A New Approach to Voice Dialing

Neena Jain
B.Tech., Indian Institute of Technology, Bombay, India, 1992

A thesis submitted to the faculty of the
Oregon Graduate Institute of Science & Technology
in partial fulfillment of the
requirements for the degree
Master of Science
in
Electrical Engineering

July 1995

The thesis "A New Approach to Voice Dialing" by Neena Jain has been examined and approved by the following Examination Committee:

Ronald A. Cole
Professor
Thesis Research Advisor

Etienne Barnard
Assistant Professor

Hynek Hermansky
Associate Professor

Dedication

To my parents

Acknowledgements

First and foremost, I would like to express deep gratitude to my thesis advisor Dr. Ron Cole for his guidance, encouragement and support. I also wish to thank Dr. Etienne Barnard for his unswerving patience and many valuable discussions. Thanks are also due to Dr. Hynek Hermansky for his encouraging feedback as a member of my thesis committee.

Furthermore, I would like to thank my colleagues Jacques, Lodewyk and Johan for their help in converting this thesis into a working system.

Priya, Zhihong, Kay, Carlos, Sarel and lots of other friends have made my stay at OGI a memorable one. I have also benefited greatly from the OGI Staff, in particular Kathy Feyer, Barbara Olsen, Terri Durham and Vince Weatherill.

Finally, I would like to express my appreciation for Anurag for his moral support through all of this.

Contents

List of Tables

List of Figures

Abstract

A New Approach to Voice Dialing

Neena Jain, M.S.
Oregon Graduate Institute of Science & Technology, 1995

Supervising Professor: Ronald A. Cole

Automatic Voice Dialing (AVD) is a rapidly growing service being offered by the telecommunications industry. AVD systems have tremendous commercial applications since they recognise phrases spoken by a user, and dial the corresponding phone number stored in the user's database. In this thesis, a new approach to AVD has been presented and investigated. The two main components of this system are speaker verification and speaker dependent speech recognition. A novel approach to speaker dependent speech recognition has been presented, which uses dramatically less storage to represent a speaker's words, with minimal degradation in recognition accuracy. In this approach, the symbolic string produced by the output of the speech recognizer is used for representing a phrase. For speaker verification, two designs, one based on the distinguishing characteristics of phonemes, and the other based on averaging of temporal spectra, have been thoroughly investigated.

Compared to conventional techniques, the new method of performing speaker specific speech recognition lead to a reduction of about 1:300 in the storage requirements with comparable recognition accuracy of 97%. Using averaging of temporal spectra for speaker verification lead to a reduction of about 1:30 with verification rate of 93%. A complete

working voice dialing system that incorporates this new technology has been developed. This system performs both speaker verification and speaker specific speech recognition for voice dialing.

Chapter 1

Introduction

Speech recognition systems are rapidly moving from the laboratory to the consumer market. During the past decade, a variety of speech-based systems have been successfully deployed in the market. Since the telephone is the most handy mode of communication, many new services are becoming available in the telephone network, and speech is the natural way to control these services. Examples include voice dialing, voice messaging and control of service options (e.g. cancel call waiting). These services generally require access to databases, computer networks or some protected resources. Hence security is becoming an important issue. For example, one would desire to have security in banking over the telephone. Voice dialing is a perfect example of a telephone-based service. The automatic voice dialing system recognizes the phrase spoken by the user and dials the number registered in the user's database corresponding to the phrase uttered. In this thesis, a new approach to voice-dialing has been presented and investigated.

1.1 Goal

There are two main goals of this thesis. Firstly, to demonstrate the feasibility of a new approach to speaker dependent speech recognition that uses dramatically less storage than current approaches to represent a speaker's words, but does not produce a significant degradation in recognition accuracy. Secondly, to develop and demonstrate a complete working voice dialing system that incorporates this new technology. The system will perform both speaker verification and speaker dependent speech recognition for voice dialing. Speaker verification refers to verifying the identity of the speaker based on his or

her voice.

1.2 Motivation

Voice dialing systems are attractive and convenient to use because:

- spoken communication is invariably preferred over push-button (DTMF) input.
- phone numbers do not have to be remembered.
- push-button input is more prone to error.
- spoken communication will be significantly safer in cellular phones in cars, in comparison with push-button input, which requires looking away from the road (USA today recently reported that cars with cellular phones have 34% more accidents).
- voice dialing systems are more secure than phone cards.

1.3 Existing Systems

Voice dialing is a commercial application. Nynex, Sprint and Cellular One are few of the companies which offer this service. Since this is a commercial application, there is very little literature available on the algorithms used in these voice dialing systems.

Noguchi [1] has described a voice dialing system for PC based platforms. This system requires entries in the database in the form of Kana characters. A speaker independent speech recognizer is used for recognizing the speech. The main drawback of this system is that it can work only for the Japanese language wherein the pronunciation can be obtained directly from the Kana characters.

Geller [2] has described a voice dialing system for the car environment. The main focus of this system was to suppress noise in vehicles. It is based on whole-word Hidden Markov Models (HMM). The main drawback of this system is that it needs a significant amount of training data and has large storage requirements. This system requires about 1280 floating point numbers to represent each word.

1.4 The Approach

Traditional speaker dependent recognition systems are based on parameters derived from the acoustic signal [3]. The approach presented in this thesis uses the symbolic string produced by a speech recognizer to represent the word or phrase. The user decides the set of words or phrases to be used. The models for these words or phrases, henceforth referred to as "labels", are automatically generated from the output of a speaker independent phonetic recognizer. These models are subsequently used for recognizing speech. The proposed algorithm can achieve accuracy up to 97%. Compared to conventional techniques, the storage requirements reduce by a factor of 300 with comparable recognition accuracy.

Two designs with low storage requirements are proposed for performing speaker verification. The first design depends on the ability of phonemes to distinguish between speakers. The second design is based on sequential averaging of LPC coefficients, and can achieve verification rates of 93% with storage requirements of 40 floating point numbers.

1.5 Outline of thesis

This thesis is arranged as follows. Chapter 2 presents an overview of the proposed system. In chapter 3, a new way of performing speaker dependent speech recognition is presented and the performance of the proposed algorithm is evaluated. In chapter 4, two designs for speaker verification are presented and the research performed to evaluate them is presented. The conclusions and future research directions are discussed in chapter 5.

Chapter 2

System Overview

2.1 Introduction

A number of considerations motivated the design of the proposed system. In this chapter, we will first describe the set of desirable features of the proposed voice dialing system. This is followed by a detailed description of the design of the system.

2.2 Desirable Features

The following are the essential features of a practical voice dialing system.

User Interface: A new user should be able to enroll very easily and quickly. Hence the training period required should be small. For the system to respond immediately, all the algorithms have to be real time. In addition, it should be easy to edit the existing database. In case of an error, the system should have a mechanism to exit gracefully.

Security: Adequate security is important to avoid fraudulent usage of the voice dialing service. Security can be provided by performing speaker verification.

Algorithms: The algorithms should be designed such that they have low storage requirements and low computational complexity . Furthermore, the system should be able to measure the confidence of the output of the system. The algorithms should be robust to changing environmental conditions, different telephone channels and to variations within each speaker.

2.3 Design

The proposed system can be divided into different components based upon either their functionality, or the type of speech technology employed. In the following sections, the system is described in terms of these two factors.

2.3.1 Functionality

On the basis of functionality, the system consists of three phases, namely, registration, application and adaptation.

- **Phase 1: Registration/Enrollment**

  Registration is the first phase of the system wherein the user completes a registration form with a list of phone numbers and labels (*ie.* names or phrases), each associated with a phone number (see Appendix A). At this phase, a password has to be selected in order to prevent fraudulent phone calls.

  After selecting the "speed labels", associated numbers, and password, the user calls up the system and records a password, followed by the list of phone numbers and labels associated with each number. This forms the user's database. Figure 2.1 shows the basic structure of the registration process. The principal aim of this phase is to collect the data needed to develop speaker-specific models for speaker verification and speech recognition. The entire protocol is presented in Appendix B.

- **Phase 2: Application/Usage**

  Figure 2.2 shows the basic structure of the application process. This phase deals with the actual usage of the system, which involves verification of the user, recognition of speech, and dialing of the phone number associated with the label recognized. The models built in the registration process are used for performing the above actions. The user can also edit the database to add or delete speed labels and phone numbers. The entire protocol is presented in Appendix C.

