or segment-based methods. Frame-based speech attribute and phoneme detection approaches are more appropriate for characterizing static, short-time and unchanging properties of speech sounds, while segment-based phoneme classification methods can capture time-varying information that is necessary for dynamic phoneme detection and classification.

In this thesis, we investigated both frame-based parallel speech attribute detection and segment-based hierarchical speech attribute and phoneme classification. We integrated the two approaches in different ways to improve phoneme classification and speech attribute detection. The main contributions of this work are:

- The design of the overall front-end processing of the ASAT system (Chapter 3).
- Investigation of parallel speech detection with emphasis on the importance of balancing the training data for specific tasks (Chapter 4). We also compared different auditory models consisting of MFCC, PLP and RASTA-PLP, and found that different auditory models were of benefit to different speech features
- Investigation of hierarchical speech attribute and phoneme classification using Time-Delay Neural Networks (Chapter 5). We discussed key issues in the design and training of a TDNN and showed that by transforming each input parameter to the TDNN to be a zero mean, unit variance distribution (separately for each phoneme class) we could greatly improve the overall classification performance. We achieved state-of-the-art stop sound classification performance by using this method.
- Combination of frame-based parallel speech attribute detection and segment-based speech feature and phoneme classification to improve the phoneme classification performance (Chapter 6). We combined frame-based speech attribute detection and segment-based phoneme classification and then linearly interpolated the two sets of classification results. Most of the

phoneme classification scores showed better performance than using TDNN alone.

- Combination of frame-based parallel speech attribute detection and segment-based speech feature and phoneme classification to improve the parallel speech attribute detection performance (Chapter 7). We showed that a TDNN trained for segment classification can be directly used in the speech feature detection problem and the resulting performance is even better than the frame-based MLP results. The combined speech attribute detection system using both TDNN and MLP had 26% fewer errors than using MLP alone, and 18% fewer errors than using TDNN alone.

Future work should seek to find more compact and efficient speech attribute organization topologies, more appropriate speech parameters for each specific speech attribute detection, more powerful speech attribute classifiers and better combination methods in order to improve the phoneme recognition accuracy.

Speech parameters based on auditory models seem promising, since such parameters mimic the signal processing process inherent in human ears, and we expect them to provide a more accurate description of the properties of speech sounds.

Currently the classifiers we are using are Multi-Layer Perceptrons for speech feature detection and Time-Delay Neural Networks for feature and phoneme classification. TDNN input is scalable in a sense that the TDNN looks for the same speech event using duplicated weights within the segment. Instead of looking for the speech event within the segment, we can use the entire sentence as TDNN inputs, and the TDNN can be trained using variable lengths of input tokens.

The combination method being used is a simple linear interpolation in order to integrate detection or classification results from different speech attribute organizations. We hope to find better ways to combine results and further improve the phoneme recognition accuracy.

The Time-Delay Neural Network is a powerful classifier, and in the future work we can also use segment-based TDNN classification in the verification module in a speech recognition system.

Appendix A

Sound Pattern of English Feature Values for the TIMIT 61-Phoneme Alphabet

	Vocalic	Consonantal	High	Back	Low	Anterior	Coronal	Round	Tense	Voice	Continuant	Nasal	Strident	Silence
aa	1	0	0	1	1	0	0	0	1	1	1	0	0	0
ae	1	0	0	0	1	0	0	0	1	1	1	0	0	0
ah	1	0	0	1	0	0	0	0	0	1	1	0	0	0
ao	1	0	0	1	1	0	0	1	1	1	1	0	0	0
aw	1	0	0	1	1	0	0	1	1	1	1	0	0	0
ax	1	0	0	1	0	0	0	0	0	1	1	0	0	0
ax-h	1	0	0	1	0	0	0	0	0	1	1	0	0	0
axr	1	0	0	0	0	0	0	0	0	1	1	0	0	0
ay	1	0	0	1	1	0	0	0	1	1	1	0	0	0
b	0	1	0	0	0	1	0	0	0	1	0	0	0	0
bcl	0	1	0	0	0	1	0	0	0	0	0	0	0	0
ch	0	1	0	1	0	0	1	0	0	0	0	0	1	0
d	0	1	0	0	0	1	1	0	0	1	0	0	0	0
dcl	0	1	0	0	0	1	1	0	0	0	0	0	0	0
dh	0	1	0	0	0	1	1	0	0	1	1	0	0	0
dx	0	1	0	0	0	1	1	0	0	1	0	0	0	0
eh	1	0	0	0	0	0	0	0	0	1	1	0	0	0
el	1	1	0	0	0	1	1	0	0	1	1	0	0	0
em	0	1	0	0	0	1	0	0	0	1	0	1	0	0
en	0	1	0	0	0	1	1	0	0	1	0	1	0	0
eng	0	1	0	1	0	0	0	0	0	1	0	1	0	0
epi	0	0	0	0	0	0	0	0	0	0	0	0	0	1
er	1	0	0	0	0	0	0	0	0	1	1	0	0	0
ey	1	0	0	0	0	0	0	0	1	1	1	0	0	0
f	0	1	0	0	0	1	0	0	0	0	1	0	1	0
g	0	1	1	1	0	0	0	0	0	1	0	0	0	0
gcl	0	1	1	1	0	0	0	0	0	0	0	0	0	0
h#	0	0	0	0	0	0	0	0	0	0	0	0	0	1
hh	0	0	0	0	1	0	0	0	0	0	1	0	0	0
hv	0	0	0	0	1	0	0	0	0	1	1	0	0	0
ih	1	0	0	0	0	0	0	0	0	1	1	0	0	0
ix	1	0	1	0	0	0	0	0	0	1	1	0	0	0
iy	1	0	1	0	0	0	0	0	1	1	1	0	0	0
jh	0	1	1	0	0	0	1	0	0	1	0	0	1	0

