Improving Wordspotting Performance with Limited Training Data

by

Eric I-Chao Chang

S.B., Massachusetts Institute of Technology (1990)
S.M., Massachusetts Institute of Technology (1990)
Electrical Engineer, Massachusetts Institute of Technology (1994)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1995

Author ....
Department of Electrical Engineering and Computer Science
May 16, 1995

Certified by .... 5/16/95 ....
Richard P. Lippmann
Senior Technical Staff
Thesis Co-Supervisor

Certified by .... 5/16 ....
David H. Staelin
Professor of Electrical Engineering
Thesis Co-Supervisor

Accepted by ....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Improving Wordspotting Performance with Limited Training Data

by

Eric I-Chao Chang

Submitted to the Department of Electrical Engineering and Computer Science
on May 16, 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis addresses the problem of limited training data in pattern detection problems where a small number of target classes must be detected in a varied background. There is typically limited training data and limited knowledge about class distributions in this type of spotting problem and in this case a statistical pattern classifier can not accurately model class distributions. The domain of wordspotting is used to explore new approaches that improve spotting system performance with limited training data. First, a high performance, state-of-the-art whole-word based wordspotter is developed. Two complementary approaches are then introduced to help compensate for the lack of data. Figure of Merit training, a new type of discriminative training algorithm, modifies the spotting system parameters according to the metric used to evaluate wordspotting systems. The effectiveness of discriminative training approaches may be limited due to overtraining a classifier on insufficient training data. While the classifier's performance on the training data improves, the classifier's performance on unseen test data degrades. To alleviate this problem, voice transformation techniques are used to generate more training examples that improve the robustness of the spotting system. The wordspotter is trained and tested on the Switchboard credit-card database, a database of spontaneous conversations recorded over the telephone. The baseline wordspotter achieves a Figure of Merit of 62.5% on a testing set. With Figure of Merit training, the Figure of Merit improves to 65.8%. When Figure of Merit training and voice transformations are used together, the Figure of Merit improves to 71.9%. The final wordspotter system achieves a Figure of Merit of 64.2% on the National Institute of Standards and Technology (NIST) September 1992 official benchmark, surpassing the 1992 results from other whole-word based wordspotting systems.

Thesis Co-Supervisor: Richard P. Lippmann
Title: Senior Technical Staff

Thesis Co-Supervisor: David H. Staelin
Title: Professor of Electrical Engineering

Acknowledgments

I would like to thank my co-supervisors Dr. Lippmann and Prof. Staelin for their generous advice and insightful directions. Working with them has been a privilege. Their questions always led to better ways of organizing my thoughts. I thank Dr. Lippmann for the guidance and support he provided through my four year stay at Lincoln Laboratory. I am grateful to Prof. Staelin for shepherding me through the many ups and downs on the path toward a Ph.D.

Thanks to Professor Kenneth Stevens and Dr. Victor Zue for their contributions in making this thesis more understandable and for their professional advice. I have also benefited greatly from their courses in Speech Communication and Automatic Speech Recognition.

Thanks to the members of the Speech Systems Technology Group at Lincoln Laboratory. Confucius once said that among three people encountered, at least one can be a teacher. In my case everyone at Lincoln has truly taught me something useful and profound. Both Elliot Singer and Beth Carlson were involved in the wordspotting effort and provided much help along the way. Charles Jankowski Jr. was a wonderful officemate for sounding out ideas and did a wonderful job of proofreading my thesis draft. Richard Lippmann, Marc Zissman, Doug Reynolds, and Beth Carlson participated in the lunch discussion group on speaker adaptation and contributed much lively discussion. Tom Quatieri provided the sinusoidal transform coding algorithm which made experiments with voice transformation possible. Linda Kukolich kindly spent many hours participating in the human wordspotting experiments and also wrote the great *LNKnet* programs that make pattern classification experiments almost a child's play. Linda Nessman kept the group running through her cheerful demeanor. Cliff Weinstein and Gerald O'Leary provided the freedom and the encouragement for me to explore my thesis topic.

I am blessed with making many friends throughout my stay at M.I.T. The group at Next House, John, Scott, Matt, Curtis, and Andrew, has always been a source of support through the long nights doing problem sets. Judy, Kevin, and Victor were great study group pals when we were preparing for qualification exams. A group of classmates who have similar interests in speech processing provided much assistance. Hwa-Ping has been a great classmate through the courses in Digital Signal Processing and Digital Speech Processing. Thanks to Sharlene, Helen, and Jeff for generously donating a portion of a fine

Sunday afternoon to listen to my thesis defense dry run. The friends I have met through my association with ROCSA, Shun-Min, Yusin, Cliff, Che-Chi, and countless others have provided many fun diversions and the crucial network of support. My TA and good friend Ying provided help and encouragement throughout the qualification exam process. My former roommates Lian and Hsien-Jen always provided interesting conversations for breaks after long periods of typing.

Thanks to my parents and my siblings, Mr. and Mrs. Yin-Fu Chang, Terisa, Stephanie, and Lily, for their constant faith and encouragement. Everything I have achieved so far would not have been possible without them. A special thanks to my godmother, Mrs. Laurie Chen, for the concern and support that she has shown throughout the years. Lastly, thanks to my wife Chuin-Ping for making the last few years as rewarding personally as professionally. She is always there to share the joy and the difficulties. She also manages to run the household and free me from many daily chores even when she has considerable coursework. Words alone are not enough to convey my gratitude and I look forward to showing her every day that her efforts are worthwhile.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As computers become more ubiquitous in our daily lives, the need to interact with computers and other modern technology becomes more frequent and widespread. Thus, there has been an effort to improve the interface between human beings and computers. Speech is one of the most natural and efficient media that humans use to communicate. Over the years, the ability to communicate with computers using speech has gradually improved. Recently, through research on new algorithms and with faster computers, discrete, limited vocabulary, speaker-independent speech recognition systems have become a reality. However, most systems so far have been designed for use in a quiet office environment with adaptation toward a specific speaker.

Wordspotting has become an increasingly important application in speech recognition since the amount of computation required for wordspotting is feasible on currently available computer systems [10, 25, 45, 67, 68, 74, 75, 79]. Some wordspotter research has focused on automatic operator services [10, 74, 79], others have focused on monitoring keywords in unconstrained speech [25, 28, 52, 68, 83]. There have also been studies on using wordspotting to index an audio message [78].

Instead of the traditional focus of dictation, many current research efforts focus on enabling human-machine interfaces using speech. Recognition of certain command words is sometimes sufficient to implement such an interface system. When using a wordspotting system, a user speaks certain keywords embedded in a sentence and the system detects the occurrence of these keywords. One example is an automatic directory service, illustrated in Figure 1-1. The user may say, *"May I have the hardware department please."* The computer

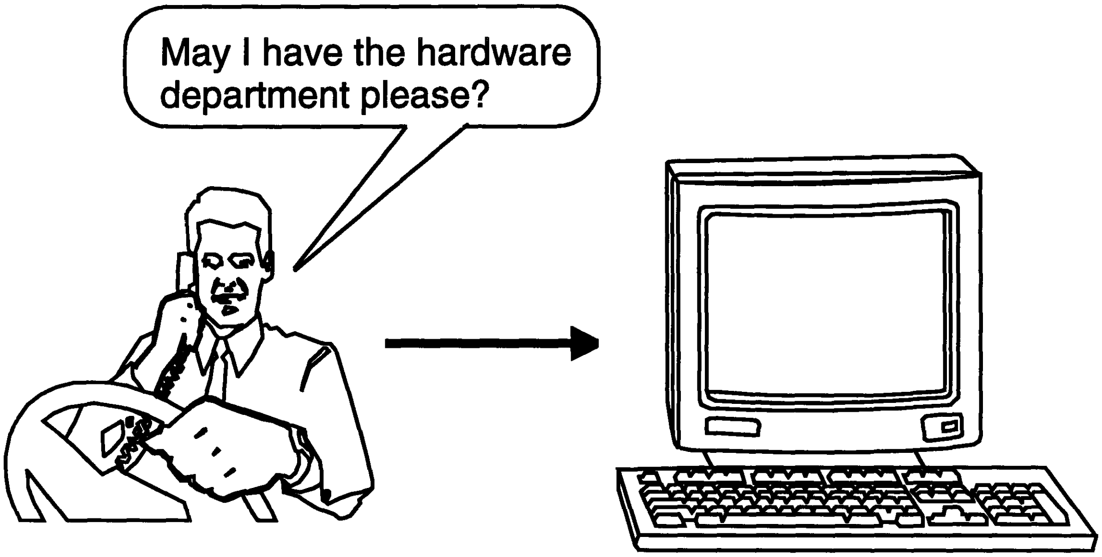


Figure 1-1: An illustration of using a wordspotting system to direct a caller's inquiry to the appropriate department.

will spot for the keyword *hardware* and direct the user to the hardware department even though the keyword is embedded in extraneous speech (i.e. *department, please*) that is not in the system's list of recognizable keywords. When users speak spontaneously, there are many grammatical errors, pauses, and disfluencies that a continuous speech recognition system may not be able to handle. For these situations, a wordspotting system will concentrate on spotting particular keywords and ignore the extraneous speech. Such a system can also work well in other command and control systems such as voice dialing a cellular phone while one is driving, changing a television set's channel, and retrieving voice mail and email messages through the telephone.

1.1 Problem Statement

Most modern pattern recognition systems depend on a statistical approach to extract useful information about different classes of patterns from a set of training patterns. However, difficulty in distinguishing between different classes arises when the amount of training data is not sufficient to characterize all possible patterns of a particular class. For example, a speech recognition system may be trained using speech patterns from tens or hundreds of speakers. However, there may still be speakers (who differ in accent, age, or speaking style) who are not well modelled by the pool of training speakers. A speech recognition system will perform poorly for these unrepresented speakers. In many cases, the number of training speakers is even more limited and thus the problem of mismatch between a novel speaker and the system trained using limited training data arises even more frequently.

There are two major problems associated with the lack of training data: inter-word variability and inter-speaker variability. An example of inter-word variability is in determining the difference between two similar sounding words such as *card* and *hard*. These two words sound very similar to each other except for the beginning of the word. Most modern speech recognition systems rely on a statistical modelling technique, called Hidden Markov Models (HMM), to model the temporal variability of speech [62]. The traditional Hidden Markov model training technique, using the Expectation-Maximization algorithm [14], utilizes only the training data for a particular keyword, e.g., *card*, to estimate the model for that keyword. With limited training data, a wordspotter trained to spot the word *card* will produce high output scores for similar sounding words such as *hard* and *guard*. When *card* is the word to be detected, detection of the words *hard* and *guard* are called *false alarms*. The goal in wordspotting is to ensure a high detection rate with a low false alarm rate. A system which generates many false alarms is not useful. Given limited training data, a different training technique which trains the models with the goal of discriminating true hits from false alarms can provide better overall performance.

Inter-speaker variability represents another major problem in speech recognition. The speakers in the training set may not represent the user population sufficiently. For example, speakers differ from each other in sex, age, accent, physiological dimension, glottal characteristics, etc. With the large number of dimensions in which speakers differ, even hundreds of speakers used in training speech recognizers may not cover a user population well.