- **Phase 3: Adaptation**

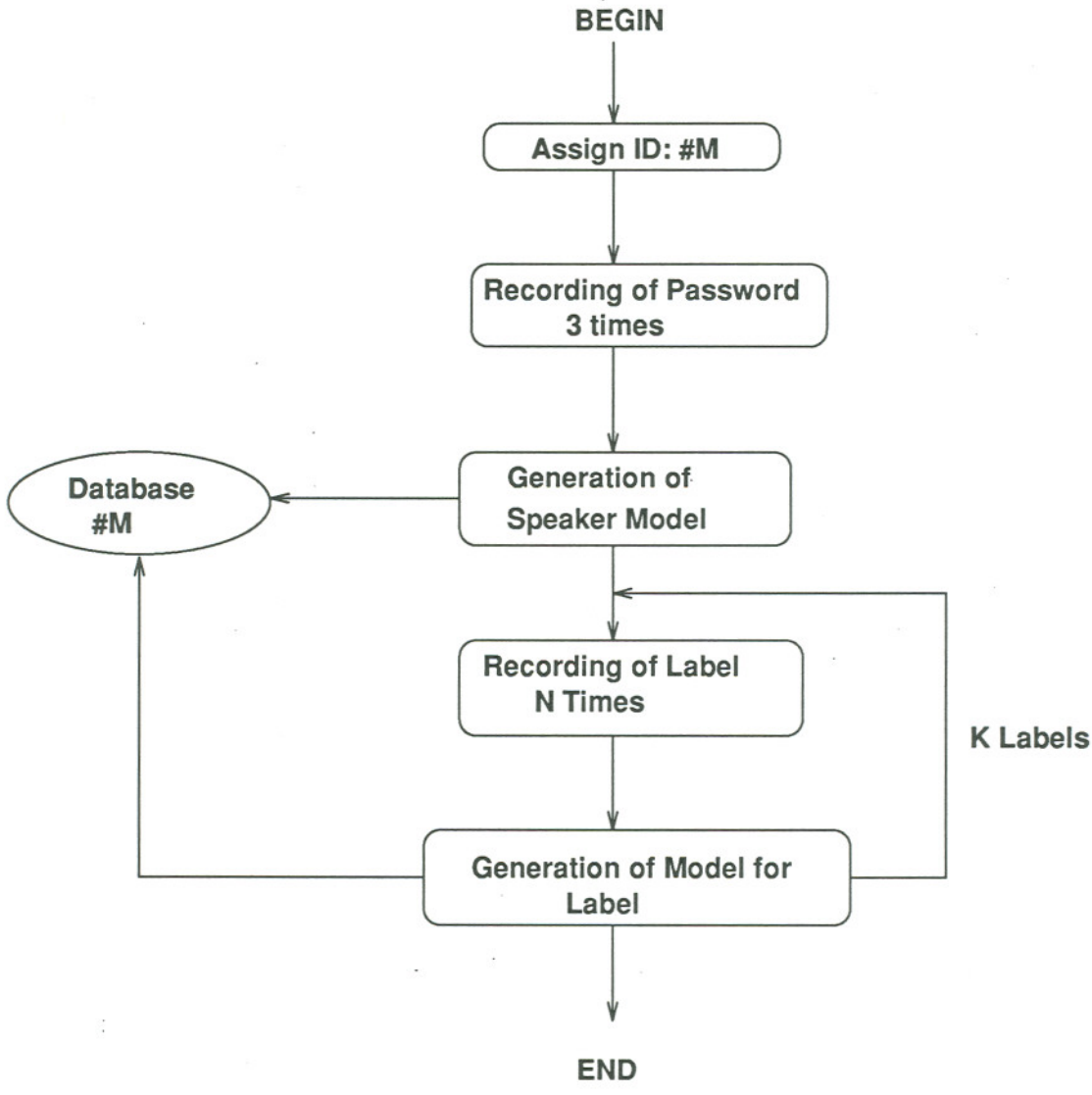

Figure 2.1: Registration Phase

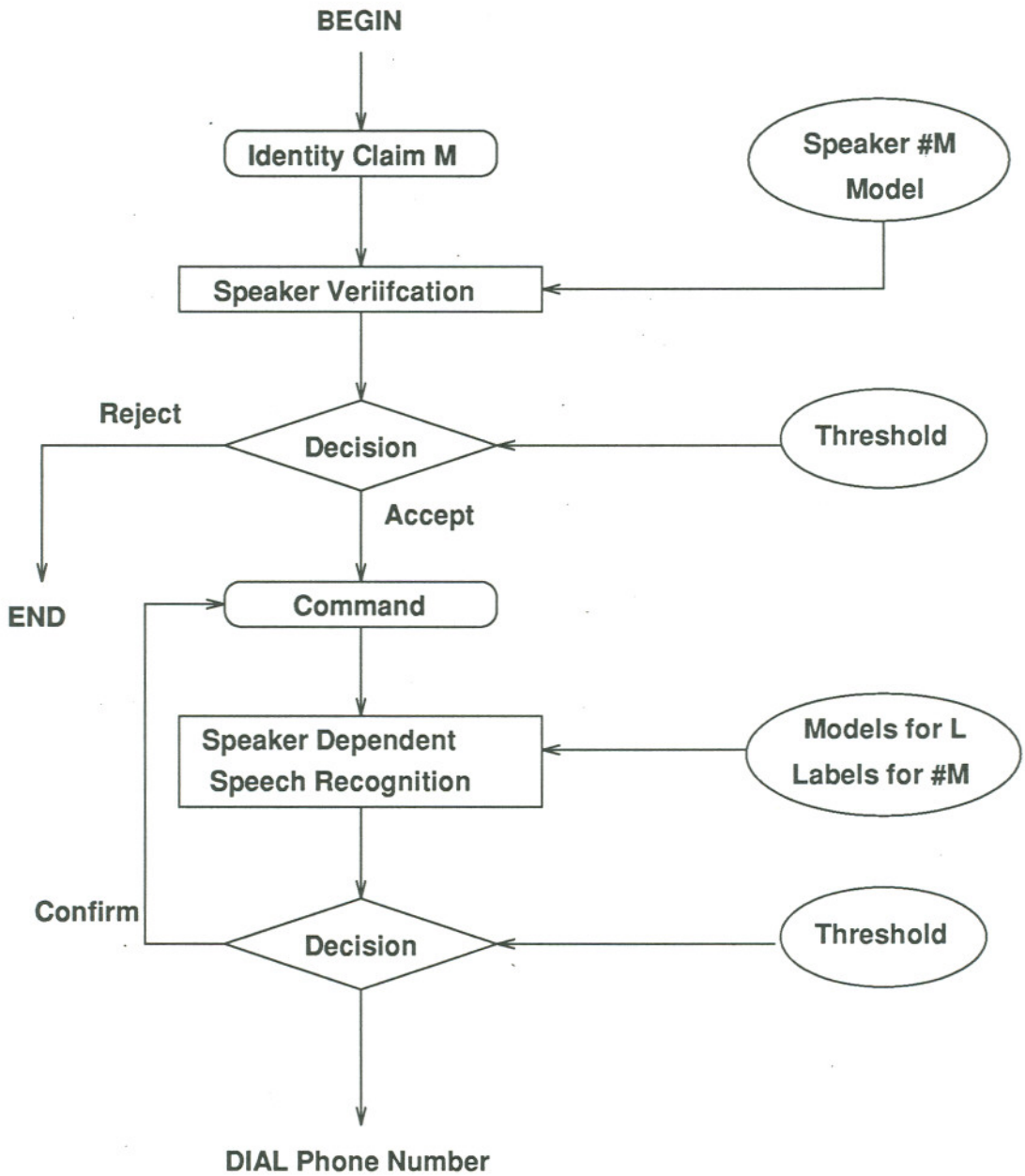

Figure 2.2: Application Phase

The speech of an individual may change depending upon the time of the day, the emotional or physical state of the user, and a number of other factors. Adaptation to the speech of the user is a desirable feature for accommodating this kind of intra-speaker variability. This can be accomplished only after the actual usage of the proposed system. Repeated usage of the system helps in providing examples of the password, and other labels, which may be used for modifying the previously stored models. Therefore, the models eventually become more robust to changing environmental and channel conditions. At this point, this phase has not yet been implemented since the system is currently in the process of deployment to the telephone network.

2.3.2 Technology

On the basis of technology employed, the two important modules are speech recognition and speaker verification.

- **Speech Recognition:**

  This is the most critical component of this system. There are two types of speech recognition systems used, namely, speaker independent and speaker dependent. The speaker independent recognizers are used for fixed vocabulary recognition. During registration, speaker independent recognizers are used for yes-no type recognition to confirm answers and for digit recognition of phone numbers associated with the labels. These recognizers were developed at CSLU [4] and a detailed description is beyond the scope of this thesis.

  A Speaker dependent system uses the speaker's voice to recognize speech. Typically, this process involves creating word models based on acoustic parameters of the speaker's words. Our system introduces a new twist: During registration, a speaker independent phoneme classifier is used to generate models for the user defined labels. These models are a string of phonemes representing speech. During application, these models are used along with the phoneme classifier to recognize the spoken label. We call these models "speaker specific" models and not "speaker

dependent" as they are based on speaker independent classification. In this manner, a very concise representation of all the labels is achieved, since relatively few sound categories or phonemes are needed to represent the words of a language. The research performed for speaker specific recognition will be discussed in detail in the following chapter.

- **Speaker Verification:**

  Speaker verification is the task of deciding whether the unlabelled speech belongs to a specific person or not. This is a binary decision which requires that the user declare his or her identity to the system using a spoken password. Since the voice dialing system accesses an individual's database and dials out, it is important to prevent fraudulent use of this service. Speaker verification offers a unique way of performing personal identification verification (PIV) [5] over the telephone network because:

  1. it is based on a specific user's voice which is suitable for a telephone based system.
  2. it can co-exist with speech recognition systems.
  3. user preference is higher than other biometric verificators.
  4. different levels of security can be easily achieved through different dialogues with the user (e.g., multiple passwords for applications requiring more security).

  In this system, a text dependent approach has been used for speaker verification. This implies that the verification is performed on the basis of fixed text; *ie.*, the same word or phrase is used each time. Each user chooses his or her password. During registration, a template consisting of 20th order LPC coefficients is obtained from the password. During application, the unlabelled speech is matched with the template of the speaker. Based on the measure of similarity and a predefined threshold, a decision is made about the authenticity of the caller. The experiments performed for speaker verification will be discussed in detail in chapter 4.

Chapter 3

Speaker-Specific Discrete Utterance Recognition

3.1 Motivation

This chapter provides the rationale for research on speaker-specific isolated word or phrase recognition, and describes the experiments that resulted in the proposed system. The goal of the research is twofold

1. To produce a system with good recognition accuracy and
2. To produce a system with low storage requirements.

In a system used by millions of people storage becomes a critical issue; the more parameters needed to represent word models, the more storage is required. Storage is expensive, both because more disk space is needed, and because more data must be accessed quickly. Small vocabulary speaker dependent systems [3] typically use about 12 LPC coefficients for every 10 msec of speech. Hence for one second of speech, 19.2 K bits are needed (assuming 16 bits are needed to represent each floating point number). When multiplied by the number of words and the number of people, a great deal of storage is needed.

This chapter describes a novel approach to derive speaker specific templates from the speech produced by each speaker. Instead of representing templates as acoustically derived parameters, we represent them as strings of phonemes. A speaker independent phonetic classifier is used to generate these speaker-specific templates. Note that for this approach to work, all that is needed is that the phonetic recognizer perform in a consistent way

for the same speaker for the same word. It does not matter if *home* (/h/ /ow/ /m/) is consistently recognized as *fawn* (/f/ /ao/ /n/); all that matters is that /f/ /ao/ /n/ is produced by the phonetic recognizer each time that *home* is produced.

In this chapter, the speaker specific models will be described first. The different experiments performed with these models will then be described.

3.2 Speaker-Specific Models

The vocabulary of the voice dialing system is defined by the user during registration. A novel approach was used for obtaining the word models [1]. In this approach, word models or templates were created for a label by using the phonetic recognizer. The goal of **speaker specific template generation** is to produce a "good" phonetic representation of the label used for voice dialing. The following steps produce a phonetic transcription.

1. **Endpoint Detection**

   The speech signal is recorded using a standard telephone line and sampled at 8 kHz. The first step in the digital processing is endpoint detection to determine the time at which speech begins and ends. Endpoint detection can lead to errors when there are clicks on the lines or heavy breathing. Endpoint detection helps in removing the unnecessary silence before and after the label.

2. **Feature Extraction**

   The next step is to generate features for each 6 msec time frame. This is done by extracting PLP [6] coefficients from each frame of the chopped waveform. Seventh order PLP coefficients along with logarithm of energy are computed for each frame. The analysis window size is 10 msec. Two features corresponding to voicing are also computed. Thus a total of 10 features are computed for each frame of the speech signal. To account for the influence of context on the current frame, PLP features from a window of 156 msec centered on the current frame are computed. Ten features are computed from the frame to be classified and averaged over each

[1] I thank Ron Cole for describing this notion

of the following regions before and after the frame to be classified: 6 to 18 msec, 36 to 48 msec and 72 to 84 msec. Figure 3.1 shows the distribution of frames used for computing these features. A total of 70 features are used for representing each frame.