	Vocalic	Consonantal	High	Back	Low	Anterior	Coronal	Round	Tense	Voice	Continuant	Nasal	Strident	Silence
k	0	1	1	1	0	0	0	0	0	0	0	0	0	0
kcl	0	1	1	1	0	0	0	0	0	0	0	0	0	0
l	1	1	1	0	0	1	1	0	0	1	1	0	0	0
m	0	1	0	0	0	1	0	0	0	1	0	1	0	0
n	0	1	0	0	0	1	1	0	0	1	0	1	0	0
ng	0	1	0	1	0	0	0	0	0	1	0	1	0	0
nx	0	1	1	0	0	1	1	0	0	1	0	1	0	0
ow	1	0	0	1	0	0	0	1	1	1	1	0	0	0
oy	1	0	0	1	0	0	0	0	1	1	1	0	0	0
p	0	1	1	0	0	1	0	0	0	0	0	0	0	0
pau	0	0	0	0	0	0	0	0	0	0	0	0	0	1
pcl	0	1	0	0	0	1	0	0	0	0	0	0	0	0
q	0	1	0	1	1	0	0	0	0	0	0	0	0	0
r	1	1	0	0	0	0	1	0	0	1	1	0	0	0
s	0	1	0	0	0	1	1	0	0	0	1	0	1	0
sh	0	1	1	0	0	0	1	0	0	0	1	0	1	0
t	0	1	0	0	0	1	1	0	0	0	0	0	0	0
tcl	0	1	0	0	0	1	1	0	0	0	0	0	0	0
th	0	1	0	0	0	1	1	0	0	0	1	0	0	0
uh	1	0	1	1	0	0	0	0	0	1	1	0	0	0
uw	1	0	1	1	0	0	0	1	1	1	1	0	0	0
ux	1	0	1	1	0	0	0	1	1	1	1	0	0	0
v	0	1	0	0	0	1	0	0	0	1	1	0	1	0
w	0	0	1	1	0	0	0	1	0	1	1	0	0	0
y	0	1	1	0	0	0	0	1	0	1	1	0	0	0
z	0	1	0	0	0	1	1	0	0	1	1	0	1	0
zh	0	1	1	0	0	0	1	0	0	1	1	0	1	0

References

[1] A. M. A. Ali, J. Van der Spiegel, P. Mueller, “Auditory-based speech processing based on the average localized synchrony detection”, *Proc. ICASSP’2000*, pp. 1623-1626, 2000.

[2] A. M. A. Ali, J. Van der Spiegel, P. Mueller, “Robust Auditory-based speech Processing Using the Average Localized Synchrony Detection”, *IEEE Transactions on Speech and Audio Processing, vol.* 10, no. 5, July 2002.

[3] A. M. A. Ali, J. Van der Spiegel, P. Mueller, “Acoustic-Phonetic Features for the Automatic Classification of Fricatives”, *J. Acoust. Soc. Am.*, Vol. 109, No. 5, pp.2217-2235, May 2001.

[4] A. M. A. Ali, J. Van der Spiegel, P. Mueller, “Acoustic-Phonetic Features for the Automatic Classification of Stop Consonants”, *IEEE Transactions on Speech and Audio Processing*, Vol. 9, No. 8, pp.833-841, Nov. 2001.