To train a high performance wordspotter, data which adequately represent the inter-speaker variability as well as inter-word variability are needed. One solution is to collect more data which represent better the variabilities encountered among the user population. However, since collecting more training data can be costly, techniques which allow the training of models which maximize the use of available training data are valuable. This thesis develops a high performance whole-word based wordspotter as the baseline wordspotter and investigates two complementary approaches to improve the performance of a high performing wordspotter through better use of existing data. These two approaches, *Figure of Merit Training*, and *Voice Transformation* are presented in the next section.

1.2 Proposed Approach

The research described in this thesis focuses on developing a high performance wordspotter and exploring techniques that can be applied to improve wordspotter performance with limited training data. The following three topics that have been completed in the course of this research are discussed in the subsequent chapters:

- Development of a High Performance Wordspotter
- Investigation of Figure of Merit Training
- Investigation of Applying Voice Transformations to Generate More Training Data

A basic, high performance wordspotter is first implemented and is the basis for additional improvements. To maximize the discriminative ability of the models, a new technique, *Figure of Merit Training,* is studied. Finally, to improve the wordspotter's robustness against speaker variability, *Voice Transformation* techniques are introduced and studied. More complete descriptions of these steps are presented below.

1.3 A High Performance Wordspotter

1.3.1 Introduction

The function of a wordspotter is to process continuous speech input and to generate hypotheses of where a keyword may have occurred. Each hypothesis consists of the time and the duration of the keyword and the wordspotter's confidence in that hypothesis. In this thesis, each hypothesis of keyword occurrence is called a ***putative hit.***

A high performance wordspotter that has the state of the art performance was implemented as a part of this thesis. By starting with a high performance wordspotter, new techniques that are developed later would be applicable to recognition systems that are already very good. Furthermore, any resulting improvements would be significant in comparison to current state of the art.

1.3.2 A Whole-Word Based Wordspotter

The techniques studied in this work were tried on a whole-word wordspotter. While other more complicated approaches such as a large vocabulary continuous speech recog-

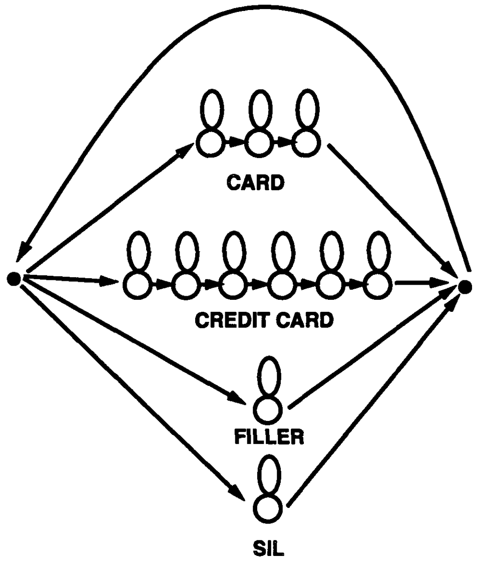


Figure 1-2: Network structure of the whole-word wordspotter

nizer (LVCSR) system are possible, the whole-word approach has a structure which allows for discriminative training for each individual keyword without affecting other keywords. Comparisons between the LVCSR approach and the whole-word approach are presented in Section 8.2.

Each keyword to be spotted is modelled by a distinct Hidden Markov Model (HMM) [62] while speech background and silence are modelled by general filler and silence models respectively. The structure of the wordspotter is shown in Figure 1-2. For each keyword, a normalized score is generated at 10 ms intervals. The normalized score is the end state Viterbi log likelihood of each keyword minus the end state log likelihood of the filler model and approximates the *a posteriori* probability of keyword occurrence. As more speech is processed, the Viterbi log likelihood of the end state will decrease, but by calculating the normalized score as the difference between the keyword's log likelihood and the filler's log likelihood, the normalized score will be positive when the keyword model matches the incoming speech better than the general filler model. Putative hits are generated by detecting peaks in the normalized score of the wordspotter. The score for each putative hit is the normalized score at each peak. More detailed description of the putative hit generation process can be found in Section 4.5.

The normalized score for each keyword is processed with a peak-picking algorithm to generate a set of putative hits. The wordspotter generates a putative hit if the putative hit's score is above a pre-set threshold. Lowering the threshold generates more putative hits and also more false alarms. Thus, by varying the threshold, the number of false alarms generated by the wordspotter over a duration of speech signal, or the false alarm rate, can be adjusted.

1.4 Figure of Merit Training

1.4.1 Introduction

Currently, most Hidden Markov Models are trained with the Expectation-Maximization algorithm [14]. Given a series of output feature vectors o and a model m, the EM algorithm iteratively calculates the likelihood that the feature vector is generated by the model, $p(o|m)$, and maximizes $p(o|m)$. The EM algorithm is an iterative method that has been shown to converge to a local maxima of $p(o|m)$. Traditional techniques of maximum likelihood estimation assume that enough data are available to train all models in the system and that models of the distributions of input features are accurate. Both assumptions do not hold in reality. Recently there has been much work in the area of corrective training [8, 38]. By training a speech recognizer to maximize recognition accuracy instead of likelihood, improvements in recognition performance have been reported. For example, Lee et al. reported a reduction in word error rate (including substitutions, deletions, and insertions) from 29.4% to 26.3% on the large vocabulary Resource Management task using corrective training [38]. Many other approaches have been proposed which use methods other than Maximum Likelihood Estimation to estimate models for speech recognition [38, 68].

The difference between a maximum likelihood approach and a discriminative approach is illustrated in Figure 1-3. In this figure, the two regions represent the distribution of patterns from two different classes. The two axes represent two possible input features. For example, to discriminate between vowels, the first two spectral peaks of each vowel can be used as the input features. The maximum likelihood estimation approach separately models each class with a Gaussian distribution, and the line where the Gaussian distribution values are the same is the boundary between two classes. But with limited amount of training data, outlier data points that are far away from the class boundary can distort the estimated distribution for each class and the resulting class boundary may not minimize the number of classification errors. The discriminative approach, on the other hand, focuses on reducing

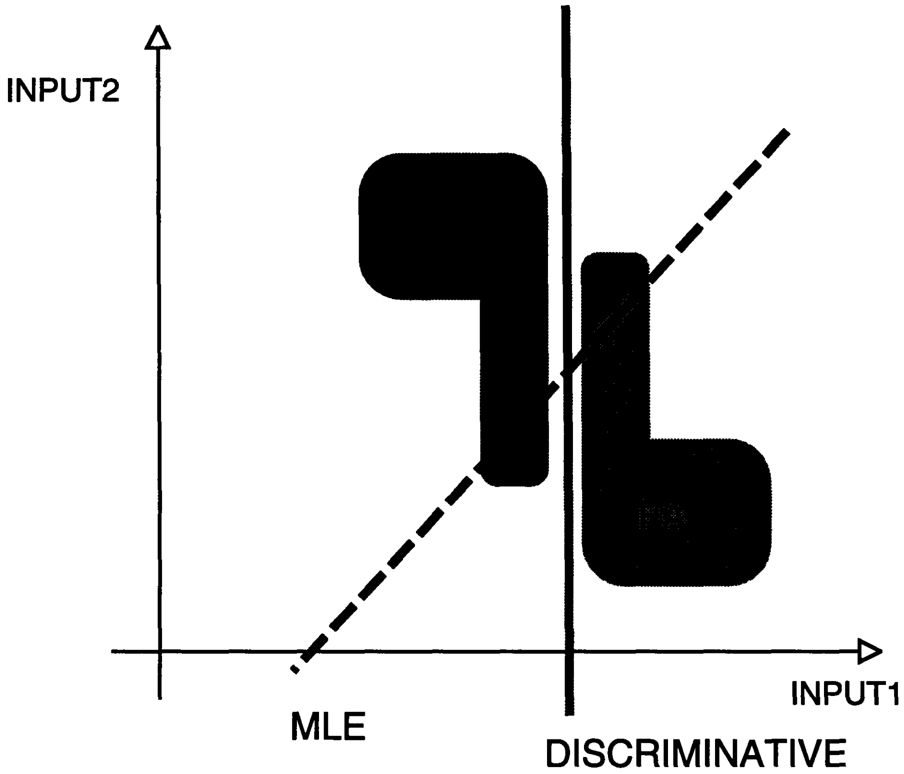


Figure 1-3: An illustrated comparison of Discriminative Training vs. Maximum Likelihood Training.

the number of misclassified data points and thus adjust the boundary between the two classes with more weight given to the data points close to the boundary.

A simple set of experiments was performed to illustrate the difference between a maximum likelihood training approach and a discriminative training approach. A set of two classes of patterns were generated. Each class contains patterns from three Gaussian distributions. Fifty percent of the patterns are from the main Gaussian distribution (H2 and F2) while twenty-five percent of patterns are from two other Gaussian distributions (H1, H3, F1, and F3). To illustrate the problem of not knowing the underlying probability distribution of each class in real life, only one Gaussian distribution is used to model each class. The classifiers were trained using two approaches:

- A Gaussian classifier, a maximum likelihood training approach [39].
- Incremental Radial Basis Function (IRBF) Classifier, a discriminative training approach [44].

These two types of classifiers have exactly the same structure, the only difference between them are the training algorithms. The classifiers were trained using the software package *LNKnet* that has been developed at Lincoln Laboratory by Richard Lippmann, David Nation, and Linda Kukolich [44]. Figure 1-4 shows the Gaussian distributions in the Gaussian classifier that represent each class. Figure 1-5 shows the Gaussian distributions in the IRBF classifier. A boundary between two classes can be drawn by plotting the points in the input space where the classifier outputs for the two classes are equal. The boundaries drawn by the Gaussian classifier and the IRBF classifier are very different. The Gaussian classifier's distributions try to model three Gaussian distributions with just one Gaussian distribution, and thus the boundary created by the Gaussian distributions do not separate the two classes well. On the other hand, the Gaussian distribution parameters of the IRBF classifiers were trained to minimize classification error rate, so patterns that are far away from the boundary between two classes do not affect the resulting Gaussian distribution.

The example just shown illustrates the benefit of discriminative training approaches for minimizing static pattern classification errors. Other discriminative training approaches can be developed for other pattern classification tasks. A new discriminative training approach, called Figure of Merit Training, that is tailored for the task of wordspotting is introduced in this thesis.