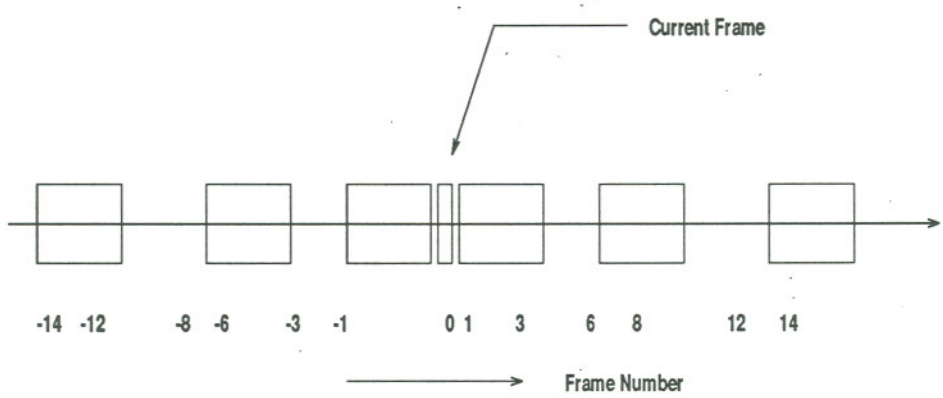

Figure 3.1: Frames used for Extraction of Features

The two features that estimate voicing for each frame are provided by a separate three-layer neural network trained on voiced and voiceless speech frames. Although the voicing classifier is trained with the same PLP features, experiments have shown that including these features improves classification performance.

3. **Frame Classification**

It has been established by many researchers that artificial neural networks are particularly good for pattern classification. The multi-layer Perceptron (MLP) is a feed forward network with one or more hidden layers. A weight is attached to each connection between the neurons. A three layered MLP trained with the conjugate-gradient optimization algorithm [7] was used to assign phoneme scores to each individual time frame. The OGI stories database [8] was used for training this MLP. A total of 39 phonemes were used for representing speech. The exact details of the training and the architecture are described in [9]. The 70 features corresponding to each frame are given as input to the MLP and output values for each phoneme are recorded.

These values represent the probability of the frame being in a certain phonetic state [10].

4. **Search**

A Viterbi search is used for generating the phonetic string for each spoken word or phrase. The Viterbi search maps frame scores to phonemes. It is constrained in two ways - bigram grammar and duration. A general English bigram grammar is used in this system. This grammar consists of the probabilities of each of the 39 phonemes being followed by each other. These transition values were computed from the large phonetically labelled OGI stories database [8]. The minimum and maximum durations of each phoneme are also used for constraining the search. The Viterbi search algorithm uses the grammar and the duration constraints to find the maximum scoring phonetic sequence given the frame-by-frame output scores. The score of any sequence is the sum of the log probabilities from the neural network for the matching phoneme and the transition probabilities for the transition from one phoneme to another.

The Viterbi search provides an efficient procedure for obtaining a non-linear time alignment between all possible phoneme sequences specified by the grammar and the output of the MLP. The input speech is assigned the phoneme sequence with the highest scoring path.

5. **Automatic post-processing of string**

The post-processing was done to eliminate some of the highly unlikely sequences of phonemes, particularly in the beginning and the end of the utterance. Clicks on the telephone line and heavy breathing can cause errors in end-point detection. This leads to very unnatural phoneme sequences. These are eliminated by using some simple rules. For example, the word *home* led to a sequence like "cl k cl hh ow m" due to a click in the beginning of the utterance. The "k" can be eliminated by use of a simple rule that no consonants can occur between silence in the beginning or end of the utterance. Hence the string becomes "hh ow m". Another sequence obtained was "hh ow m cl s cl s cl s cl s cl" due to heavy breathing at the end of the utterance.

This can also be eliminated by the same rule. However, the errors due to the neural network cannot be eliminated by post-processing.

In this manner, a phonetic representation of each utterance is obtained. The research described next addresses the question of how to select the "best" template from the N repetitions of each label to achieve high recognition accuracy.

3.3 Experiments

3.3.1 Database

The database used for evaluating the performance of this system consisted of 30 speakers saying a predetermined list of 21 phrases 6 times over the telephone. Each phrase consisted of 1-3 words. They are listed in Table 3.1. The average length of each utterance was 1 second. The speech was sampled at 8kHz.

Table 3.1: List of phrases in the database

	Label		Label		Label
1.	call home	8.	the office	15.	erase
2.	call mom	9.	dan	16.	forward
3.	call david	10.	sandra	17.	note to myself
4.	call my secretary	11.	the boss	18.	emergency
5.	call susan	12.	steve	19.	help
6.	call damien	13.	diane	20.	call
7.	call martina	14.	anthony	21.	add

3.3.2 Choosing a template

Given N phoneme strings representing the same label for a particular speaker, how do we obtain the string which best represents the label? One possibility is to select one or more of the phoneme strings obtained to represent the label. Another method is to combine the phoneme strings by using some kind of averaging technique.

The nature of the problem can be shown by the three strings generated for the label *call_home*. These strings are as follows.

/ k aa l hh ow n/

/ ah n cl k aa hh aa m/

/ dh ah cl k aa l hh aa l m/

Looking at the variability in the phoneme string, it is difficult to pick the best one merely on the basis of the static string. Due to the variability, it was not feasible to use an averaging technique to find a string which best represents all the three.

It was decided to use the Viterbi score as the decision criterion. The Viterbi score represents the closeness of the best match obtained during the search. There are many ways in which the templates can be chosen.

- The most simple method is to use the string with the maximum Viterbi score, since a higher score indicates a better match between the phoneme scores and the phoneme sequence.
- Another method is to force-align each phoneme string with the other waveforms and compute the average Viterbi score. The template with the maximum average can be selected.
- More than one string can be chosen to represent the label to capture intra-speaker variability.
- The number of training samples needed is another important factor which may affect the recognition accuracy of the system.

The following experiments were performed to determine the effect of these various parameters on the recognition accuracy of the proposed system.

Effect of force-alignment

This experiment was performed to evaluate the performance of the system when the template is selected by using force-alignment as compared with random selection to represent the labels. The training set consists of 3 repetitions of each label and the test set consists of the remaining 3 repetitions. To see the effect of random selection, any one of the three phoneme strings representing each label is selected at random to form the word-models.

These strings form the dictionary which is used during recognition of the test set. The recognition rate obtained was **87.3 %**. To see the effect of force-alignment, each one of the 3 phoneme strings is matched with other 3 waveforms and the string with the maximum average score is selected. These strings form the dictionary which is used during recognition of the test set. The recognition rate obtained was **90.1 %**. Thus an increase of 3% is observed as one moves from random selection of templates to selection using force-alignment strategy. This represents a decrease in the error rate of **22 %**. The random selection experiment forms the baseline experiment to evaluate the strategies used for choosing and rejecting templates.

Effect of number of templates selected

The number of templates needed for representing each label is another factor which may affect the overall recognition performance of the system. Multiple templates should help in capturing the intra-speaker variability in speech. Since human beings are generally quite consistent with respect to themselves in how they pronounce words, too many templates should not be required. The training set consists of 4 repetitions of each label and the test set consists of the remaining 2 repetitions. Given 4 templates, they can be arranged in a descending order of representibility by using the force-alignment technique. The system was tested with 4 different dictionaries. The dictionaries were obtained by picking the top 1, 2, 3 and all 4 templates arranged in descending order according to the average Viterbi score. Table 3.2 shows the change in the error rate with the increase in the number of templates selected to represent each label. It was observed that the error rate decreased as the number of templates increased from 1 to 3 but increased slightly when all the 4 templates were used. This confirms our earlier notion that we do not need too many templates to represent each label.

Table 3.2: Effect of Number of Templates Selected

Number of Templates	1	2	3	4
Error Rate (in %)	9.6	7.2	6.5	6.8

Effect of size of training data

This experiment was performed to see the effect of the size of the training data *ie.* the number of repetitions available during training on the recognition accuracy of the system. The training set consists of the 1 - 4 repetitions of each label and the test set consists of the remaining 2. One template is chosen from 1, 2, 3 and 4 training templates respectively and the system is evaluated. Table 3.3 shows the change in error rate with the increase in the training data size.

Table 3.3: Effect of Size of Training Data

Number of Training templates	1	2	3	4
Error Rate	11.6	10.2	9.9	8.5

3.3.3 Rejecting a template

Since the front-end is not perfect, it is very important to realize its limited capabilities and deal with it. Many different levels of rejection and checks can be used.

1. The phoneme string representing a utterance may be rejected during training.
2. An entire label may be rejected during training.
3. The output of the recognition system may be rejected during testing.

Each one of these rejections and their impact on the performance of the system will now be described.

Rejection within a label

The Viterbi score with which the phoneme string is generated can be considered as a measure of how well the phoneme string matches with the acoustics according to the neural network. It can be used to reject those strings which have a very low score and hence indicate that they are a poor match with the waveform. The rejection is performed by using a fixed threshold. This threshold is very low so that it rejects only the extreme cases where the match is very bad. For example, "s aa l eh r ey cl d ah" is obtained instead

of "f aor w er cl d". Figure 3.2 shows the histogram of Viterbi scores obtained during the generation of the template. The threshold used for rejection in this system was 0.4 . Considering the first 4 utterances of 30 speakers saying 21 words, this led to rejection of 33 utterances. Since there are multiple utterances for each label, none of the labels were rejected entirely. This is the first step in rejecting a template.

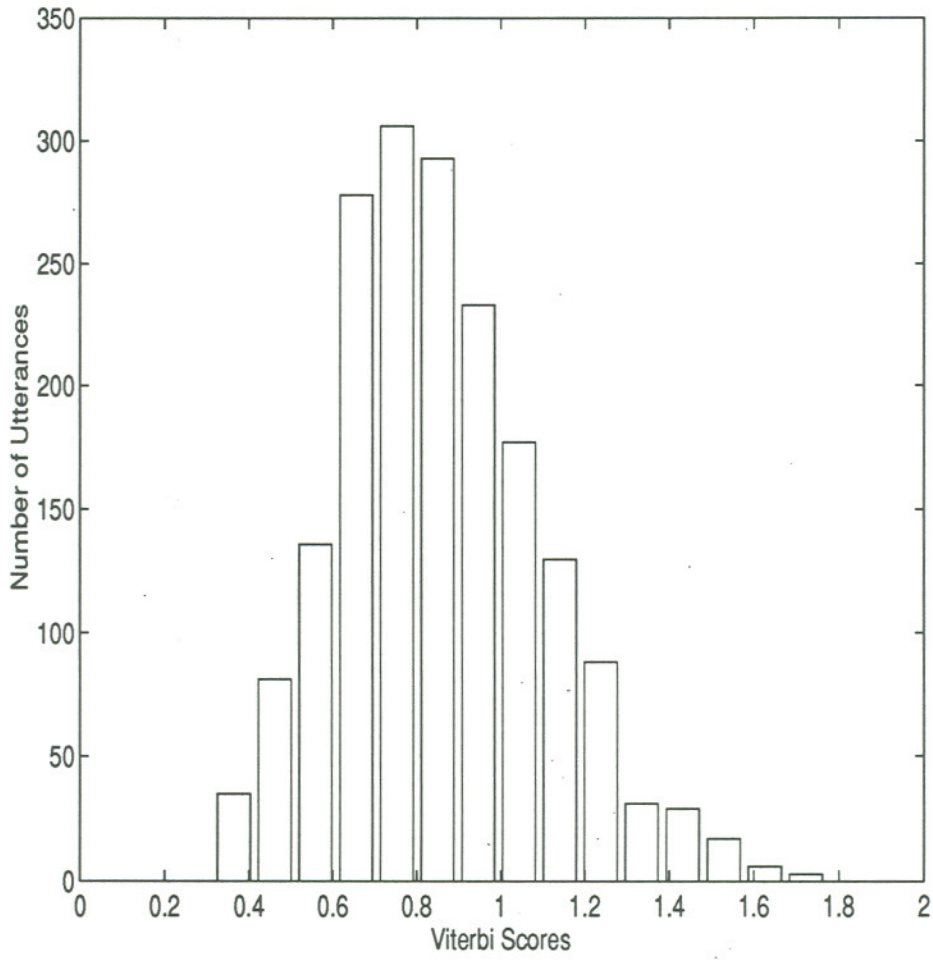

Figure 3.2: Histogram of Viterbi Score during Generation of Template

After the initial rejection within a label, the best template is chosen by doing a cross-match with the other remaining templates of the same label. The template with the

maximum average Viterbi score is selected. A low average score indicates that the phoneme string does not fit the other waveforms very well. Thus, a threshold can be used to reject the label if the best fit has a low score. The system in which one template is selected from three training samples, is chosen to test the different rejection thresholds. It was found that the threshold of 0.6 is adequate. This lead to rejection of about 1.5 % of the labels With this rejection, the performance of the system increased from 90.1 % to 90.7 % Most of the templates which were rejected had noise and clicks in the background.