[5] A. M. A. Ali, J. Van der Spiegel, P. Mueller, G. Haentjens and J. Berman, “An acoustic-phonetic feature-based system for automatic phoneme recognition in continuous speech”, *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems*, 1999.

[6] B. S. Atal and L. R. Rabiner, “A Pattern Recognition Approach to Voiced-Unvoiced-Silence Classification with Applications in Speech Recognition,” *IEEE Transactions on Acoustics, Speech and Signal Processing,* Vol. ASSP-24, No.3, 1976.

[7] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, Boston, 1994.

[8] I. Bromberg, Q. Fu, J. Hou, etc., “Detection-Based ASR in the Automatic Speech Attribute Transcription Project”, in Proc. *Interspeech 2007*, Antwerp, Belgium.

[9] N. Chomsky and M. Halle, *The Sound Pattern of English*, MIT press, 1991.

[10] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[11] The Oregon Institute of Technology Center for Spoken Language Understandering (CSLU) toolkit, http://cslu.cse.ogi.edu/toolkit/index.html.

[12] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllable word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 357-366, 1980.

[13] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, Series B (Methodological) 39 (1): 1–38, 1977.

[14] S. Dusan, "Estimation of Speaker's Height and Vocal Tract Length from Speech Signal", *Proc. Interspeech-Eurospeech 2005,* Lisbon, Portugal, Sept. 2005.

[15] Sorin Dusan and Lawrence Rabiner, "On the Relation between Maximum Spectral Transition Positions and Phone Boundaries", *INTERSPEECH-ICSLP 2006* Pittsburgh, PA, USA, Sep. 17-21, 2006.

[16] E. Fosler-Lussier, "Tandem Acoustic Models with Gaussian Likelihood Spaces for ASR", *Proc. Interspeech 2006*, Pittsburg, PA.

[17] J. Frankel and S. King, "A Hybrid ANN/DBN Approach to Articulatory Feature Recognition", in *Proc. Eurospeech*, Lisbon, September 2005.

[18] T. Fukuda, O. Ichikawa and M. Nishimura, "Short- and Long-term Dynamic Features for Robust Speech Recognition", in *Proc. Interspeech 2008*, Brisbane, Australia.

[19] J. Glass, "A probabilistic framework for segment-based speech recognition", Computer Speech and Language 17: 137-152, 2003.

[20] P. Haffner, "Connectionist speech Recognition with a Global MMI Algorithm", in *Eurospeech'93*, Berlin, Germany, September 1993.

[21] P. Haffner, and A. Waibel, "Multi-state time-delay neural networks for continuous speech recognition", *Advances in Neural Information Processing Systems*, volume 4, pp. 579-588, Morgan Kaufmann, San Mateo, 1992.

[22] J. Harris, *English Sound Structure*, Blackwell, 1994.

[23] M. Hasegawa-Johnson, J. Baker, etc. "Landmark-based Speech Recognition: Report of the 2004 Johns Hopkins Summer Workshop", in Proc. *ICASSP 2005*, Philadelphia.

[24] H. Hermansky, "Perceptual linear predictive PLP analysis for speech", *Journal of the Acoustic Society of America*, 87(4):1738--1752, 1990.

[25] H. Hermansky, D. P. W. Ellis and S. Sharma, "Tandem Connectionist feature extraction for conventional HMM systems", *Proc. ICASSP 2000*, Beijing, China.

[26] H. Hermansky and N. Morgan, "RASTA processing of speech". *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, 1994.

[27] H. Hild and A. Waibel, "Multi-speaker/speaker-independent architectures for the multi-state time delay neural network", *Proc. ICASSP 1993*.

[28] H. G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluations of speech recognition systems under noisy conditions", in *Proc. ICSA ITRW ASR2000*, September 2000.

[29] H.-W. Hon, and K. Wang, "Unified Frame and Segment Based Models for Automatic Speech Recognition", in *Proceedings of ICASSP 2000*, v2, pp1017-1020.

[30] F. Hönig, G. Stemmer, C. Hacker and F. Brugnara, "Revising Perceptual Linear Prediction (PLP)", *Proc. INTERSPEECH*, 2005.

[31] "HTK Web-Site", http://htk.eng.cam.ac.uk.

[32] R. Jakobson and M. Halle, *Fundamentals of Language*, Walter de Gruyter, Jan 2002.

[33] A. Juneja, and C. Espy-Wilson, "Speech segmentation using probabilistic phonetic feature hierarchy and support vector machines", in the Proceedings of *International Joint Conference on Neural Networks*, Portland, Oregon, 2003.