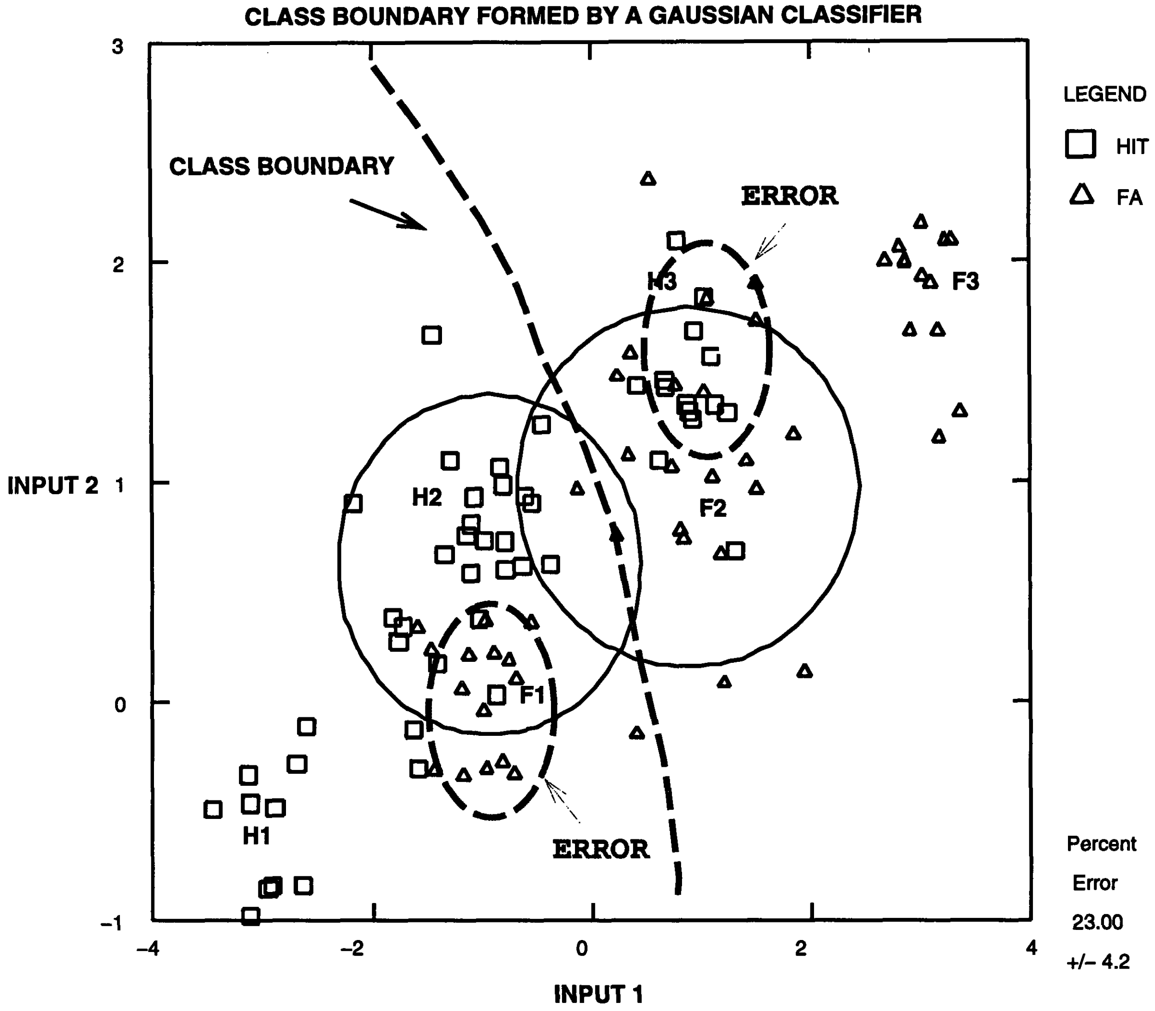


Figure 1-4: Class boundary formed by a Gaussian classifier on a database consist of patterns from 6 Gaussian distributions. The class HIT consists of patterns from Gaussian distributions H1, H2, and H3. The class FA consists of patterns from Gaussian distributions F1, F2, and F3. The ovals illustrate the Gaussian distributions of the Gaussian classifier. The dashed ovals highlight where classification errors occur.

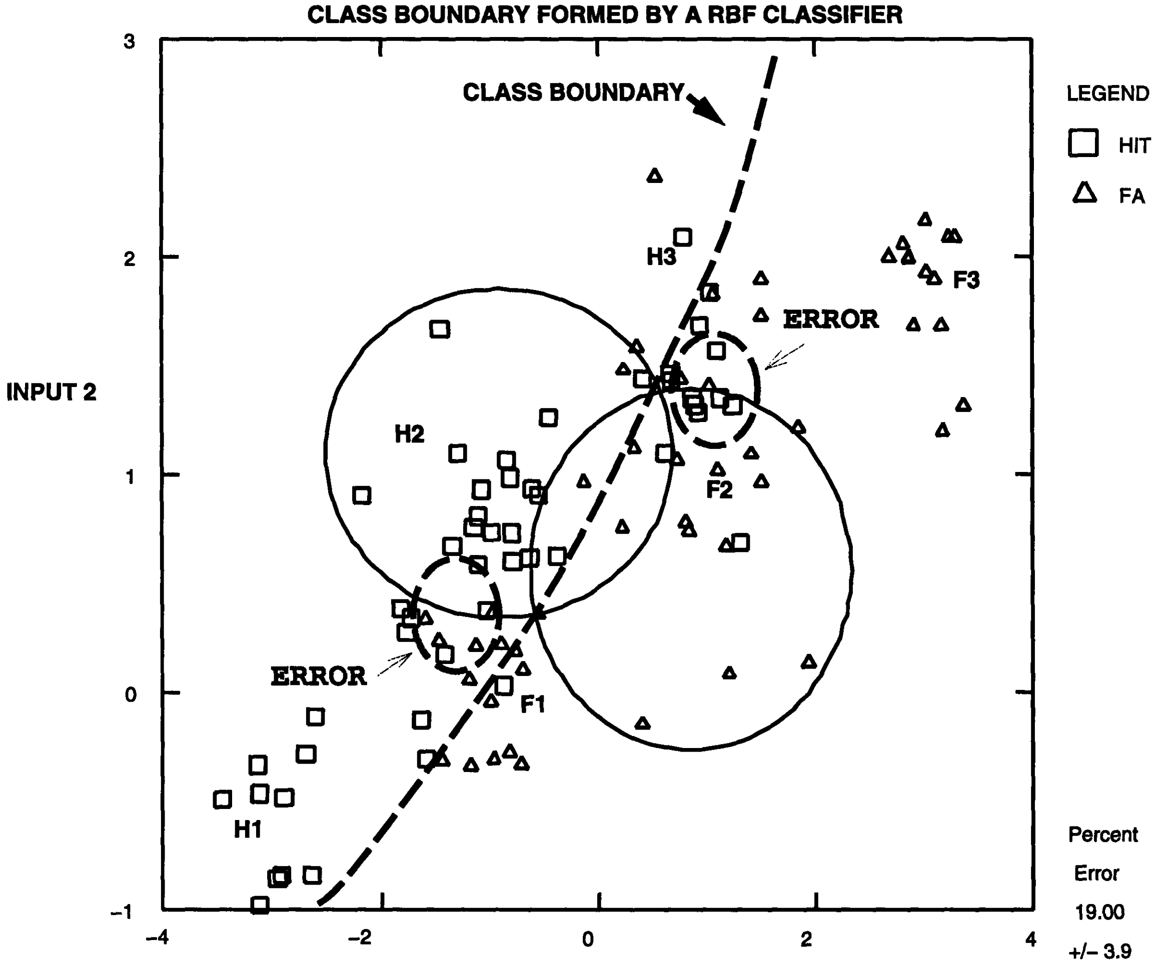


Figure 1-5: Class boundary formed by a RBF classifier on a database consist of patterns from 6 Gaussian distributions. The class HIT consists of patterns from Gaussian distributions H1, H2, and H3. The class FA consists of patterns from Gaussian distributions F1, F2, and F3. The ovals illustrate the Gaussian distributions of the RBF classifier. The dashed ovals highlight where classification errors occur.

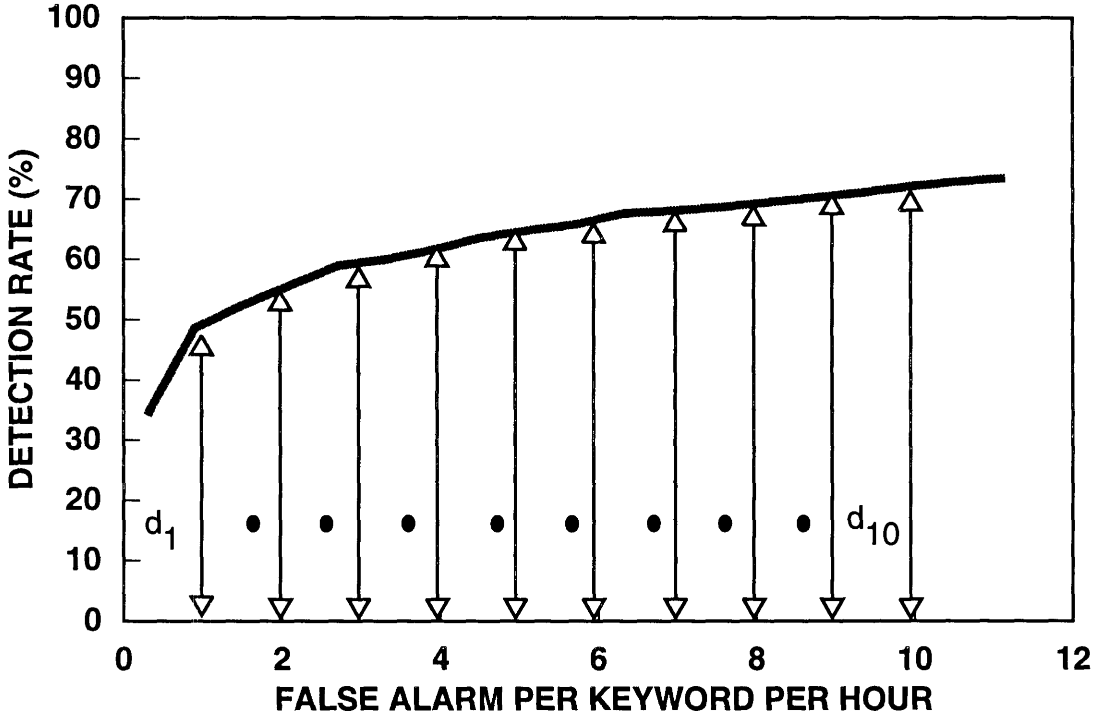
1.4.2 Figure of Merit

Wordspotting systems are currently evaluated using a metric called *Figure of Merit.* The Figure of Merit is calculated by averaging the detection rate of the wordspotter over a range of false alarm rates. Figure 1-6 illustrates the calculation of the Figure of Merit from a receiver operating characteristics (ROC) curve. The ROC curve is a plot of the detection rate versus the false alarm rate of a detection system. The Figure of Merit summarizes the performance of a spotting system in detecting a signal over a range of false alarm rates and is used frequently in evaluating spotting systems. Currently, a standard measurement defined by National Institute of Standards and Technology calls for averaging the detection rate over the range of 0 to 10 false alarms per keyword per hour [53], as shown in Figure 1-6. The Figure of Merit evaluates a wordspotting system differently than conventional speech recognizers. A conventional speech recognition system generates a stream of phone labels or word labels for each input sentence and generally all words are weighted equally in importance. The common metric for evaluating such a continuous speech recognition system is the word error rate. For example, if a string of digits sent to the recognizer is *one, three, five, two, seven*, and the recognizer's output is *nine, three, five, two, seven*, then the word error rate for this speech recognizer on this sentence is 20%. Typically, the amount of confidence that the speech recognition system has in each label is not used during scoring.

On the other hand, a wordspotting system generates a list of hypotheses, called putative hits, that consist of the location of a word and a confidence score. A higher confidence score indicates that the system is more certain that the specified word actually occurs at the prescribed location. When the Figure of Merit is calculated, the confidence score plays an important role since the putative hits are sorted by the confidence score and the number of false alarms encountered at each confidence score level determines the false alarm rate at the score level.