Rejection of confusable labels

Since some labels may have similar acoustic and phonetic content they have the potential of getting confused by the system. For example, Dan and Diane. The system has been designed to eliminate such possibilities. It has the option of asking the user to substitute certain labels. The effect of this design choice is now tested.

After the best templates representing each label have been chosen, they are cross-checked with all the other labels. This is done by force-aligning the selected template to the waveforms corresponding to other labels and recording the Viterbi score obtained. A high Viterbi score obtained during force-aligning the template of label A with the waveform of label B, indicates potential for confusion during the application of the system. Hence either label A or B must be rejected or substituted.

Two kinds of thresholds can be used to decide if two labels are close to each other - absolute and relative thresholds. The use of an absolute threshold leads to rejection of a label if the Viterbi score obtained during force-alignment is higher than the absolute threshold. The use of a relative threshold leads to the rejection of a label if the difference in the score obtained during force-alignment with its own waveform and with that of the other label is less than the relative threshold. Both these thresholds can also be used together. Figure 3.3 shows the distribution of Viterbi scores when the phonetic string representing a label is force-aligned with itself and with other labels. As the thresholds, both relative and absolute, are increased, the recognition rate increases since the number of confusable case decreases but the rejection rate also increases. A compromise between the rejection rate and the recognition rate has to be reached for optimum performance of

the system. The system was tested with different combinations of relative and absolute thresholds. In the proposed system, it was decided to use an absolute threshold of 0.7. In the system in which one template is selected from three training samples, the recognition accuracy increased from 90.1% to 93.3% with this absolute threshold of 0.7, while 10% of the labels were rejected.

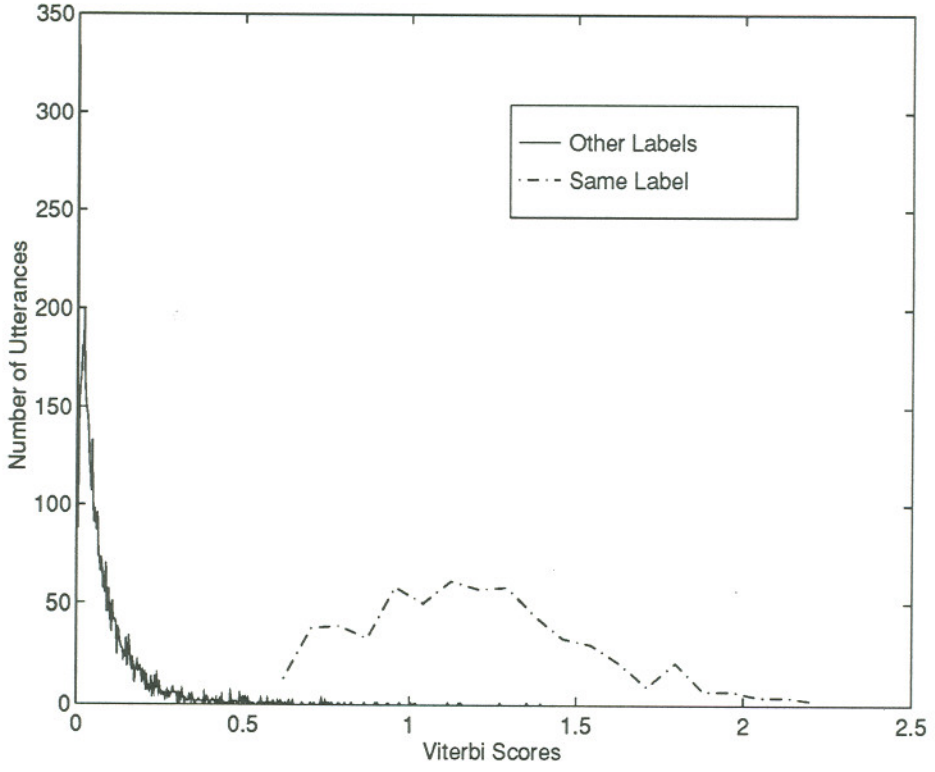

Figure 3.3: Distribution of Viterbi Scores during Cross-Check

Rejection during testing

A low Viterbi score during recognition indicates a low confidence in the output of the system. Rejection of these low confidence outputs will help in increasing the recognition rate. It is better to ask the user to repeat the utterance than to make a mistake and dial the wrong number. The Viterbi scores obtained during testing of the system in which

one template is chosen from three training examples are shown in Figure 3.4. Figure 3.4 shows distribution of Viterbi scores obtained for the correctly recognized utterances and the incorrectly recognized utterances. As the threshold for rejection is increased, the number of errors decreases but the number of correctly recognized utterances will also increase. Hence the reject rate increases with the increase in recognition rate and an optimum balance has to be found. In the proposed system, it was decided to use a fixed threshold of 0.5 to reject the output of the recognizer. This lead to rejection of about 15% of the utterances.

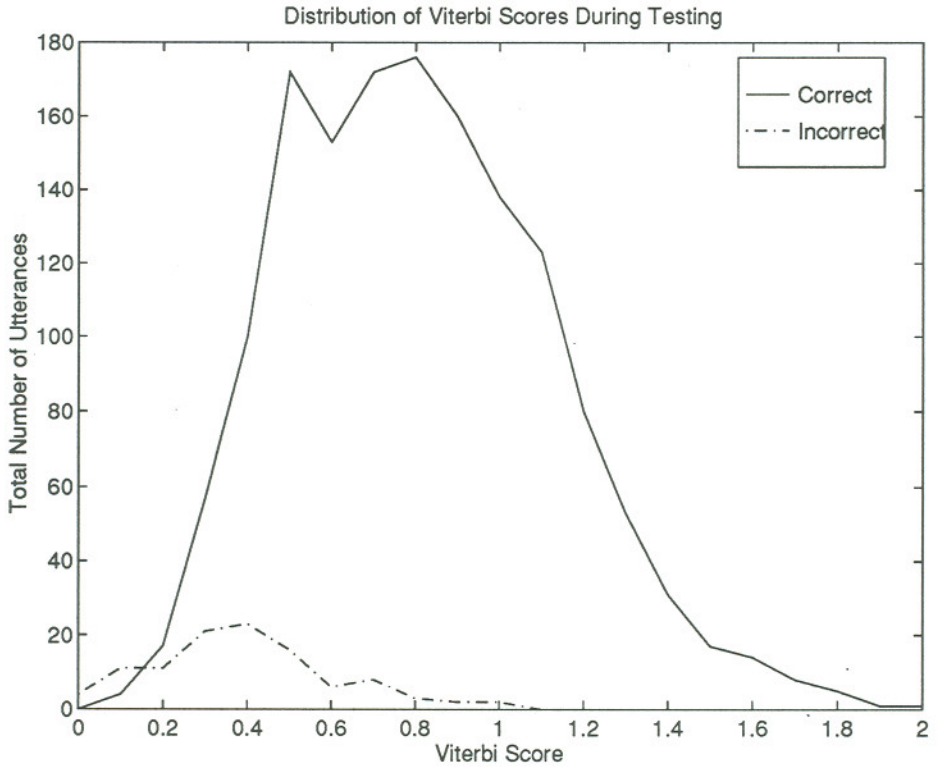

Figure 3.4: Distribution of Viterbi Scores during Testing

3.4 Comparison with DTW

Dynamic Time Warping (DTW) with LPC features can be considered as one of the state-of-the art classification systems for speaker dependent recognition [3]. DTW is a very efficient method of achieving non-linear normalization of the time axis of the speech signal [11]. This non-linear normalization is important because of the fluctuations in the speaking rate within a speaker. 2 systems based on DTW with LPC coefficients were developed to observe the trade-offs in recognition accuracy with number of parameters used.

In the first system, 20th order LPC cepstral coefficients computed for every 10 msec of speech were used to build the reference templates for each label. During testing, the unknown speech signal was aligned using DTW with all the reference templates. The reference template with the best match *ie.* the smallest distance was declared as the output of the recognition system.

In the second system, 20th order LPC cepstral coefficients were computed for every 10 msec of speech. These were then quantized by using Vector Quantization [12]. The codebook used for quantizing was generated by using the OGI stories database [8]. 2 different codebooks were used for quantizing the lower and the upper 10 LPC coefficients. The size of each of the codebooks was 512. The quantized vectors were used to build the reference templates for each label. During testing, the unknown speech vector was quantized using the same codebook. This was then matched with all the reference templates using DTW. The label with the best match was declared as the output of the recognition system.

Both these systems were evaluated for 4 different training conditions - selecting a random template, selecting the best of three training templates as the reference template, using rejection during training, and using rejection during training and testing. These performances are compared with the MLP based approach with exactly the same training conditions. Figure 3.5 shows the error rates for all the 3 approaches for the 4 training conditions. It can be observed that the direct DTW approach always does better than the other 2 approaches, The gap between all of them decreases with increase in the reject rate.