[34] S. King, T. Stephenson, S. Isard, P. Taylor and A. Strachan, "Speech recognition via phonetically featured syllables", *Proc. ICSLP,* 1998.

[35] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks", *Computer Speech and Language* 14(4), pp. 333-353, 2000.

[36] K. Kirchhoff, "Combining Articulatory and Acoustic Information for Speech Recognition in Noisy Environments", *Proc. ICSLP* 1998, pp.891-894, Sydney, Australia, Dec.1998.

[37] K. Kirchhoff, G. A. Fink, and G. Sagerer, "Combining acoustic and articulatory feature information for robust speech recognition", *Speech Communication 37*, 2002, pp. 303-319.

[38] J. Lafferty, A. McCallum and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", *Proceedings of the*

Eighteenth International Conference on Machine Learning, p.282-289, June 28-July 01, 2001.

[39] C.-H. Lee, “From Decoding-Driven to Detection-Based Paradigms for Automatic Speech Recognition”, in Proc. *ICSLP 2004*, Korea.

[40] C.-H. Lee, M. Clements, S. Dusan, K. Johnson, B. Juang, E. Fosler-Lucier, and L. Rabiner, “An Overview on Automatic Speech Attribute Transcription (ASAT)”, in *Proc. Interspeech 2007*, Antwerp, Belgium.

[41] J. Li, C.-H. Lee, “A Study on Lattice Rescoring with Knowledge Scores for Automatic Speech Recognition”, *Proc. ICASSP 2005*.

[42] D.-T. Lin, J. E. Dayhoff, and P. A. Ligomenides, “Trajectory production with the adaptive time-delay neural network”, *Neural Networks*, vol. 8, no. 3, pp. 447–461, 1995.

[43] R. P. Lippmann, “Speech recognition by machines and humans”, *Speech Communication*, 22(1):1–15, 1997.

[44] K. Livescu, J. Glass, and J. Bilmes, “Hidden Feature Modeling for Speech Recognition Using Dynamic Bayesian Networks”, *Proc. EuroSpeech-2003*, Geneva, Switzerland, Sep. 2003.

[45] R. F. Lyon, “An Analog electronic Cochlea”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 7, 1988.

[46] N. Morgan, Q. Zhu, etc. “Pushing the Envelope – Aside”, *IEEE Signal Processing Magazine*, pp.81-88, September 2005.

[47] J. Morris and E. Fosler-Lussier, “Combining phonetic attributes using conditional random fields”, *Proc. InterSpeech 2006*, Toulouse, France.

[48] I. T. Nabney, *NETLAB: Algorithms for Pattern Recognition*, Springer, 2001.

[49] The NICO toolkit, http://www.speech.kth.se/NICO/.

[50] M. Ostendorf, V. Digalakis, and O. Kimball, “From HMM’s to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition”, IEEE Trans. Speech and Audio Proc., 4(5): 360-378, 1996.

[51] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[52] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, 1978.

[53] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. *IEEE*, Vol. 77, No. 2, pp. 257-286, Feb. 1989.

[54] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[55] M. D. Richard and R. P. Lippman, "Neural Network classifiers estimate Bayesian a posteriori probabilities", *Neural Computation*, vol.3, pp.461-483, 1991.

[56] S. K. Riis and Anders Krogh, "Hidden Neural Networks: A Framework for HMM/NN Hybrids", in *Proc. of ICASSP-97*, Munich, Germany, Apr. 1997.

[57] A. J. Robinson, "An application of recurrent nets to phone probability estimation", *IEEE Trans. Neural Networks*, Vol.5, No.2, March 1994, pp.298-305.

[58] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization For Spoken Word Recognition", *IEEE Trans. on Acoustics. Speech and Signal Processing*, 26(1), 43-49, Feb.1978.

[59] H. Sawai, "Frequency-Time-Shift-Invariant Time-Delay Neural Networks for Robust Continuous Speech Recognition", in *Proc. of ICASSP-91,* pp:45-48, 1991.

[60] S. Seneff, "A joint synchrony/mean-rate model of auditory speech processing", *Journal of Phonetics*, 16, pp. 55-76, 1988.

[61] M. Slaney and R. F. Lyon, "On the importance of time - a temporal representation of sound", In M. Cooke, S. Beet, and M. Crawford, editors, *Visual Representations of Speech Signals*, pages 95-116, John Wiley, 1993.

[62] M. Slaney, *Auditory Toolbox*, version 2, Technical Report # 1998-010, Interval Research Corporation, 1998.

[63] P. Somervuo, "Experiments with linear and nonlinear feature transformations in HMM based phone recognition", *Proc. ICASSP 2003*.