1.4.3 Training to Maximize Figure of Merit

A new approach, called *Figure of Merit Training* [9, 43], which attempts to train model parameters to maximize the Figure of Merit of a wordspotter, is introduced in this thesis. To adjust the model parameters, a gradient relating the Figure of Merit to each parameter



$$\text{FOM} = \frac{1}{N} \sum_{i=1}^{N} d_i = 63.3$$

Figure 1-6: The Figure of Merit (FOM) is the average detection rate over a range of false alarm rates.

in the wordspotting system is derived. Each putative hit's impact on the overall Figure of Merit is calculated by interpolating the changes in overall Figure of Merit as the putative hit's score is varied over a small range. After the gradient has been calculated, parameters are adjusted in a direction that increases the Figure of Merit. The discriminative ability of wordspotter models is improved as examples of keywords and false alarms are used to modify the parameters of wordspotter models.

The FOM training approach has been implemented on a wordspotting system. Analysis has been performed on the type of errors that are removed by FOM training and more detailed results are presented in Chapter 5.

1.5 Voice Transformations

The speech recognition process can be modelled as a communication problem, where a speaker speaks a sentence with certain intent, the sound is generated by the speaker's vocal system and then decoded by the listener. Although each individual may intend to say the same sentence, the actual acoustic signal that is sent out will depend on individual characteristics such as gender, accent, age, physiological characteristics, and speech habits. The variability inherent in the manifestation of the acoustic signal requires the collection of a large amount of training data to model this variability. Many approaches have been tried to compensate for this speaker variability. The taxonomy of possible approaches can be roughly divided into three different branches: obtaining training data from a large number of speakers, supervised speaker adaptation and unsupervised speaker adaptation.

Most modern speech recognizers rely on the first approach to compensate for speaker variability. By using data collected from a large number of speakers and complex models that can represent the speaker variability, the speech recognition system is made more robust to speaker variability. The drawback of this approach is that collecting data from a large number of speakers is frequently expensive.

Supervised speaker adaptation requires that the speaker speak sentences and that the content of these sentences is known. In fact, many more restrictive systems require that the same identical sentence be spoken for adaptation [22, 23, 35, 84]. These systems typically use codebook mapping techniques to map a training speaker's feature space into the testing speaker's feature space or the testing speaker's feature space into the training speaker's feature space. The training speaker is the speaker whose speech patterns are used to train the speech recognition system, while the testing speaker is a novel speaker whose speech has not been used in training. The system adapts model parameters based on the labelled speech samples from the testing speaker.

Unsupervised speaker adaptation does not require the speaker to speak any fixed words or sentences. Instead, the speaker's speech is monitored in real-time and used to adapt a recognition model. Since in supervised speaker adaptation, the data used in adaptation contain known words or phonemes, the reliability of the adaptation data is higher. For unsupervised speaker adaptation, the input speech data is not labelled, so less confidence can be placed on the adaptation data.

The difference in the confidence placed in the adaptation speech data also influences the degree of adaptation of the models. For example, supervised speaker adaptation systems frequently adapt codebook values or Gaussian mixture densities which contain upward of hundreds of parameters [22, 23, 29, 69]. On the other hand, unsupervised speaker adaptation frequently is performed on simple parameters such as one that controls the degree of spectral shift applied to input spectra [55].

Approaches that deal with speaker variability such as supervised adaptation and unsupervised adaptation described above require computation during the recognition process to adjust the system parameters. Also, the amount of improvement in performance is constrained by the limited confidence in the adjustment of the parameters. A new approach similar in spirit to the approach of obtaining more training data but that requires no collection of new training data is introduced in this thesis. Voice transformations have been performed to the training data to generate more variability in the training data and to improve the generalization ability of the wordspotting system after Figure of Merit training.

A graphical description of the benefit of voice transformation is shown in Figure 1-7. In this chart, the squares and the circles represent data points from two different classes. The two axes are two input features that contain information which separate the hits from the false alarms. As described in the previous section, the more data points available, the more accurate the estimated boundary between the two classes will be. Figure of Merit Training maximizes the evaluation criterion using the information gained from the limited data points. However, if the available data points are not completely representative of all possible data points, then the boundary generated through Figure of Merit training may

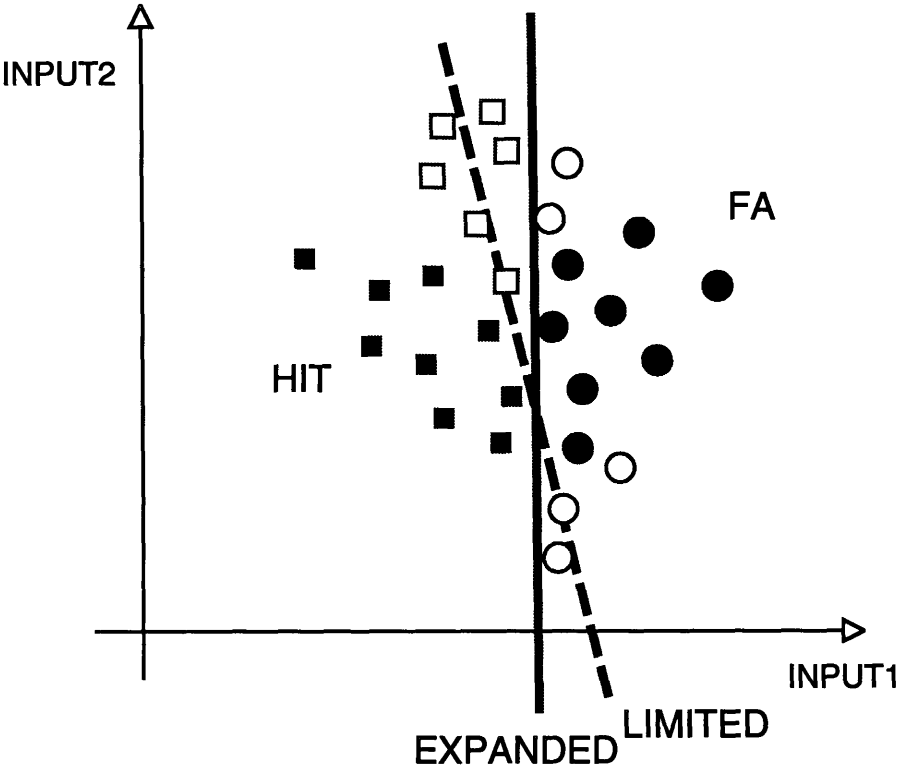


Figure 1-7: Generating more training data through using voice transformation.

overfit the training data points and be incorrect for unseen data points. By using *a priori* knowledge about the differences between data points in the same class, artificial data points (illustrated as hollow circles and squares) can be generated and used to lessen the danger of overfitting the training data. As seen in the figure, the boundary obtained from the expanded training set separates the *HIT* class from the *FA* class more reliably when unseen data points are taken into account.

1.5.1 Proposed Approach

People's speech differ in many different dimensions, such as formant frequency, fundamental frequency, glottal source characteristics, speaking rate, accent, and semantics. In this study, the focus is on performing voice transformation in the domain of formant frequencies. It is well known that formant frequencies are important cues to the identity of vowels in the English language [59]. Also, it is easier to characterize a person's formant frequencies than other characteristics such as accent and semantics. Finally, the amount of variability in formant frequencies in speech databases is better documented and thus more detailed studies are possible. Most current databases do not contain enough examples of accent regions or semantic styles to allow a complete study.

Wakita has shown previously that if a person's vocal tract is modelled as a series of lossless, acoustic tubes and if, for a given vowel, people's vocal tract shapes are similar except for the length of the vocal tract, then the formant frequencies identifying the vowel should vary linearly across different speakers [77]. Furthermore, the formant frequencies are inversely proportional to the vocal tract length. A method of changing a person's frequency scale is developed in this thesis. This method allows the spectral frequency scale of a speech recording to be expanded or contracted. The resulting speech recording can be listened to and evaluated on its naturalness.

Once a method of changing one voice to sound like a different voice is developed, it is used to generate more training speech artificially, thereby enlarging the training set and providing better coverage of the possible speakers who may use the wordspotter. Experiments have been conducted in training wordspotters with this enlarged training set to determine the efficacy of the voice transformation approach and results are presented in Chapter 6.

1.6 Thesis Overview

The rest of this thesis is outlined below: Chapter 2 presents the task, the database, and the evaluation metric used in this study. Chapter 3 describes the two fundamental technologies upon which the wordspotter is based: Hidden Markov Models (HMM) and neural network training techniques. By combining the strength of these two techniques, a high performance wordspotter was implemented. Chapter 4 describes the different stages of the wordspotter including pre-processing, wordspotter score calculation, gender detection, and post-processing. In chapter 5 the theory and the experimental results of applying Figure of Merit Training are presented. Chapter 6 illustrates the process of transforming a speaker's voice into different voices to increase the amount of variability in the data. To compare the effectiveness of the wordspotter to human performance, two human subjects participated in wordspotting experiments and the results are shown in Chapter 7. Chapter 8 summarizes the experimental results presented in the thesis, contrasts them to that of other approaches, and presents the main results and the conclusion of the thesis.

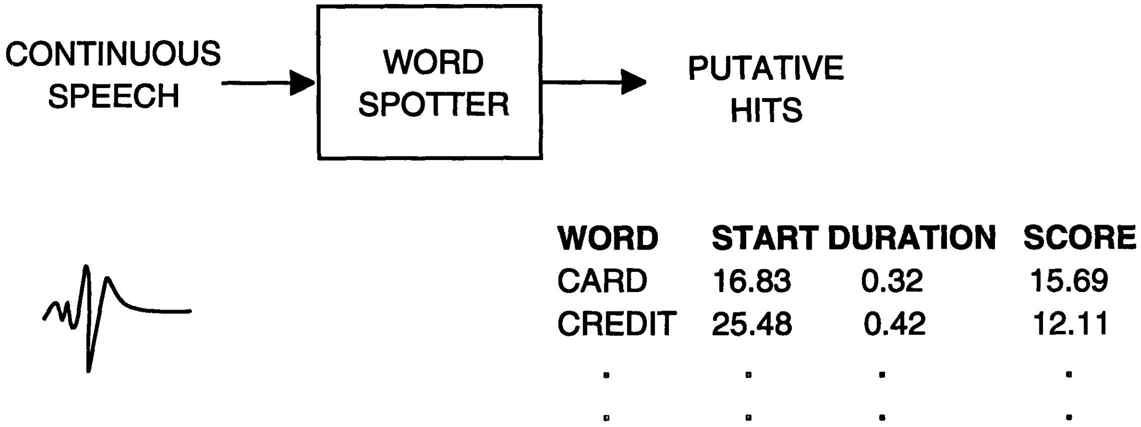
Chapter 2

Experimental Methodology

Wordspotting is a type of speech recognition problem that is very different from the more traditional transcription task. In this chapter, the experimental methodology used in this thesis is described. Section 2.1 defines the task of wordspotting. Section 2.2 explains the method used to evaluate a wordspotting system. Section 2.3 describes the database chosen for study in this thesis. The split of the database into training and testing sets is listed in Section 2.4. Section 2.5 introduces a separate database that was used by many sites to evaluate wordspotting performance.

2.1 Task

The task that the wordspotter is designed to perform in this thesis is detecting the occurrence of a pre-defined list of keywords in unconstrained, continuous speech. The basic process is described in Figure 2-1. The wordspotter system accepts continuous speech and generates hypotheses of where keywords occurred. The speech database consists of unconstrained, spontaneous conversations recorded on the telephone. The goal of the wordspotter is to spot the occurrence of a predefined set of keywords. For each conversation, the spotter generates a list of putative hits. Each putative hit is identified by the keyword, the starting time, the duration, and the confidence of the keyword's occurrence.