The direct DTW approach with LPC coefficients requires 2000 floating point numbers

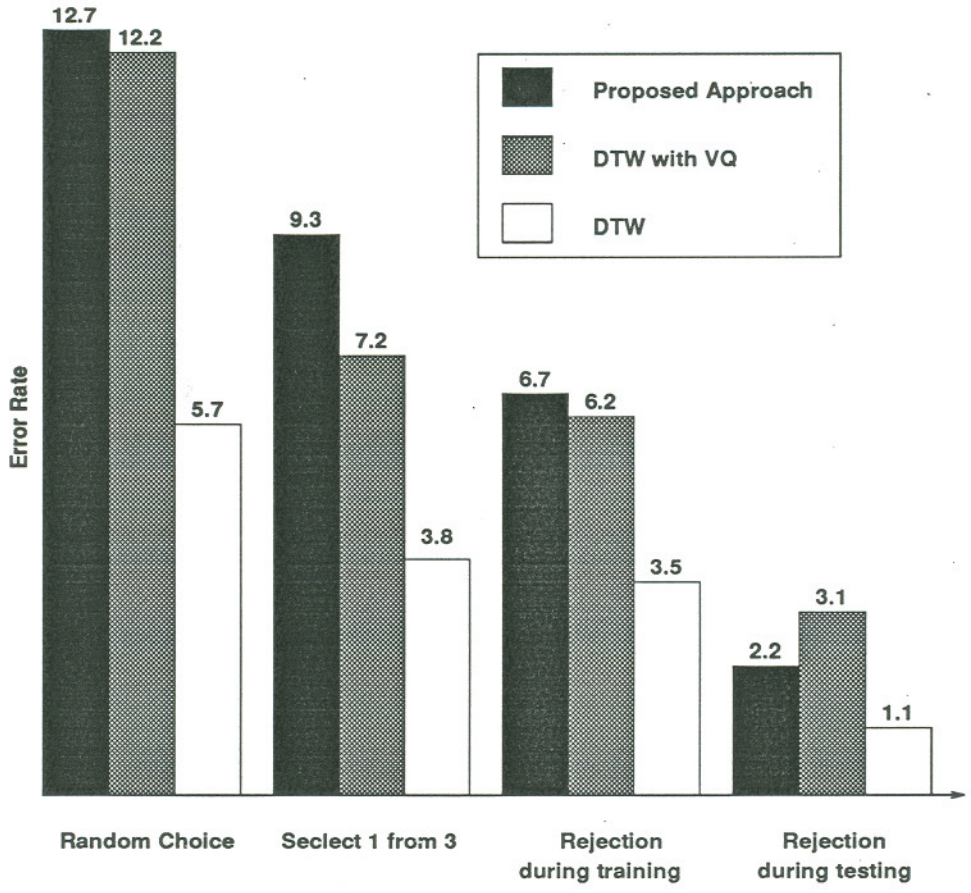

Figure 3.5: Comparison of DTW Approach with Proposed Approach

to represent 1 second of speech. Assuming that 16 bits are required to represent each LPC coefficient, 32 K bits are required for storing 1 second of speech. The VQ based approach with DTW requires 1800 bits to store 1 second of speech since 18 bits are needed to represent 10 msec of speech. The MLP based approach proposed in this thesis, represents 1 second of speech by using 10 phonemes on average. Since there are a total of 39 phonemes, 6 bits are needed for representing each phoneme. Hence on an average, 60 bits are required for storing the speaker-specific models. There is a significant difference between the storage requirements of the direct DTW system compared to the MLP based system, about 1:500, but a very small difference between the error rates. The difference between the storage requirements of the VQ based approach and the proposed approach is about 1:30 but very little difference in the error rates. It can concluded that the proposed

system has met the goal of low storage requirements for representing the user's speech with very little degradation in the recognition accuracy.

3.5 Error Analysis

A detailed analysis of the errors produced by the system was performed. Confusion matrices of two parameter choices are presented to show the effect of the different techniques used for selection and rejection of templates. The first system is the baseline system in which one template is selected at random and no rejection is performed. In the second system, one template is selected from 3 training samples. The threshold used for self rejection is 0.4. The absolute threshold used for cross-checking with other labels is 0.7. No rejection is performed during testing in both the systems. The test data consists of 3 repetitions of all the 21 labels.

In the first system, recognition rate obtained was 87.3%. Figure 3.7 shows that the vast majority of errors occur within classes with high acoustical and phonetic similarity. 28 errors occur due to confusion between *Dan* and *Diane*. 18 errors occur due to confusion between *call* and *call_home*. 49 errors occur due to confusion between *Dan*, *Diane* and *add*.

In the second system, recognition rate obtained was 93.3%. There were no obvious trends in the errors. Comparison between Figure 3.7 and Figure 3.8 show that many of errors present in the first system have been eliminated.

It was observed that most of the errors were due to :

- acoustically similar labels *eg*. *Dan* and *Diane*.
- errors in end-point detection.
- noise in the line and background.
- bad phoneme string templates.

There were also a few errors which could not be attributed to any of these reasons.

It was observed that the errors were unevenly distributed across labels. The distribution of performance across different speakers has been shown in Figure 3.6. Speaker 30 and

13 report very low recognition rates. Examination of the speech files belonging to these speakers indicate a lot of background noise and unusual clicks occurring very frequently in the telephone channel. The speech belonging to speaker 30 was barely recognizable.

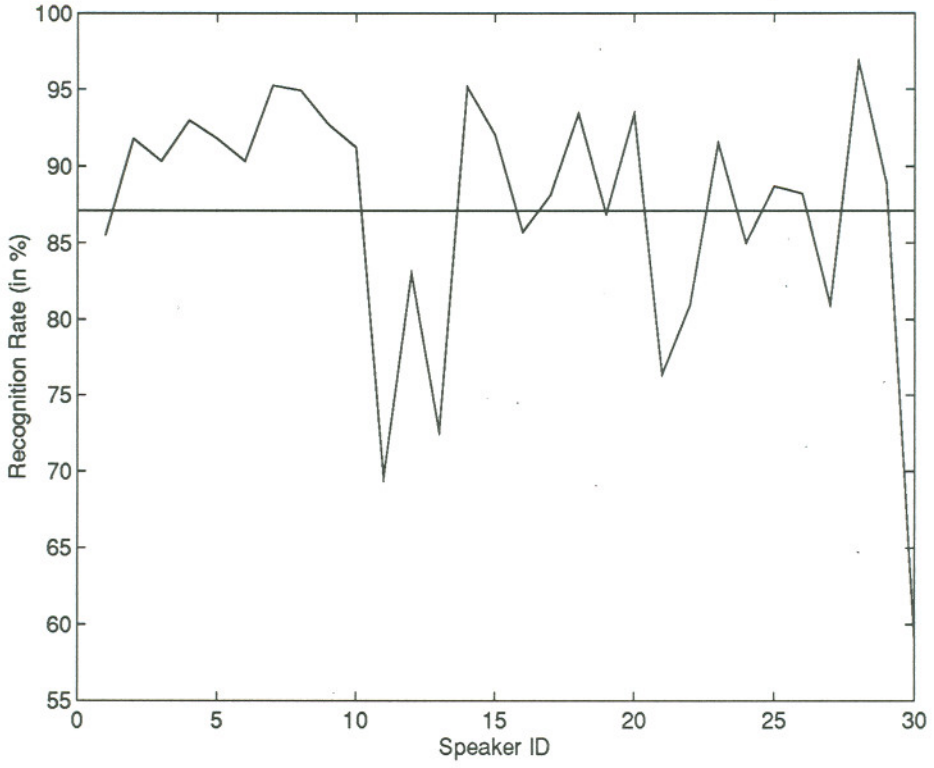

Figure 3.6: Variation in Recognition Rates Across Speakers

3.6 Conclusions

The aim of the current research was to obtain a low storage discrete word recognizer with high recognition accuracy. Based on the experiments performed, the following conclusions can be drawn.

1. The proposed system has very low storage requirements. Approximately 60 bits to represent an one second long label.

2. Using all the different rejection criteria, recognition accuracy of about 97 % can be obtained which is comparable to other existing speaker dependent discrete word

OUTPUT OF THE RECOGNITION SYSTEM 1

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
call_home	1	74	8	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	4	.
call_mom	2	5	80	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
call_david	3	.	2	76	.	3	4	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.
call_my_secretary	4	.	.	.	85	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
call_susan	5	.	.	.	1	87	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
call_damien	6	.	1	5	.	.	70	1	.	.	.	.	.	.	1	.	.	.	.	.	.	.
call_martina	7	.	1	2	.	.	.	87	.	.	.	.	.	.	.	.	.	.	.	.	.	.
the_office	8	.	.	.	.	1	.	.	78	.	1	2	2	.	.	2	.	.	.	.	.	.
dan	9	.	1	.	.	.	2	.	.	61	4	.	1	13	.	.	.	.	.	.	.	5
sandra	10	.	2	1	.	.	.	.	.	.	79	2	.	1	1	.	1	.	.	.	.	1
the_boss	11	.	2	1	.	.	.	.	4	.	.	78	.	.	.	.	.	.	.	.	1	.
steve	12	.	.	2	.	2	2	.	.	.	.	.	67	1	5	5	.	.	.	.	.	.
diane	13	.	.	.	.	.	1	.	.	15	1	.	1	59	.	.	.	.	.	1	.	3
anthony	14	.	.	1	.	1	.	.	.	.	.	.	.	.	77	.	.	.	.	.	.	2
erase	15	.	.	1	.	.	.	.	1	1	1	.	1	.	2	74	.	.	.	.	.	.
forward	16	2	.	1	.	.	.	.	.	.	.	.	.	.	.	.	82	.	.	1	.	.
note_to_myself	17	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	87	.	.	.	.
emergency	18	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.	83	.	.	.
help	19	2	.	.	.	.	.	.	.	.	3	.	.	.	.	.	.	.	.	63	12	4
call	20	18	3	.	.	1	.	.	1	.	.	.	.	.	.	.	.	.	.	5	61	.
add	21	.	1	.	.	.	.	.	.	8	4	.	2	5	.	.	3	.	.	4	.	54

Recognition Rate = 87.3% errors= 228, total= 1790

Figure 3.7: Confusion Matrix For System 1 (Experiment with random selection of 1 template with no rejection)

OUTPUT OF THE RECOGNITION SYSTEM 2

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
call_home	1	84	2	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.
call_mom	2	2	52	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
call_david	3	.	2	83	1	.	2	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.
call_my_secretary	4	.	.	2	85	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.
call_susan	5	.	.	1	1	66	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
call_damien	6	.	1	4	.	1	33	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
call_martina	7	1	.	1	4	.	.	84	.	.	.	.	.	.	.	.	.	.	.	.	.	.
the_office	8	.	.	.	.	.	.	.	85	.	.	2	1	.	.	.	.	.	1	.	.	.
dan	9	.	1	.	.	.	1	.	.	80	.	.	3	4	.	.	.	.	.	.	.	1
sandra	10	.	.	.	.	.	.	.	.	.	80	.	.	1	.	.	.	.	.	.	.	1
the_boss	11	1	1	1	.	.	.	.	1	.	.	82	.	.	.	.	.	.	.	.	.	.
steve	12	.	.	1	.	.	.	1	1	.	.	.	78	.	1	1	.	.	.	.	.	.
diane	13	.	1	1	.	.	.	.	.	3	.	.	.	24	1	.	.	.	.	.	.	1
anthony	14	.	.	1	.	.	.	.	.	3	.	.	.	.	74	.	.	.	.	.	.	4
erase	15	.	.	.	.	.	.	.	.	1	.	.	.	.	.	78	.	.	1	.	.	.
forward	16	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	87	.	.	.	.	.
note_to_myself	17	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	86	1	.	.	.
emergency	18	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	84	.	.	.
help	19	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	81	.	.
call	20	8	3	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	3	38	.
add	21	1	.	.	1	.	.	.	.	6	1	.	.	2	1	.	3	.	.	.	.	38

Recognition Rate = 93.3% errors= 107, total= 1589

Figure 3.8: Confusion Matrix For System 2 (Experiment with selection of 1 from 3 templates with cross-rejection)

recognizers.