[64] P. Somervuo, B. Chen, and Q. Zhu, "Feature Transformations and Combinations for Improving ASR Performance", Proc Eurospeech 2003.

[65] T. Stephenson, H. Bourlard, S. Bengio, and A. Morris, "Automatic speech recognition using dynamic Bayesian networks with both acoustic and articulatory variables," *IDIAP, Tech. Rep.* 00-19, 2000.

[66] K. N. Stevens, *Acoustic Phonetics*, MIT Press, 1998.

[67] K. N. Stevens, “Toward a model for lexical access based on acoustic landmarks and distinctive features”, *J.Acoust. Soc. Am.*, 111(4), April, 2002, pp1872-1891.

[68] A. Suchato, “Classification of Stop Consonant Place of Articulation”, Ph.D. Thesis, MIT, 2004.

[69] G. G. Tajchman and N. Intrator, “Phonetic classification of TIMIT segments preprocessed with Lyon’s cochlear model using a supervised/unsupervised hybrid neural network”, *Proc. ICSLP 1992*.

[70] A. Waibel, T. Hanazawa, etc., “Phoneme Recognition Using Time-Delay Neural Networks”, *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol.37, No.3, March 1989.

[71] A. Waibel, H. Sawai, and K. Shikano, “Modularity and Scaling in Large Phonemic Neural Networks”, *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol.37, No.12, December 1989.

[72] S. Young. “A Review of Large-vocabulary Continuous-speech Recognition,” *IEEE Signal Processing Magazine*, 13:5, Sept. 1996, pp.45-57.

[73] Y. Zhang, Q. Diao, S. Huang, W. Hu, C. Bartels, and J. Bilmes, “DBN based multi-stream models for speech”, In Proc. ICASSP 2003, Hong Kong, China, April 2003.

[74] S. Y. Zhao and N. Morgan, “Multi-Stream Spectro-Temporal Features for Robust Speech Recognition”, in Proc. Interspeech 2008, Brisbane, Australia.

[75] Y. Zheng, M. Hasegawa-Johnson, and S. Borys, “Stop Consonant Classification by Dynamic Formant Trajectory”, in Proc. Interspeech 2004, pp.2181-2184.

[76] G. Zweig, Speech Recognition Using Dynamic Bayesian Networks, Ph.D. thesis, University of California, Berkeley, 1998.

Curriculum Vitae

Jun Hou

9/1993 – 7/1998	Tsinghua University, Beijing, P. R. China. Bachelor of Engineering in Electronics Engineering
9/1998 – 7/2001	Tsinghua Universitiy, Beijing, P. R. China Master of Engineering in Electronics Engineering
9/2001 – 10/2009	Rutgers, The State University of New Jersey, Department of Electrical and Computer Engineering, New Brunswick, New Jersey. Doctor of Philosophy in Electrical and Computer Engineering.

Publications

1. J. Hou, L. Rabiner and S. Dusan, "Parallel and Hierarchical Speech Feature Classification Using Frame and Segment-Based Methods", in *Proceedings of Interspeech 2008*, Sep. 22 – 26, 2008, Brisbane, Australia.
2. J. Hou, L. Rabiner and S. Dusan, "On the Use of Time-Delay Neural Networks for Highly Accurate Classification of Stop Consonants", in *Proceedings of Interspeech 2007*, Aug. 27 – 31, 2007, Antwerp, Belgium.
3. I. Bromberg, Q. Fu, J. Hou, J. Li, C. Ma, B. Matthews, A. Moreno-Daniel, J. Morris, S. M. Siniscalchi, Y. Tsao, Y. Wang. "Detection-Based ASR in the Automatic Speech Attribute Transcription Project", in *Proceedings of Interspeech 2007*, Aug. 27 – 31, 2007, Antwerp, Belgium.
4. J. Hou, L. Rabiner and S. Dusan. "Automatic Speech Attribute Transcription (ASAT) – the Front End Processor", in *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 14 - 19, 2006.
5. C. D. Correa, A. Agudelo, A. M. Krebs, I. Marsic, J. Hou, A. Morde, and S. K. Ganapathy, "The Parallel Worlds System for Collaboration among Virtual and Augmented Reality Users". Demo at *the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, Arlington, VA, Nov. 2 - 5, 2004.
6. A. Morde, J. Hou, S. K. Ganapathy, C. Correa, A. Krebs, L. R. Rabiner, "Collaboration in Parallel Worlds", in *the Sixth International Conference on Multimodal Interfaces*, ICMI'04. Penn State University, State College, PA, October 14 - 15, 2004.