WORD	START	DURATION	SCORE
CARD	16.83	0.32	15.69
CREDIT	25.48	0.42	12.11
.	.	.	.
.	.	.	.

Figure 2-1: The basic function of a wordspotter is to accept continuous speech input and generate hypotheses, i.e. putative hits, of where keywords occur.

2.2 Evaluation Methods

Putative hits are compared to marking files provided by the National Institute of Standards and Technology (NIST) which record the actual keyword occurrences in the conversations. Certain keywords in the conversations are marked as bad examples when they were mispronounced or when the transcriber was not sure of what the speaker said [54]. When the wordspotter spots a bad example, the putative hit is not counted as a false alarm. However, the bad examples are not counted in measuring the detection rate of the wordspotter. Also, a keyword that appears as a part of another keyphrase, such as the word *card* in the phrase *credit-card*, is not counted as an error if spotted nor is it counted as a miss if it is not detected.

NIST has defined a scoring methodology to be used for evaluating wordspotters. The wordspotter processes files of recorded speech and generates hypotheses of word occurrences, called putative hits. Each putative hit consists of the following information:

- identity of the word,
- identity of the conversation,
- beginning time of the word,
- duration of the word,
- confidence in the occurrence of the word.

During scoring, each putative hit is compared to the marking file of the identified conversation. A putative hit is considered a *true hit* if the midpoint between the beginning and the end of its occurrence falls within the beginning and the end of the same word in the marking file. If the word identified by the putative hit did not occur, then the putative hit is considered a *false alarm*. Words that did occur in the conversation but that are not detected by the wordspotter are called *misses*.

To calculate a number that can concisely summarize the performance of a wordspotter, NIST has defined a measurement called *Figure of Merit*. The Figure of Merit is the average detection rate of the wordspotter over the range of zero to ten false alarms per keyword per hour. A program called *rocplot* has been implemented by the author as a part of this thesis. This program accepts as input a file of putative hits, compares each putative hit to the corresponding marking file, and generates the Figure of Merit for each individual conversation, for each keyword, and over all the conversations. The program also plots the detection rate over the false alarm rate of the wordspotter. Such curves are usually called Receiver Operating Characteristic (ROC) curves and examples of them can be found in this thesis (Figure 1-6 and Figure 8-2).

2.3 Database

The main data to be used for this research is the Switchboard Credit Card database collected by Texas Instruments and provided by NIST [26]. The Switchboard database consists of recordings of prompted telephone conversations about specific topics. The conversations are digitally recorded onto a personal computer through a direct T1 digital line to the long distance trunk line, thus by-passing the analog distortions from local exchange offices. The two sides of the conversations are recorded synchronously as two different files, but due to the hybrid four wire to two wire converters that exist in the telephone network, a recording of one side of a conversation will also contain information from the other side of the conversation; this effect is called *crosstalk*. The recordings dealing with the topic of credit cards have been chosen as the database to be used for wordspotting research. A total of 35 conversations are provided. The conversations were recorded telephone conversations and thus telephone characteristics such as distortion, limited bandwidth, crosstalk, and line noise are present. This database was chosen by Defense Advanced Research Projects Agency for evaluating wordspotting algorithms. Many sites have also used this same database to perform wordspotting research. Thus, this database was chosen for the research conducted in this thesis so that comparisons can be made with results from other sites. This database is relatively difficult due to the fact that it is spontaneous speech, it is recorded over the telephone, and it does not have a constrained vocabulary. More discussions on the differences between this database and other databases are provided in 7.4.1.

The twenty keywords chosen for this database are listed in Table 2.1. The number of times each keyword appears in the 35 conversations are also listed. The counts for the word *card* and the word *credit* include the occurrence of these words when they are part of the phrase *credit card.*

Table 2.1: List of 20 Keywords in the Credit Card Database

Keyword	Number of Instances
account	37
american express	49
balance	42
bank	56
card	633
cash	102
charge	135
check	114
credit	465
credit card	365
discover	29
dollar	99
hundred	41
interest	104
limit	34
money	112
month	122
percent	54
twenty	17
visa	76

2.4 Division of the Database

The database chosen for this thesis is the Switchboard Credit Card database available from NIST. The database is consisted of duplex telephone conversations recorded with different circuits and headsets. The recorded human subjects talked spontaneously about a pre-specified topic, in this case, credit cards. The official training set consists of 35 duplex conversations, resulting in 70 single-side recordings. The majority of this thesis uses a split in which the 70 conversations are divided into a training set and a testing set. The training set database is used to train the parameters of the wordspotting system, while the testing set is used to evaluate the performance of the wordspotting system on unseen data. Table 2.2 lists the training set recordings while Table 2.3 lists the evaluation set recordings. This division is the same one used by the group at Bolt Beranek and Newman (BBN) and allows for comparison of results [67].

The selected division includes 24 speakers of each gender in the training set and 11 speakers in the testing set. During the early part of the research, a high-performance gender detection classifier was developed that can reliably separate the genders. Thus separate gender-specific wordspotting systems are trained and the appropriate wordspotting system is chosen based on the output of the gender classifier. More details about the gender detection system are presented in Section 4.8.

The number of times each keyword occurs in the training and testing conversations are shown in Table 2.4. One can see that there is a wide range of frequency, ranging from 435 occurrences for the word *card* to 15 occurrences for the word *twenty*. When the number of occurrences for each keyword in each gender is counted separately, the number roughly halves. The scarcity of training data can also be measured in terms of the number of speakers that uttered each keyword. The number of speakers for each keyword is listed in Table 2.5. For certain words such as *discover* for males and *account* for females, less than five speakers out of 24 spoke those keywords. Such a small number of keywords is a problem in training the wordspotter because the keyword models trained from so few speakers are not general enough to spot the same keywords from other speakers. Such problems can be alleviated by performing voice transformation, to be described in Chapter 6.

Table 2.2: List of Male and Female Conversations in the Credit Card Database Used for Training

Male	Female
sw1026_a	sw2023_a
sw1037_a	sw2023_b
sw1037_b	sw2067_a
sw1044_a	sw2067_b
sw1044_b	sw2301_a
sw1060_a	sw2390_a
sw1060_b	sw2390_b
sw1083_a	sw2409_a
sw1083_b	sw2682_b
sw1088_a	sw2718_b
sw1088_b	sw2800_b
sw2301_b	sw2917_a
sw2313_a	sw2917_b
sw2313_b	sw2951_a
sw2399_a	sw2951_b
sw2399_b	sw2999_a
sw2409_b	sw2999_b
sw2536_b	sw3332_a
sw2718_a	sw3409_b
sw2764_b	sw3439_b
sw2987_a	sw3781_a
sw3751_b	sw3781_b
sw3821_a	sw3855_a
sw3821_b	sw3855_b

2.5 The NIST Official Testing Set

During September of 1992, NIST held an official benchmark for the credit card task. 10 new conversations were provided on a separate CD-ROM to be used as the official testing set. Throughout the research presented in this thesis, the official testing set was not used to evaluate wordspotter performance or speaker variance. However, at the conclusion of research, the official testing set was used in a spotting test to compare the performance of the wordspotter to the wordspotter from other sites such as BBN and SRI International. The results on the official testing set are presented in Section 8.1. The complete set of conversations from the database described above was used for training the wordspotter. The number of speakers and the number of keyword occurrences for each gender in the

Table 2.3: List of Male and Female Conversations in the Credit Card Database Used for Evaluation

Male	Female
sw1026_b	sw2163_a
sw1038_a	sw2163_b
sw1038_b	sw2681_a
sw1081_a	sw2710_a
sw1081_b	sw2710_b
sw2536_a	sw2883_a
sw2764_a	sw2883_b
sw2800_a	sw2987_b
sw3332_b	sw3170_a
sw3409_a	sw3170_b
sw3409_b	sw3751_a

training set database are listed in Table 2.6. The total sum for the training set is different from the total sum in Table 2.4 because in Table 2.6 the occurrences of the words *card* and *credit* that are a part of the phrase *credit-card* are not counted.

2.6 Chapter Summary

In this chapter the task that was studied in this thesis was presented. Also, the methodology used in scoring the performance of the wordspotter on the task was defined. Finally, the data chosen for this study was described. The database split presented was used in all subsequent experiments.

Table 2.4: The Number of Keyword Occurrences in the Male and Female Training and Testing Splits of the Credit Card Database (*card* and *credit* that occurred in *credit card* are counted as well)

	Train			Test		
Keyword	Male	Female	Both	Male	Female	Both
account	17	8	25	8	4	12
american express	19	18	37	8	4	12
balance	17	10	27	8	7	15
bank	19	12	31	19	6	25
card	219	216	435	124	74	198
cash	41	31	72	15	15	30
charge	42	56	98	20	17	37
check	44	35	79	27	8	35
credit	146	170	316	88	61	149
credit card	106	144	250	75	40	115
discover	9	10	19	2	8	10
dollar	42	32	74	14	11	25
hundred	17	7	24	14	3	17
interest	39	33	72	12	20	32
limit	8	14	22	4	8	12
money	44	42	86	12	14	26
month	38	45	83	22	17	39
percent	17	16	33	10	11	21
twenty	9	6	15	10	6	16
visa	26	24	50	17	9	26
Overall	919	929	1848	509	343	852

Table 2.5: The Number of Speakers Represented for Each Keyword in the Male and Female Training Splits of the Credit Card Database

Keyword	Male	Female
account	7	3
american express	8	8
balance	6	6
bank	14	6
card	24	24
cash	15	17
charge	16	16
check	16	15
credit	24	23
credit card	24	23
discover	4	7
dollar	15	12
hundred	9	5
interest	16	15
limit	5	8
money	17	19
month	16	15
percent	10	8
twenty	8	5
visa	11	13

Table 2.6: The Amount of Speech Available in the Credit Card Database for the Official Test (All 35 duplex conversations first provided by NIST are used for training, 10 new duplex conversations in the official testing set are used for testing.)

	OFFICIAL TRAIN			OFFICIAL TEST		
	Male	Female	Both	Male	Female	Both
Number of Speakers	35	35	70	13	7	20
Duration (Hours)	4.1	3.9	7.9	1.1	0.6	1.7
Keyword Occurrences	1066	896	1962	340	154	494

Chapter 3

Background

3.1 Introduction

In this thesis, a hybrid wordspotter composed of neural networks and hidden Markov models (HMM's) is developed. This chapter presents an introduction to both Hidden Markov Models and neural networks. Extensive references can be found on both topics. Readers interested in applications of these two techniques for speech recognition can find good introductions in the work by Morgan and Scofield and a recent book by Bourlard and Morgan [5, 49].

3.2 Hidden Markov Models

3.2.1 Introduction

Hidden Markov Models (HMM) have been used extensively in speech recognition to model the variabilities of the speech [2, 58, 62]. A good introduction to hidden Markov Models can be found in an article by Rabiner and Juang [62].

A Hidden Markov Model is a special type of Markov model. A Markov model is a set of states in which the probability of transition between different states depends solely on the identity of the current state [16]. A Markov model can be used to model a sequence of events.

In a HMM, each Markov state has an associated output probability distribution. Upon entering a Markov state, an output is generated according to the output distribution. The transition between Markov states is governed by the transition probability between states.