3. The training data required consists of three repetitions of each label. On a Dec Alpha, model 240, running OSF1, the amount of time needed for generating one speaker model is about 0.2 seconds. Hence the proposed algorithm can be considered to be real time.

4. In the proposed system, non-English labels can also be used. Although the phonetic recognizer is trained with English speech and has English phonemes as output, it can work with non-English labels as long as the recognizer consistently maps the non-English phonemes to English phonemes. An informal experiment was performed wherein 10 labels in Hindi were registered. A recognition accuracy of 100% was obtained during testing. Only one user participated in this experiment.

5. The proposed system can be used by non-native speakers of English as well native speakers.

6. The speaker-specific labels can easily be integrated with speaker independent labels. For example, the phoneme string representing *home*, a speaker-specific label, can be combined with the phoneme string representing *dial*, a speaker independent word, to recognize *dial_home*. This is advantageous as it is not necessary to ask the user to register function words like "call", "dial" etc.

Chapter 4

Speaker Verification

4.1 Introduction

Speaker verification is the task of accepting or rejecting the identity claim of a speaker. This decision is made on the basis of individual information included in the speech waves. A speaker verification system can perform correctly in two ways– by accepting the true speaker, and by rejecting an imposter. It can also make errors in two ways; by rejecting the true speaker, or by accepting an imposter. Performance of a speaker verification system is typically measured in terms of equal error rate for these two error types. **Equal error rate** can be defined as the error wherein the percentage of false acceptance is equal to that of false rejection. This provides a single number to measure the performance of an algorithm.

There are several sources of variability in speech which make this task difficult. There is inter-session variability due to the telephone, the environment, and also the speaker. Variability in telephone could arise due to the telephone channel, the line noise, and also the hand set. Variability could also stem from differences in recording and transmission conditions and also from noise in the environment. Furthermore, the variability may be caused by the speakers due to emotion, mood, physical health (cold), and speaking rate. Hence, the speaker verification algorithm has to be resistant to all these sources of variations.

This chapter describes the research performed for speaker verification and describes the experiments that led to the proposed system. The goal of this research was threefold.

- To build a speaker verification algorithm with low storage requirements.

- To build a speaker verification algorithm with low equal error rate.
- To build a speaker verification algorithm with a small training period.

In this chapter, we will describe the structure of the basic speaker verification system. We will then propose two designs with low storage requirements. The first design is based on the ability of phonemes to distinguish between speakers. The second design is based on sequential averaging of spectra. Each approach will be followed by the experiments performed to evaluate the system design and the results obtained.

4.2 Basic System

4.2.1 Typical Structure

Figure 4.1 shows the basic structure of a speaker verification systems (SVS). During training, the reference model is generated by extracting some parameters from the user's speech. During application, an identity claim is made by an unknown speaker. An utterance of the unknown speaker is matched with the reference model of the claimed speaker. If the match is above a certain threshold, the identity claim is verified; otherwise it is rejected.

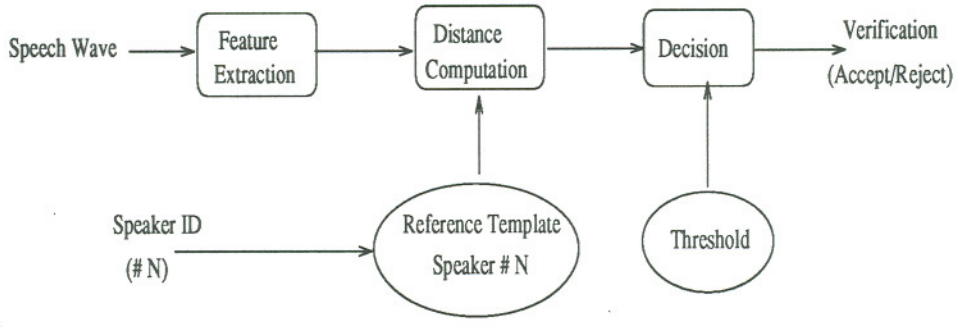

Figure 4.1: Basic structure of Speaker Verification System

SVS can be categorized by the type of speech material used for validating an identity claim. **Fixed-text** systems require the recitation of a predetermined text. **Free-text** systems accept speech utterances of unrestricted text. The performance of fixed-text

systems is generally higher than that of free-text systems as reliable comparisons can be made between the test and the reference utterances with adequate time alignment. Furthermore, free-text systems generally require longer speech signals for training and for verification.

In a service like voice-dialing, the verification has to be performed quickly and hence a fixed-text system is more desirable. Smaller training period is another advantage of a fixed-text system. In this thesis, only fixed-text systems have been investigated.

4.2.2 Existing Verification Systems

There are a variety of speaker verification systems reported in literature with very low equal error rates. References [13], [5] and [14] provide detailed reviews of the existing speaker verification systems. The methodologies used for speaker verification can be summarized as follows.

- Different types of features that have been used are : LPC, delta LPC, PLP, RASTA PLP and log spectrum.
- Different pattern matching techniques that have been used are : DTW, VQ, HMM, Neural Nets and combinations of these.
- Different decision making strategies that have been used are : Fixed threshold, relative threshold, multiple level of thresholding and cohort normalization of scores.

A quantitative comparison of the performance of these algorithms is not simple since the results have been obtained on significantly different databases and with different decision strategies. All these systems have been optimized for low equal error rates, but the storage requirements for each speaker's model are, in general, quite large. There is a need for as concise a representation of the speaker as possible with comparable low error rates. The following sections will describe two designs which lead to concise representations for speaker verification.

4.3 Phoneme Based Method

4.3.1 Approach

Our belief is that there is a unique signature in each person's speech. Each human being has a unique characteristic way of saying phonemes. Some phonemes have been shown to be more distinguishing than others [15], [16]. The aim of this approach is to use these distinguishing phonemes for verification.

During training, the phonetic string representing the password is obtained by the general purpose neural net classifier as described in chapter 3. Some phonemes are selected as they are considered to be more representative of the speaker than other phonemes. Features are computed from the speech signal corresponding to the selected phonemes. These feature vectors along with the phonetic string form the speaker model.

During application, the phonetic string of the claimed identity is force-aligned with the incoming speech. Features corresponding to the selected phonemes are computed and the distance from the reference vector is computed. The decision to accept or reject a speaker is taken by comparing this distance with a threshold.

The following issues are involved with this design.

- Which phonemes should be used?
- What kind of features should be used?
- How should these features be extracted from the selected phonemes?
- Which distance measure should be used?
- What kind of threshold should be used?

The following experiments were performed to resolve these issues.

4.3.2 Experiments

The database used for performing these experiments is the same as used for the speaker-specific speech recognition. The speaker models are generated from one utterance of each speaker for each phrase. The remaining repetitions of the phrase are used as examples of

genuine speaker saying the correct password. The same phrase spoken by other speakers is used as examples of an impostor saying the correct password. Other phrases spoken by other speakers are used as examples of the an impostor saying an incorrect password.

Effect of type of phonemes

The aim of this experiment was to decide which group of phonemes should be used for representing the speaker. Since nasals and vowels have been shown to be more distinguishing than other phonemes [15], certain sub-classes of phonemes were selected for representing the speaker. To see the effect of the class of phonemes chosen for representing the speaker, experiments were performed wherein 2 phonemes were picked from the class of (a) all vowels, (b) only stressed vowels, (c) nasals and all vowels, and (d) nasals and stressed vowels. Twentieth order LPC cepstral coefficients were averaged over the interval corresponding to the first two phonemes belonging to the above sub-groups. Hence, the two 20 dimensional vector along with the phonetic string formed the speaker model. Euclidean distance was used to compute the distance between the reference features and the test features.

Table 4.1 shows the results obtained with these different classes of phonemes. It is seen that stressed vowels give the best performance. Including nasals with the vowels lead to a significant increase in the verification error.

Table 4.1: Effect of Class of Phonemes

Phoneme Class	Phonemes	Performance
Vowels	iy ih ey ae eh ah uw uh ow aw aa ay oy er aor	86.5
Stressed Vowels	iy ey ae aa ay ow uw	88.5
Nasals + Vowels	m n iy ih ey ae eh ah uw uh ow aw aa ay oy er aor	77.7
Nasals + Stressed Vowels	m n iy ey ae aa ay ow uw	78.3

Effect of feature set

Given the phonetic string representing the password and the class of distinguishing phonemes, the questions which arise are - What kinds of features should be extracted and exactly

how should they be extracted? The aim of this experiment was to see the effect of the different kind of features.

In this experiment, 7th order PLP coefficients and 20th order LPC coefficients were used for generating the speaker models. The first two phonemes belonging to the class of vowels were used for computing the reference features. Euclidean distance was used for computing the distance between the reference features and the test features. Table 4.2 shows the verification rates for both types of features for three different methods of choosing the frame to extract features. It is observed that LPC coefficients perform significantly better than PLP coefficients.

Table 4.2: Effect of Feature Extraction

	Center Frame	Max. Amplitude Frame	Average Frame
PLP	67.1	60.0	76.1
LPC	80.3	76.5	86.5

Effect of feature extraction

The aim of this experiment was to obtain the best method of feature extraction. Since the beginning and end of each phoneme is known from the alignment, it was decided to extract features from the center frame, the maximum amplitude frame, and also the average of all the frames in that segment. Table 4.2 shows that the strategy used for picking the features can influence the performance. Averaging over the phoneme segment gives a higher performance in comparison to picking either the center frame or the maximum amplitude frame.

Effect of order

The aim of this experiment was to examine the effect of the order of model used for representing a speaker. The system was evaluated with the order of LPC varying from 8 to 20. The features were averaged over the interval of speech corresponding to the first two phonemes belonging to the broad class of stressed vowels. Table 4.3 shows that the

verification rate improved significantly when the order was increased from 8 to 10, while only small increments were observed when the order increased from 10 to 20.

Table 4.3: Effect of Order of LPC

Order of LPC	8	10	12	15	20
Verification Rate	86.6	87.9	88.2	88.7	89.1

Effect of number of phonemes

The aim of this experiment was to evaluate the effect of the number of phonemes on performance. Since most of the utterances are less than a second long and can be represented with 10 phonemes on an average, the count of the distinguishing phonemes like nasals and vowels is not very large. Twentieth order LPC coefficients were computed by averaging over the phoneme intervals. The number of phonemes picked was varied from 1 to 4 and the performance was recorded. Table 4.4 shows that increasing the number of phonemes from 1 to 2 helped considerably in improving the performance. Further increases in the number of phonemes lead to minor improvements in the performance. Due to the short length of passwords in the database, it was not possible to test the performance with more than 4 phonemes.

Table 4.4: Effect of Number of Phonemes

Number of Phonemes	1	2	3	4
Verification Rate	82.4	87.1	88.7	89.5

4.3.3 Conclusions

From all the experiments performed to investigate the usefulness of a phoneme based approach for building an SVS with low storage, the following conclusions can be drawn.

It can be concluded that the LPC gives higher performance than PLP. This is expected since PLP has been designed to smooth over the speaker dependent features [6].

It is observed that the method of averaging features over the interval of the phoneme leads to a better performance than the method of picking a particular frame.