A common explanation is to think of each Markov state as a jar with different colored balls in the jar. Associated with each jar is a dice which determines which jar to use next. Upon drawing a ball from a jar, the next jar to withdraw the ball from is chosen by throwing the dice associated with the jar. The output probability distribution of the Markov state represents the probability of drawing a ball of certain color from the jar. The transition probability of each Markov state represents the probability of choosing the next jar by throwing the dice.

By starting from an initial jar, the methodology described above can be used to generate a sequence of colored balls, or observation vectors O. The sequence of jars that were used to generate the sequence of colored balls O is not obvious, hence the name *Hidden* Markov Model. In the speech domain, each colored ball can be thought of as a different phone, and the sequence of colored ball can be thought of as a sequence of phones. The sequence of states that are used to generate the observation vectors O is denoted as Q, $Q = q_1, \ldots, q_T$, where q_t denotes that the HMM system was in state q at time t.

A word can thus be modelled by a HMM model. Each word is modelled as a sequence of outputs from a collection of Markov states, with the transition from one Markov state to another governed by the transition probability between the Markov states. Such transitions are necessary because each word has variable duration. For example, the speaking rate of different speakers can vary by as much as 50%. Also, different individuals have different sounding phones. The Hidden Markov Models model each segment of the speech pattern. Also, the transition between segments of speech can be modelled as the transition between different states of the HMM.

A simple HMM modelling the word *card* is shown in Figure 3-1. In this simple example, each phone is modelled by a single Markov state. The parameters that define a HMM system are: a_{ij}, b_i, and π_i, where i and j are state indices. The transition probability between states is called a_{ij} by convention. Each a_{ij} is between zero and one and represents the probability of a transition from state i to state j given that current state is i. The sum of transition probabilities emanating from each state i, $\sum_j a_{ij}$, equals unity. Each state has an output probability distribution which describes the probability of a particular state generating a particular observation vector. The output distribution for each state is conventionally represented by b_i. Different states have different initial probabilities which are called π_i. Most HMM's used in speech recognition are modelled as a connected sequence

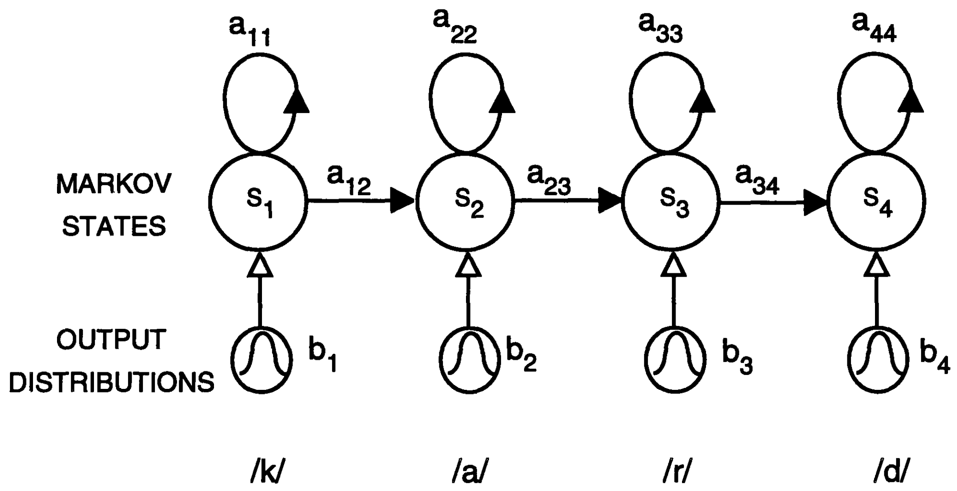


Figure 3-1: A Hidden Markov Model which models a consecutive sequence of phonemes in the word *card.*

of HMM states, thus the initial probabilities are typically one for the first state and zero for all other states.

Continuous HMMs, which use a mixture of Gaussian functions to model the output distributions of HMM states, are used in this thesis. The output probability distribution of each state is modelled using a set of Gaussian functions. Mathematically, the likelihood of generating an observation vector o from state i is:

$$P(o|s_i) = b_i(o) = \sum_{n=1}^{nmix} w_n \cdot N(o; c_{i,n}, \sigma_{i,n}), \tag{3.1}$$

where $nmix$ is the number of Gaussian mixtures in the output distribution, w_n is the weight of each Gaussian mixture function, and $N(o; c_{i,n}, \sigma_{i,n})$ is the Gaussian function evaluated at a point o with the mean of $c_{i,n}$ and the standard deviation of $\sigma_{i,n}$. This structure is identical to a type of neural network classifier called Radial Basis Function classifiers (RBF's) [50]. This similarity enables the training of the Gaussian mixture centers with a novel neural network training technique, called Figure of Merit Training, to be described in Chapter 5.

In this thesis, the observation vectors are generated every 10 milliseconds and consist of mel-scaled cepstra vectors (to be described in Chapter 4.) Given a set of labelled observation vectors O, $O = o_i, \ldots, o_T$, and a HMM model with parameters λ, with $\lambda \equiv A, B, \pi$, there are three problems that need to be solved for the HMM model to be used in a speech recognition system:

Evaluation How does one calculate the value $P(O|\lambda)$?

Matching How does one find a sequence of states Q so that $P(Q,O|\lambda)$ is maximized?

Training How does one find a set of parameters λ so that $P(O|\lambda)$ is maximized?

3.2.2 Evaluation

Assume that for a set of words W to be recognized, each word w has been modelled by a HMM model with parameters λ_w. Then given a set of observations O, a simple speech recognizer would compare the likelihood of the observation O having been generated by each word's HMM model. The model which has the highest likelihood, i.e. $P(O|\lambda_w)$, would be the most likely word spoken assuming that all words have equal prior probability. This result is derived from the Bayes rule:

$$P(\lambda_w|O) = \frac{P(O|\lambda_w)P(w)}{P(O)} \tag{3.2}$$

Since $P(O)$ is the same for all word hypotheses, with equal $P(w)$,

$$maxP(\lambda_w|O) = maxP(O|\lambda_w). \tag{3.3}$$

The algorithm used for evaluating the function $P(O|\lambda_w)$ is called the forward-backward algorithm [62].

3.2.3 Matching

As mentioned before, a simple isolated-word recognizer can be constructed by comparing the likelihood of each model generating the observations, denoted as $P(O|\lambda_w)$. But for continuous speech recognition, particular sequences of word occurrences are desired. In this case the Viterbi algorithm [62] can be used to find a sequence of states through an HMM model which is most likely to have generated the observations, i.e. $P(Q,O|\lambda_w)$. Figure 3-2 describes matching the states of an HMM model to a sequence of frames. The allowable transition from one state to the next state is constrained by the transition probabilities between states. For example, state 1 can either transit back to itself or to state 2. The likelihood of each state's output distribution generating a particular input frame is summed along a sequence of possible transitions between states. The possible sequences that match

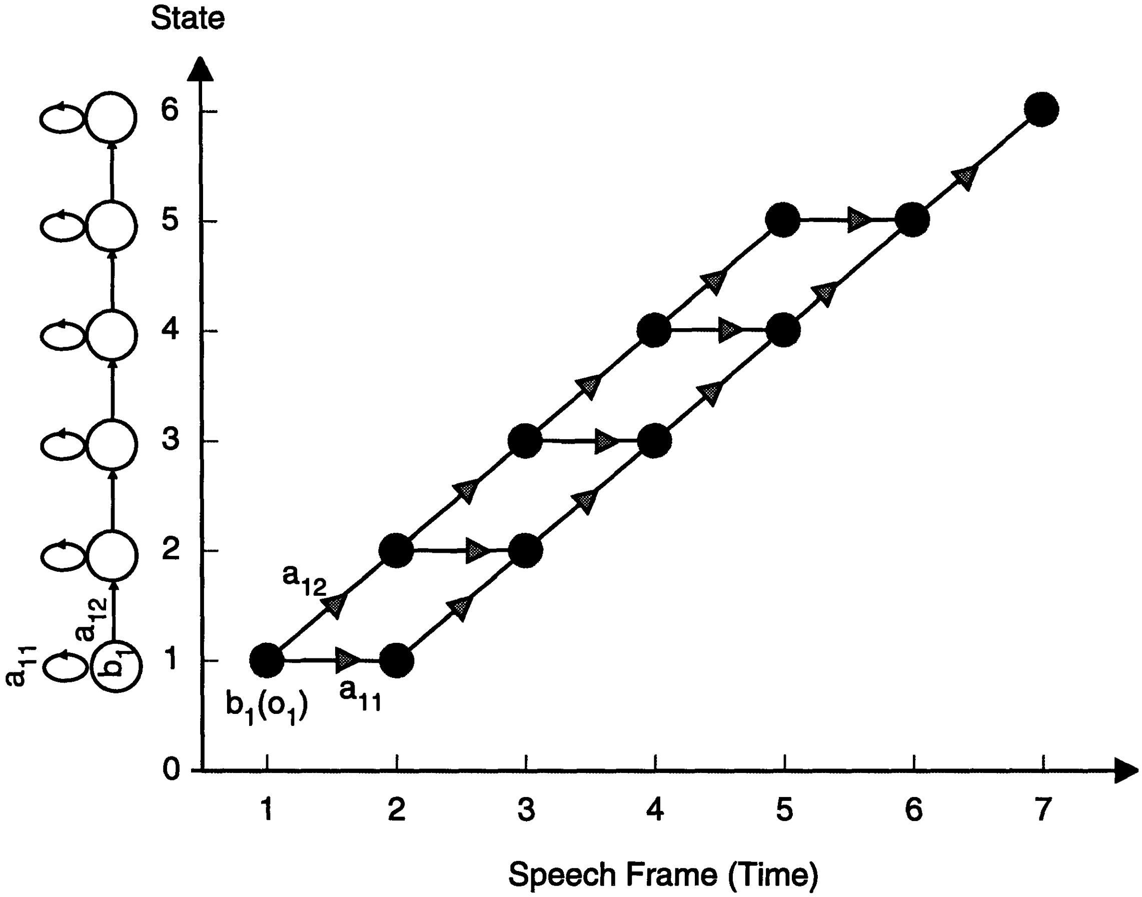


Figure 3-2: The Viterbi algorithm matches HMM states to input frames using a dynamic programming approach.

the input frames to the states in the HMM model are searched using dynamic programming technique so that the most likely path is found. This technique is used in Chapter 5 to map input frames to states in a keyword HMM model.

3.2.4 Training

The goal of training is to adjust the HMM model parameters λ so that the likelihood $P(O|\lambda)$ is maximized. For the isolated word case, the examples of each keyword are collected as a set of observations. The Baum-Welch reestimation algorithm would then be used to iteratively adjust the model parameters [62]. The algorithm conceptually works by first calculating the probability of each observation at time t having been generated by a particular state in the HMM, then the parameters of each state are adjusted by taking the weighted average of the observations assigned to each state. The algorithm is guaranteed to improve the likelihood $P(O|\lambda)$ on a given set of data and is typically run until the change in $P(O|\lambda)$ is very small.

3.3 Neural Networks

Neural networks have been used extensively in the speech recognition domain with applications ranging from phonetic recognition [76] to large vocabulary recognition [64]. A good overview of using neural networks in pattern classification can be found in [41]. The application of neural networks to speech recognition has also been extensively studied. An overview can be found in [42].