Stressed vowels are more distinguishing than nasals or other unstressed vowels (given this system). One possible reason for this is that the frontend can identify the stressed vowels more reliably than other phonemes.

The best performance with least storage was obtained when **10th** order **LPC** coefficients were computed for the first **2** phonemes belonging to the class of **stressed vowels**. The model consisted of the phonetic string representing the password and 20 floating point numbers. Hence on an average, 60 bits are needed for representing each phonetic string (6 bits for each phonemes and 10 phonemes are present in 1 second of speech on an average) and 160 bits for representing the LPC coefficients (assuming 8 bits are needed for each coefficient). Figure 4.2 shows the distribution of the distance scores for the genuine speakers and impostors when the impostors say the correct password.

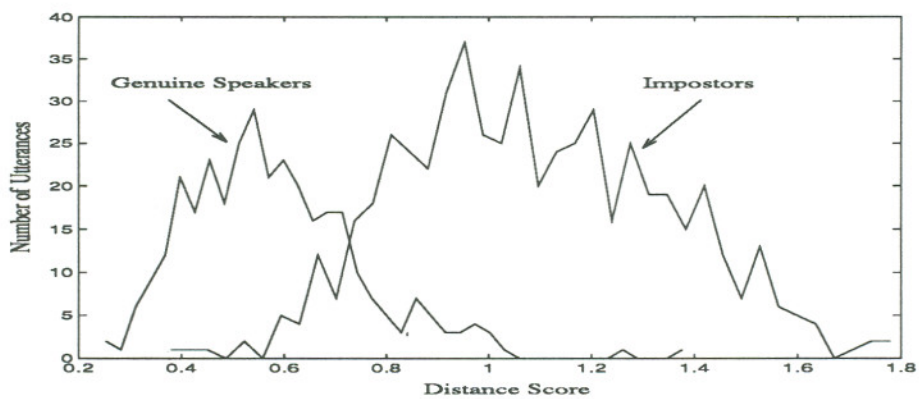

Figure 4.2: Histogram of Scores of Genuine Speakers and Impostors

Figure 4.3 shows the distribution of the Viterbi scores and the distance scores for the 3 cases of interest *ie.* genuine speakers with correct password, impostors with correct password, and impostors with incorrect passwords. It is seen that the Viterbi score can be used for rejecting most of the impostors especially those who say the incorrect password. Thus the Viterbi score can be used to determine if the correct password has been spoken

before trying to verifying the speaker. This figure also indicates the potential of multiple levels of thresholding.

Attempts made to use a text dependent and speaker dependent threshold did not show any significant improvement.

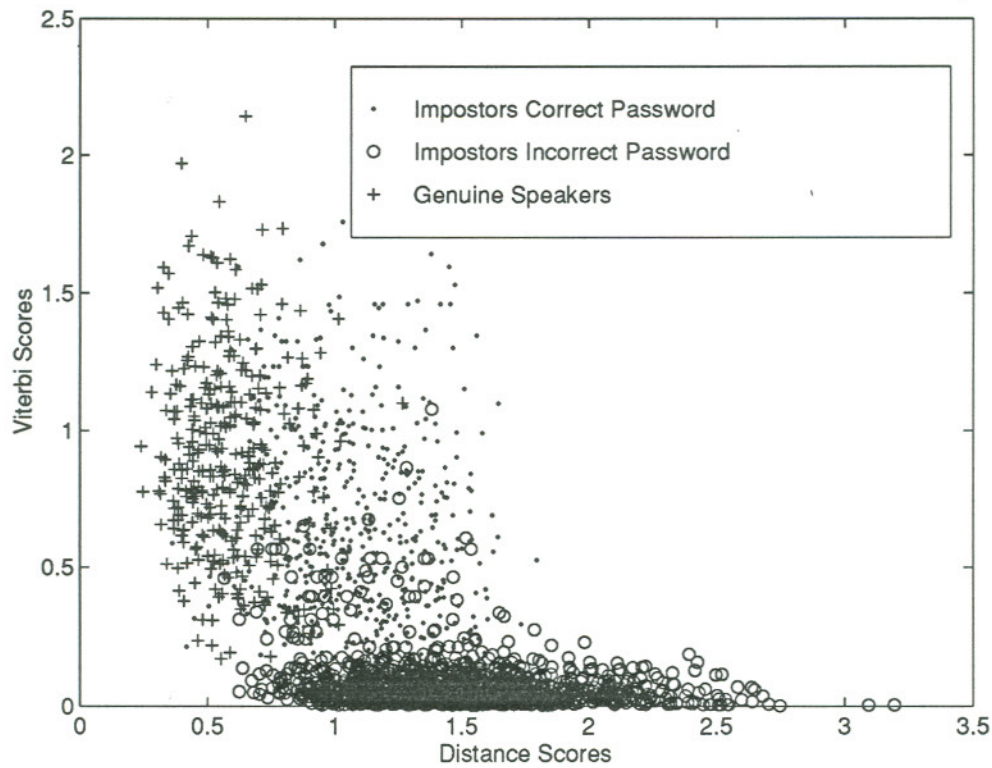

Figure 4.3: Distribution of Viterbi Scores and Distance Scores for Genuine Speakers and Impostors

4.4 Temporal Averaging Method

4.4.1 Approach

The incoming speech waveform is divided into a number of segments, say n. Features are computed from each of the n segments. The n vectors form the template of that

speaker. During application, the incoming speech is divided into n segments and features are computed. The distance from the reference features is computed and decision is made based on a threshold. This method provides average sequential information of the codeword.

The following issues are involved with this design.

- How many segments should be used?
- Should the length of the segment be fixed or should the number be fixed?
- Which features should be used?
- Which distance measure should be used?
- What kind of threshold should be used?

The following experiments were performed to resolve the above mentioned issues.

4.4.2 Experiments

The database used for performing these experiments was the same as used in the phoneme based approach. It was decided to divide the speech utterance into fixed number of segments. Experiments were performed with the number of segments varying from 1 to 32, with both the LPC and the PLP coefficients. It was decided to use Euclidean distance to measure the distance between the reference feature vector and the test feature vector. In the case of larger number of segments, some experiments were also performed with DTW [11] as a matching tool. The following sections indicate the effects of these various parameters.

Effect of number of segments

The aim of this experiment was to examine the effect of the number of segments on the performance. The number of segments was varied from 1 to 32, in multiples of 2, and reference templates were generated. Table 4.5 shows that an increase in the number of segments did not improve the performance. In fact, it degraded the performance in most

cases. The best trade-off between performance and storage requirement was obtained for the case where only 2 segments were used. The number of parameters required increases in direct proportion to the number of segments. Since one of the main aims of this design is to use a small number of parameters, the maximum number of segments used was 32.

Table 4.5: Effect of Different Parameters in Segment Based System

Feature	Number of Segments					
	1	2	4	8	16	32
LPC	91.5	93.0	90.2	86.7	79.1	72.7
PLP	85.6	85.0	81.1	76.7	62.9	62.5

Effect of type of feature

In order to see the effect of the type of feature used, the system was evaluated with 7th order PLP and 20th order LPC cepstral coefficients. Table 4.5 shows that LPC performs consistently better than PLP.

Effect of method of matching

In order to see the effect of normalization with time, the DTW approach with relaxed end-point constraints was used for computing the match between the reference template and the test template. An experiment was performed with 20th order LPC coefficients computed from 16 and 32 segments. It was observed that the DTW technique did not lead to any increase in the performance. Furthermore, the DTW technique could be applied only when there were a large number of segments, implying a large number of parameters for representing each speaker. Since one of the main goals of this thesis is to build models with low storage, the DTW approach was not pursued any further.

4.4.3 Conclusions

From all the experiments performed to investigate the usefulness of a temporal averaging approach for building a SVS with low storage, we can conclude the following.

- LPC gives better results compared to PLP.
- The system with the best performance and lowest storage requirements is obtained with 20th order LPC coefficients computed from two halves of the speech signal. Using 2 segments gives better performance than 1, 4, 8, 16 and 32. The speaker model consists of 40 floating point numbers which can be represented with 640 bits. The equal error rate obtained is 7%
- If the imposter says an incorrect codeword, then with all the methods described above, correct rejection is very high (more than 98%).
- Averaging the spectra over bins performs better than mapping the speech signal to phonemes and computing features for certain phonemes. This could be explained by the fact that the mapping performed by the neural network is not perfect. It would be interesting to compare these two methods with a database which had phonetically transcribed speech files.

4.5 Discussion

In this thesis, EER has been used as a measure of performance for all the system designs. In actual practice, the absolute threshold has to be fixed. A high threshold makes it difficult for impostors to be accepted by the system but at the risk of rejecting genuine speakers. Conversely, a low threshold ensures high acceptance of genuine speakers, but at the risk of accepting impostors. Hence the threshold should be set depending on the cost of false rejection and false acceptance, and also on the distribution of the scores of the genuine speakers and the impostors. The receiver operating characteristics (ROC) can be used for assigning this threshold. The ROC curve is obtained by assigning two probabilities, the probability of correct acceptance and the probability of incorrect acceptance, to the vertical and horizontal axes respectively, and varying the decision threshold [13].

In this thesis, the EER has been used as a measure of performance. There are three different results to be considered.

1. Acceptance of correct speaker using the correct codeword.

2. Rejection of incorrect speaker using the correct codeword.

3. Rejection of incorrect speaker using an incorrect codeword.

The test data is unevenly distributed for the 3 cases. The EER has been computed using case 1 and 2. The rejection of impostor saying an incorrect codeword *ie.* case 3, is very high (more than 98%) for all the system designs.

Use of speaker-specific and text-specific threshold should reduce the average EER very significantly for all the above cases. To set a specific threshold, the distribution of the scores of the genuine speakers and impostors saying the same password are required. In practice, there are only 3 examples of the speaker saying the password. The speaker-specific threshold cannot be reliably estimated from 3 speech files and hence have not been used.

Normalization of scores has shown significant improvement [13], [17] since it reduces the variations across channels and hand sets. This normalization was done using "cohort" groups in a text dependent SVS using a combination of only digits as the password. The "cohort" group consists of speakers whose models are close to the claimed identity. The models of these "cohort" speakers are used for normalizing the verification score before comparing with a threshold. This technique could not be used since the proposed system is text-dependent with no restrictions on the choice of the password. Normalization performed in the parameter domain, spectral equalization, [18] has been shown to be effective for text-dependent SVS based on long utterances. This normalization could not be performed in the proposed system because the utterances are short (less than a second).

The aim of this research was to build a low EER system with low storage requirements and small training periods. The training requirements of the proposed system consist of repeating the chosen password three times. This requires approximately 3 seconds of speech. The method which leads to the lowest EER and the smallest storage requirements is the algorithm wherein the speech signal is divided into 2 bins, and average 20th order LPC coefficients are computed from both the halves. The performance obtained is 93% and the storage requirements are 40 floating point numbers. For this database, the best performance was obtained with DTW with relaxed end-point constraints and using 60

segments [1]. For this method, the verification rate is 97% and the storage requirements are 1200 floating point numbers. This demonstrates the trade-off between performance and number of parameters.

There is almost 30:1 reduction in storage needed for representing the speaker with a 4% difference in verification rate. Hence, we can conclude that a concise representation of the speaker has been obtained.