3.3.1 A Radial Basis Function Classifier

One type of neural network classifier that is related to the wordspotter used in this thesis is called the Radial Basis Function (RBF) classifier. The RBF classifier utilizes localized basis functions for constructing the boundary between different classes in the input space [41]. In Figure 3-3, a basic radial basis function classifier is shown. The output of the classifier is defined by the following equation:

$$output(x) = \sum_{i=1}^{ncenter} w_i \cdot N(x; m_i, \Sigma_i), \tag{3.4}$$

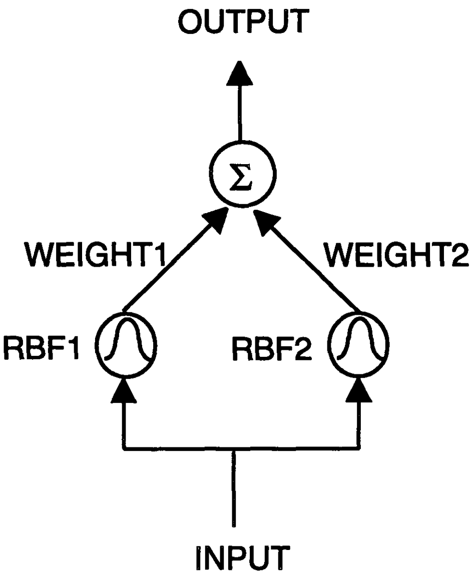


Figure 3-3: A basic radial basis function classifier.

where x is the input pattern, *ncenter* is the number of RBF centers, w_i is the weight linking each RBF center to the output node, and $N(x; m_i, \Sigma_i)$ is the Gaussian function with the mean m_i and the covariance matrix Σ_i that is used in the radial basis function. Assuming that the input x is a vector of dimension N and that the Gaussian functions have diagonal covariance, then Equation 3.5 defines the Gaussian function:

$$N(x; m_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^N \mid \Sigma_i \mid}} e^{-\frac{1}{2}(x-m_i)^T \Sigma_i^{-1}(x-m_i)}. \tag{3.5}$$

The most common training algorithm of RBF classifiers is a combination of supervised and unsupervised training. The mean m_i and the covariance matrix Σ_i are estimated by clustering the training data without supervision. Then the weights w_i are trained to minimize the squared difference between the desired output and the classifier output. A complete algorithm description can be found in [50]. By training the means m_i and the covariance matrix Σ_i in an unsupervised manner, the weights can be estimated directly using linear algebra techniques very efficiently [50]. However, another method for training the parameters is to incrementally train them. Since the incremental algorithm for training the means is related to the approach used in this thesis to train the wordspotter, it is described here. Assume that the output of the classifier has a target value t_p on an input pattern x_{np}, where $n = 1, \ldots, N$, N being the dimension of the input pattern, $p = 1, \ldots, npattern$,

npattern being the total number of patterns. Let the error be defined as the sum of squared differences between the output and the target signal over all patterns x_p:

$$E = \frac{1}{2} \sum_{p=1}^{npattern} (t_p - out_p)^2, \tag{3.6}$$

where p is the pattern number. The error function E can be minimized by adjusting the parameters $m_{i,n}$ once the gradient $\frac{\partial E}{\partial m_{i,n}}$ is derived. The expression $\frac{\partial E}{\partial m_{i,n}}$ can be expressed as:

$$\frac{\partial E}{\partial m_{i,n}} = \sum_{p=1}^{npattern} \frac{\partial E}{\partial out_p} \cdot \frac{\partial out_p}{\partial m_{i,n}}. \tag{3.7}$$

To minimize E, the parameters will be modified in the direction of the negative gradient of E, i.e., $-\frac{\partial E}{\partial m_{i,n}}$. The terms of Equation 3.7 are extended below for a given pattern x_p:

$$\frac{\partial E}{\partial out_p} = out_p - t_p, \tag{3.8}$$

the term $\frac{\partial out_p}{\partial m_{i,n}}$ is calculated by taking the derivative of the Gaussian function in Equation 3.5:

$$\frac{\partial out_p}{\partial m_{i,n}} = w_i \cdot N(x_p, m_i, \Sigma_i) \cdot \frac{(x_{n,p} - m_{i,n})}{\sigma_{i,n}} \tag{3.9}$$

To reduce the error E, the $m_{i,n}$ are adjusted along the negative gradient. So for each input pattern x_p, the Gaussian centers are changed according to Equation 3.10.

$$\begin{aligned} m_{i,n,t+1} &= m_{i,n,t} - \eta \cdot \frac{\partial E}{\partial m_{i,n}} && (3.10) \\ &= m_{i,n,t} - \eta \cdot (out_p - t_p) \cdot \frac{\partial out_p}{\partial m_{i,n}} && (3.11) \\ &= m_{i,n,t} - \eta \cdot (out_p - t_p) \cdot w_i \cdot N(x_p, m_i, \Sigma_i) \cdot \frac{(x_{n,p} - m_{i,n})}{\sigma_{i,n}} && (3.12) \end{aligned}$$

The constant η is the learning rate used during training. With a large η, larger changes are made to the model parameters at each training pattern while a smaller η allows smaller changes to be made to the model parameters at each training pattern. When the learning rate η is set too high, the model parameters may diverge from the direction toward the lowest error. Thus η is usually set to a constant that is not large enough to cause divergence but still large enough to allow significant training. Typically a reasonable η is chosen through a series of preliminary experiments.

The radial basis function equations described in this section are the ones used to train the radial basis function classifier in Figure 1-5. Since the goal is to minimize the classification error rate, the Gaussian centers were moved to adjust the boundary between the two classes until classification error rate was minimized. In comparison, the Gaussian centers of a Gaussian classifier are estimated from the training set patterns of each class with no emphasis placed on classification error rates.

In Chapter 5, a related set of equations are derived which adjusts the parameters of the wordspotter to maximize the Figure of Merit.

3.4 Chapter Summary

In this chapter, Hidden Markov Models and Radial Basis Function classifiers were introduced. The HMM representation has been successfully applied to many speech recognition tasks and is chosen for its flexibility in modelling the temporal variation of the speech signal. The Radial Basis Function classifier is a type of neural network classifier that can be trained using a discriminative training technique. The benefit of the discriminative training technique is that in the case of limited knowledge about the distribution of the training data, the neural network classifiers can perform better than other classifiers that are based on maximum likelihood estimation. The strengths of these two methodologies are combined in implementing the wordspotter. The HMM is used to model the different duration of keyword occurrences, and the neural network training approach is utilized to improve the discrimination between true hits and false alarms.

Chapter 4

Baseline Wordspotter

4.1 Introduction

Before the novel techniques of Figure of Merit training and voice transformation were explored, a high performance baseline wordspotter was developed. This chapter describes the basic building blocks of the wordspotter. Section 4.2 describes the stages of transforming the input waveform into feature vector frames. In Section 4.3, the basic structure of the wordspotting system is described. Section 4.4 discusses the training algorithms used to train the baseline wordspotter. Section 4.5 introduces two different methods studied to generate putative hits from the wordspotter. Section 4.6 presents a change to the wordspotter structure to ensure that all states contribute to the spotting task. Section 4.7 illustrates the steps that are performed to the output of the wordspotter to remove unwanted artifacts. Section 4.8 describes the design and implementation of a gender detection system.

4.2 Preprocessing

The analog speech waveforms used in this thesis had been digitized at 8000 samples per second and stored on a CD-ROM provided by NIST. There are many different methods of extracting speech features from the waveform, such as calculating the spectrum, the cepstrum, and using physiologically motivated preprocessing stages [33, 46, 63, 71]. Rabiner and Schafer present a thorough explanation of the calculation of the spectrum and the cepstrum [63]. The processing stages used and described below have provided state of the art performance on many domains [58, 66]. At the end of the preprocessing stage, the original digitized speech waveform is transformed into mel-scaled cepstra frames (described in Section 4.2) that are used as the input to the wordspotter. The mel-scaled cepstra representation [12] has been successfully applied in Lincoln Laboratory projects in speaker identification [65] and continuous speech recognition [58]. A block diagram of the processing steps taken is shown in Figure 4-1.

Mel-Scaled Filter Bank

Most current speech recognizers use input features which describe the spectral envelope of the input waveform. Many possible input features have been explored, such as linear predictor coefficients, spectral magnitudes, cepstral values, physiologically motivated features, etc. In this study, mel-scaled cepstral coefficients were chosen as the input feature because experiments have shown that they perform well in other speech tasks [33, 46]. The first steps in the calculation of the mel-scale cepstral coefficients are as follows:

- Window the incoming speech with a Hamming window that is 20 milliseconds long and separated at 10 millisecond intervals [56].

- Compute squared spectral magnitudes of the signal for each 20 msec interval through a Fast Fourier Transform (FFT) operation.

- Multiply the squared spectral magnitudes with a pre-emphasis filter which emphasizes the spectral magnitudes at high frequencies. The filter constants are:

$$preemp(f) = 1 + \frac{f^2}{250,000}, \tag{4.1}$$

  where f is the frequency, and $preemp(f)$ is the constant that is multiplied to the squared spectral magnitude at frequency f.

- Sum the squared spectral magnitudes into 24 triangular filter bank values. The triangular filters are spaced linearly from 0 to 1000 Hz with the bank center frequencies spaced 100 Hz apart at frequencies of 100 Hz, $\cdots$, 1,000 Hz. Filter banks above 1,000 Hz are spaced logarithmically apart, with the center frequencies increasing at a rate of 1.1 times the previous center frequency value. The center frequencies for filter banks 11 to 24 are 1,100 Hz, 1,210 Hz, 1,331 Hz, 1,464 Hz, 1,611 Hz, 1,772 Hz, 1,949 Hz, 2,144 Hz, 2,358 Hz, 2,594 Hz, 2,853 Hz, 3,138 Hz, 3,452 Hz, and 3,798 Hz respectively.

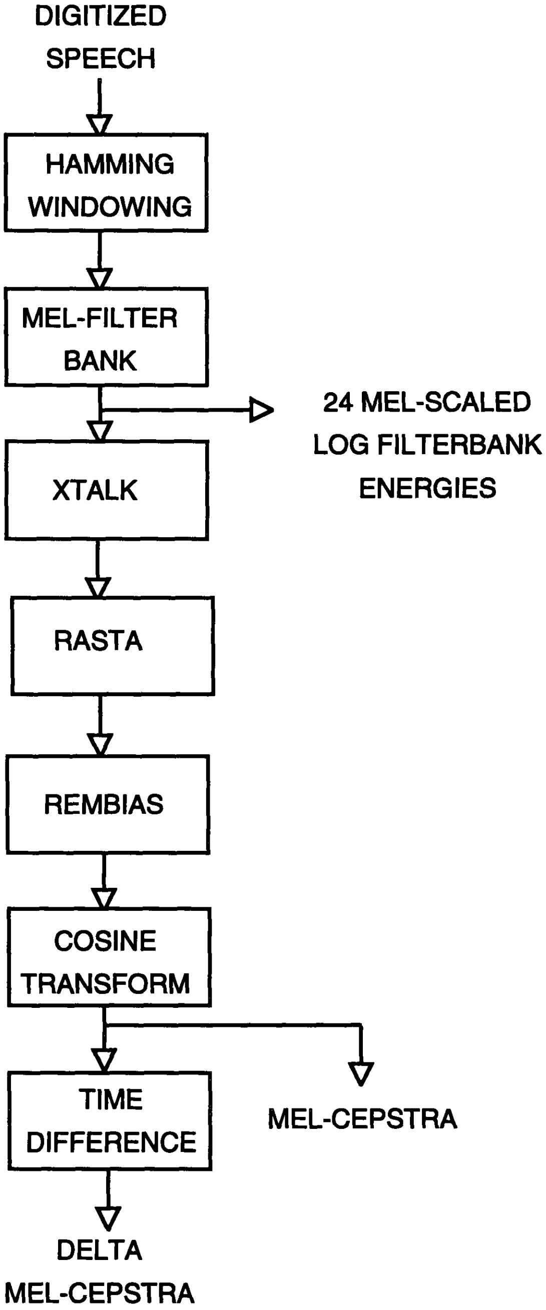


Figure 4-1: A block diagram of the preprocessing stages which transformed digitized speech into mel-scale cepstra frames that are used as input to the wordspotter.