[1] This work was done at OGI by Johan Schalkwyk. It has not yet been published

Chapter 5

Conclusions and Future Work

In this thesis, a new approach to voice dialing has been presented and investigated. The main advantage of this new approach is very low storage requirements and low error rates for representing the speaker's identity and the speaker's speech.

It was shown that it is feasible to use the output of a task independent neural network for generating word-models to represent a user's speech. Procedures for selecting a "best" word model from among several tokens were investigated, and these speaker-specific models were shown to produce competitive recognition results.

Compared to conventional techniques, the new method of performing speaker dependent speech recognition lead to a reduction of about 1:300 in the storage requirements with comparable recognition accuracy of 97%. Using averaging of temporal spectra for speaker verification lead to a reduction of about 1:30 with verification rate of 93%.

A complete working voice dialing system that incorporates this new technology has been developed. This system has been used in the laboratory. It is currently in the process of being deployed to the telephone network.

5.1 Research Challenges and Future Directions

There are a number of interesting problems that should be addressed through future research. These include the following issues.

1. The algorithm for speaker specific speech recognition could be evaluated for only 21 phrases. It will be interesting to evaluate this system with a larger number of labels.

2. One of the main drawbacks of the database used in this thesis is the lack of variability due to different sessions and different channels. Hence, the robustness of the proposed algorithms could not be evaluated. There is a need for a standard database which incorporates the variability introduced by different telephone lines, handsets, and environmental noise.

3. One of the main challenges in this system was dealing with confusable labels. Currently, the user is requested to substitute for one of the confusable labels. These confusable labels can be dealt with some special strategy. This strategy would aim at finding the small differences which distinguish the 2 labels.

4. In this thesis, the "goodness" of a phonetic string was decided by using Viterbi scores. The different phonetic strings themselves may be used to decide the best string by some averaging technique. A transition network may be used between the phonemes. The model used for representing the word may be generated from the N models instead of picking one of them.

5. A speaker independent VQ based classifier can be used instead of the neural network for classification of the frames to phonemes.

6. Speaker Adaptation can be performed only after the system has been used. Different techniques of speaker adaptation can be investigated once the system starts running. Adaptation can be performed for speaker verification models and label recognition models. Adaptation should make these speaker more robust.

More functionality can be added to the system *ie.* it can be used for more than dialing of numbers in the database. Any task which is speaker dependent and has a small vocabulary can be implemented with this system. Building a spoken interface with the computer, voice mail, banking are some of the possible applications.

Bibliography

[1] Jun Noguchi, Shinsuke Sakai, Kaichiro Hatazaki, Ken-ichi Iso and Takoe Watanabe. An Automatic Voice Dialing System developed on PC Speech I/O Platform. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 699–702, Yokohama, Japan, September 1994. The Acoustical Society of Japan.

[2] D. Geller, R. Haeb-Umbach and H. Ney. Improvements in speech recognition for voice dialing in the car environment. *Proceedings of Speech Processing in Adverse Conditions*, pages 203–206, November 1992.

[3] L. R. Rabiner and J. G. Wilpon. Some performance benchmarks for isolated word speech recognition systems. *Computer Speech and Language*, 2:343–357, 1987.

[4] Ronald Cole, David G. Novick, Mark Fanty, Pieter Vermeulen, Stephen Sutton, Dan Burnett and Johan Schalkwyk. A prototype Voice-Response Questionnaire for the U.S. Census. In *Proceedings of the International Conference on Spoken Language Processing*, volume 2, pages 683–686, Yokohama, Japan, September 1994. The Acoustical Society of Japan.

[5] Jay Naik. Speaker verification over the telephone network: Databases, algorithms and performance assessment. In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 31–37, Martigny, Switzerland, April 1994. Gerard Chollet, IDIAP Research Center, Case Postale 609, CH-1920 Martigny, Switzerland (esca@idiap.ch).

[6] Hynek Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of Acoustical Society of America*, 87(4):1738–1752, April 1990.

[7] E. Barnard and R. A. Cole. A neural-net training program based on conjugate-gradient optimization. *Technical Report CSE, Oregon Graduate Institute of Science and Technology*, 89(014), July 1989.

[8] Ronald Cole , Mark Fanty , Mike Noel and Terri Lander. Telephone Speech Corpus Development at CSLU. In *Proceedings of the International Conference on Spoken*

Language Processing, volume 4, pages 1815–1818, Yokohama, Japan, September 1994. The Acoustical Society of Japan.

[9] Li Jiang. Neural network based context-dependent modelling for speech recognition. *Research Proficiency Report, Department of Computer Science and Engineering, Oregon Graduate Institute of Science and Technology*, May 1994.

[10] Boulard H. and Wellekens C. J. Links between markov models and multi-layer perceptrons. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(12):1167–1178, December 1990.

[11] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. of Acoustics, Speech and Signal Proc.*, ASSP-26:43–49, February 1978.

[12] J. H. Juang, D. Y. Wong and A. H. Gray . Distortion performance of vector quantization for lpc voice coding. *IEEE Trans. of Acoustics, Speech and Signal Proc.*, ASSP-30(2):294–304, April 1982.

[13] Sadaoki Furui. An overview of speaker recognition technology. In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 1–9, Martigny, Switzerland, April 1994. Gerard Chollet, IDIAP Research Center, Case Postale 609, CH-1920 Martigny, Switzerland (esca@idiap.ch).

[14] Frederic Bimbot, Gerard Chollet and Andrea Paolini. Assessment methodology for speaker verification and identification systems. In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 75–82, Martigny, Switzerland, April 1994. Gerard Chollet, IDIAP Research Center, Case Postale 609, CH-1920 Martigny, Switzerland (esca@idiap.ch).

[15] J. P. Eatock and J. S. Mason. A quantitative assessment of the relative speaker discriminating properties of phonemes. In *IEEE Proceedings of International Conference on Acoustics, Speech and Signal Proc.*, volume 1, pages 133–136, Adelaide, Australia, April 1994. Piscataway, N.J., IEEE.

[16] Eluned S. Parris and Michael J. Carey. Discriminative phonemes for speaker identification. In *Proceedings of the International Conference on Spoken Language Processing*, volume 4, pages 1843–1846, Yokohama, Japan, September 1994. The Acoustical Society of Japan.

[17] A. E. Rosenberg, Joel DeLong, Chin-Hui Lee, Biing-Hwang Juang and Frank K. Soong. The use of cohort normalized scores for speaker verification. In *Proceedings*

of the International Conference on Spoken Language Processing, volume 1, pages 599–602, Banff, Canada, September 1992. University of Alberta.

[18] Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Trans. of Acoustic, Speech and Signal Proc.*, 29(2):254–272, 1981.

[19] L. R. Rabiner and B. Juang. *Fundamentals of Speech Recognition.* New York, Prentice Hall, 1993.

[20] C. Myers, L. R. Rabiner and A. E. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Trans. of Acoustics, Speech and Signal Proc.*, ASSP-28(6):623–635, December 1980.

[21] Alex Waibel and Kai-Fu Lee. *Readings in Speech Recognition.* San Francisco, Morgan Kaufmann Publishers,Inc, 1990.

Appendix A

Registration Form For Voice-Dialing Demonstration

Please fill out this form before calling the number below to register your voice for speed dialing. This will help ensure fast and efficient registration for accurate speed dialing.

Registering your voice will take about 5 minutes. We will first assign you a 4 digit identity code which you need to use in the future. We will then ask you to say your password three different times. Your password should be a month, day and year, such as "June fifteenth seventeen eighty one."

You will be asked to say each speed label three times. You will then be asked to key in the phone number associated with the label. A Speed Label is the word or phrase associated with the phone number you want to call.

You can choose up to 5 speed labels. How you select your speed labels is important. The more different they sound, the better the system will work. So "Tom, Don and Ron" are bad labels, and so are "mother, brother and father." But "Tom Jones, Don and Ronald" will work fine, and so will "mother, dad and brother John."

Please wait for the beep after all questions before speaking. During application, a *command* refers to "call" followed by a speed label.

Identity Code: _____

Password: _____

How many speed labels would you like to register? _____

	SPEED LABEL	AREA-CODE	NUMBER
1.			
2.			
3.			
4.			
5.			

Phone Number: (503) 690-1298

Appendix B

Prompts for Registration

This is an example of a conversation during registration. The assumption is that the system is performing perfectly and the user is friendly.

```
System : Thank you for calling the voice dialing demonstration.
         To continue, you must have your completed registration form.
         Do you have that with you now? Please say yes or no.
User   : Yes

         [ Yes/No Recognition ]

System : Your ID is 1503 ie. 1503.
         We will first register your password, and then up to five
         words or phrases and associated phone numbers. We will now
         proceed to record your password. Please wait for the beep
         before speaking.
System : Please say your password now.
User   : abcadabra
System : Please say your password a second time.
User   : abcadabra
System : Please say your password a third time.
User   : abcadabra

         [Generation of model for speaker verification.
          Pick the best password. ]

System : Would you like to register any speed labels? Say yes or no.
User   : Yes
```

[Yes/No Recognition]

System : Please enter the number of speed labels you would like to register.
User : (2)
System : Please refer to line 1 on your form and read the label.
User : Home
System : Please repeat the label.
User : Home
System : Please repeat the label.
User : Home

[Generation of model for label 1.
Pick the best model]

System : Please enter the area code now.
User : (111)
System : Please enter the 7 digit phone number now.
User : (111-1111)
System : Please refer to line 2 on your form and read the label.
User : Office
System : Please repeat the label.
User : Office
System : Please repeat the label.
User : Office

[Generation of model for label 2.
Pick the best model.
Cross-check with label 1]

System : Please enter the area code now.
User : (111)
System : Please enter the 7 digit phone number now.
User : (222-2222)
System : Thank you for calling the voice dialing demonstration.

Appendix C

Prompts for Application

This is an example of a conversation during application. The assumption is that the system is performing perfectly and the user is friendly.

```
System : Thank you for calling the voice dialing demonstration.
         Please enter your 4 digit identity code.
User   : 1503

         [Retrieval of database belonging to 1503 with models
         for verification and speech recognition.]

System : Please say your password.
User   : Abracadabra

         [ Speaker Verification ]

System : Command?
User   : Call office

         [ Speaker Specific Speech Recognition ]

System : Dialing (111) 222-2222
System : Would you like to give another command?
User   : No

         [ Yes/No Recognition ]

System : Thank you for using the voice dialing system.
```

Biographical Note

Neena was born on 19th February, 1971 in Bombay, India. She did all her education in Bombay. She graduated with a B. Tech degree in Electrical Engineering from I.I.T. Bombay in June 1992.

Neena then joined the Oregon Graduate Institute of Science and Technology for a M.S. degree in Electrical Engineering in September 1992. Her current research interests center around the development of real world speech applications.

List of publications include :

(1) "Perceptual benchmarks for automatic language identification", Yeshwant K. Muthusamy, Neena Jain and Ronald A. Cole, *Proceedings of the 1994 International Conference on Acoustics, Speech and Signal Processing*, vol. 1, Adelaide, South Australia, pp.333-6, April 1994.