The beginning and the end of the range of each filter bank n are the center frequencies for the filter bank $n-1$ and the filter bank $n+1$ respectively. The overall sum for each filter bank is normalized to reflect the different bandwidths of the filter banks and represents the energy in each filter bank. Finally the log of the filter bank energies are taken and the 24 mel-scaled log filter bank energies form a mel-scaled filter bank frame, denoted as $mfb(t)$.

Crosstalk Detection

A program called *xtalk*, developed by Doug Reynolds, was used to mark the mfb frames as active or silent [65]. The program can either be run using one-channel mode or two-channel mode. In the one-channel mode, the program determines if a mfb frame is active or silent by comparing the total energy in the frame, denoted as $s[n]$, to an adapted lower energy contour $c[n]$. The adapted lower bound $c[n]$ is calculated by taking the maximum of $c[n-1]$ and $s[n]$ at each frame and adding a small ϵ to $c[n]$. Whenever $s[n]$ is greater than $c[n]$ by a threshold, the mfb frame n is marked active. The normalized energy $t[n]$ is calculated for each frame by subtracting a high energy contour $f[n]$ from the frame energy $s[n]$. The high energy contour $f[n]$ is calculated by taking a minimum of $f[n-1]$ and $s[n]$ at each frame and subtracting a small ϵ at each frame. The normalized energy $t[n]$ allows accurate comparison of the speech amplitude. After all the frames have been labelled, a finite state program is used to merge locally labelled frames into silent and active regions.

When the program is operating in the two-channel mode, the normalized energies $t[n]$ for the two channels are compared and the channel with the higher $t[n]$ is marked as active while the other channel is marked as inactive.

Short Term Channel Normalization

The RASTA algorithm was introduced by Hermansky et al. to remove the influence of the frequency response of the communication channel [27]. Recorded telephone speech can be affected by many types of distortions in the communication channel. For example, different telephone handset microphones have different frequency response characteristics. Also, the transmission path from the speaker's handset to the central switch can also be affected by distortions.

The RASTA algorithm filters the mel-scaled log filter bank energies with a digital filter described by the following system equation:

$$H(z) = 0.1 \cdot \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{z^{-4}(1 - 0.98z^{-1})}. \tag{4.2}$$

Each log filter bank energy is independently filtered with digital filters defined by Equation 4.2. The filter shown above is a high-pass filter, thus rapidly changing signals are passed through while slowly varying signals are filtered out. Since speech signals change relatively quickly in comparison to channel characteristics, the channel characteristics are filtered out while the speech signal is still retained. The RASTA program implemented at Lincoln Laboratory by Marc Zissman was used in this thesis.

Spectrum Normalization

While the RASTA algorithm performs short-term filtering of the filter bank values, a spectral mean subtraction method can be used to compensate for long term frequency response of the channel. Spectrum normalization has been successfully applied for compensating long term spectral distortions [48]. The long term average of filter bank energies for each conversation is first calculated, then the average values are subtracted from the corresponding channel. Three different methods of running the spectrum normalization program developed by Doug Reynolds at Lincoln Laboratory, called *rembias*, were explored. In the first method, all incoming mfb frames are used for computing the long term average and the average values are subtracted from all incoming mfb frames. In the second method, the mfb files have been processed with *xtalk* in the one-channel mode. The long term average is calculated only from mfb frames that were marked as active and the long term averages were subtracted only from the mfb frames marked as active. The third method is similar to the second method except that the active labels were derived by running *xtalk* in the two-channel mode.

Conversion to Cepstral Coefficients

After RASTA filtering and spectrum normalization are performed, the resulting mel-scaled log filter bank energies are transformed to cepstral coefficients through a discrete cosine transform. Let $mfb(n)$, $n = 1 \cdots 24$, be the 24 mel-scaled filter bank values, then 24 cepstral coefficients are calculated with the following cosine transform:

$$mcc(i) = \frac{1}{24}\sum_{n=1}^{24} mfb(n) \cdot cos(i(n - \frac{1}{2})\frac{\pi}{24}). \tag{4.3}$$

Using only low order cepstra coefficients has been shown empirically to provide good performance on speech recognition tasks [33, 46]. The cepstra values with high indices, i.e., high *quefrency* [56], contain pitch information [63] and thus are not used to avoid the large variability that exists in speech pitch contours. In this work, high quefrency mel-scaled cepstral coefficients, $mcc(n), n > 12$, are not used. Also, since different people speak at different volumes, $mcc(0)$, which measures the overall energy of each frame, is not used to avoid the possibility of recognizing speech based on overall amplitude.

Temporal changes in spectral patterns across time are another important cue to the identity of phonemes being spoken and have been previously applied as a part of the frontend in continuous speech recognition [37]. In this work, delta mel-scaled cepstral values are used as another set of input features. The delta mel-scaled cepstra values are calculated by taking a first difference of the mel-scaled cepstra values between adjacent frames. Since the relative loudness of each phoneme can provide clues to the identity of the cues, the delta mel-scaled cepstra values are calculated from $mcc(0)$ to $mcc(12)$, resulting in a delta mel-scaled cepstra vector of length 13.

4.2.1 Experimental Results

A series of experiments was performed to evaluate the effectiveness of the pre-processing algorithms. The sequence of operations is as follows:

1. Mel-scaled spectrum calculation,
2. Crosstalk detection with *xtalk*,
3. Short-term channel normalization with *rasta*,
4. Long term mean removal with *rembias*.

As mentioned previously, the *xtalk* algorithm can be used in the one-channel mode as a speech activity detector or in the two-channel mode as a crosstalk detector. Both modes were evaluated in the experiments. Table 4.1 tabulates the experimental results. The FOM score reported was generated with a preliminary wordspotting system described in [40]. The combination of using *xtalk* in one-channel mode, *rasta*, and the *rembias* provided the highest FOM over all combinations.

Table 4.1: Comparison of Spotting Results Using Different Preprocessing Steps (xtalk1 denotes running *xtalk* in the one-channel mode, xtalk2 denotes running *xtalk* in the two-channel mode.)

Preprocessing Steps	FOM
no preprocessing	22.4%
rasta_rembias	23.9%
xtalk1_rembias	22.2%
xtalk1_rasta	22.9%
xtalk1_rasta_rembias	24.5%
xtalk2_rembias	22.9%
xtalk2_rasta	19.1%
xtalk2_rasta_rembias	22.3%

Another set of isolated word recognition experiments was performed to assess the benefit of using preprocessing steps. Simple hidden Markov models were trained to model the 20 keywords in the database. The silence segments and filler speech segments were also separately modelled with individual HMM models. The number of states for each keyword was varied between a fixed number of ten states to a variable number proportional to the number of phonemes in each keyword. For the case when the keywords were modelled with variable numbers of states, filler speech and silence were modelled with one state models with sixteen Gaussian mixtures per state. The word recognition rates are shown in Table 4.2. The importance of performing normalization is clearly demonstrated by the experimental results, as the trials with preprocessing in all three HMM configurations all outperformed the similar experiments without the preprocessing on the testing set. For the experiments in the remainder of this thesis, the combination of *xtalk* in one-channel mode, *rasta*, and *rembias* was used.

4.3 A Hybrid Architecture

The baseline wordspotter includes 20 left-to-right HMM word models, a single-state HMM for filler speech, and a single-state HMM for silence. The number of states in HMM keyword

Table 4.2: Comparison of Isolated Word Recognition Rates Using No Normalization vs. Using Selected Parameters (Under # of States, variable/1 means that a variable number of states were used for the keyword models and one state was used for the filler and silence models. Under # of Mixtures, 1/16 means that one mixture was used for the keyword models and 16 mixtures were used for the filler and silence models.)

Preprocessing	# of States	# of Mixtures	Training	Testing
no preprocessing	10	1	82.80%	76.89%
no preprocessing	10	3	91.42%	82.89%
no preprocessing	variable/1	1/16	86.37%	81.63%
xtalk1_rasta_rembias	10	1	87.09%	84.47%
xtalk1_rasta_rembias	10	3	93.31%	85.51%
xtalk1_rasta_rembias	variable/1	1/16	89.20%	86.36%

models is roughly 1.5 times the number of phonemes in the keywords. The resulting number of states is small enough so that most short examples of keywords can be used for training. States in keyword models use unimodal Gaussian distributions unless there are sufficient training examples to allow Gaussian mixture distributions. The single states in the filler and silence models have Gaussian mixture distributions with 32 mixture components.

All HMM models are initially trained using short segments of credit-card conversations containing keywords, silence, or non-keyword speech extracted from the original credit-card conversations using the NIST text files and time markings. The HMM models were trained and tested using programs in the HTK toolkit [80]. The Figure of Merit training algorithm, to be described in Chapter 5, was then implemented by the author as an added functionality to the HTK toolkit.

4.4 Training of the baseline wordspotter

There are three stages to the training of the baseline wordspotter system. The stages are:

1. Initialization,
2. Word Level Estimation,
3. Sentence Level Estimation.

During initialization, the keyword examples in the training conversations were excised from the conversations. Also, one second of silence and filler speech were also excised from each conversation. The keyword model of each particular keyword is initialized by segmenting the speech frames of all the examples of that particular keyword into the n equal portions, n being the number of states in the keyword model. Then each state's model parameters were estimated using just the portion of speech assigned to that state.

During word level estimation, the speech frames for a particular keyword were used jointly to perform expectation-maximization (EM) of the output likelihood of the hidden Markov model parameters. Instead of using the fixed portions of speech frames to estimate the parameters of each state, the EM algorithm automatically utilizes all speech frames to estimate all state parameters.

During sentence level estimation, also called embedded reestimation in [80], the original conversations were spliced into sentence length segments. The labels of each sentence contained only the symbols "silent," "filler," or keyword names, for example, *"silence filler filler card filler"*. The HMM models of keywords that appeared in the label were linked in the appropriate sequence and all state parameters were jointly estimated. This final stage has the benefit of not relying on the sometimes inaccurate word labels and temporal alignment provided by NIST. Since all the model parameters were jointly estimated, the EM algorithm can automatically incorporate speech frames for particular keywords that were left out of keyword segments during the initial training process. Ten iterations of sentence level estimation were performed in all experiments in this thesis. Figure 4-2 shows the average per-frame log probability versus the number of iterations of sentence level estimation. It is clear that by the tenth iteration the average log probability has converged.

4.5 Generating Putative Hits

4.5.1 Viterbi Scoring

The output of the initial baseline wordspotter is a list of symbols consisting of keyword names, and the symbols "filler" and "silence." An example of such sequence which includes the start and stop frames of each detected segment is shown in Figure 4-3. This transcript, obtained through Viterbi decoding, contains one putative hit for the keyword *card* embedded in filler and silence segments. Any symbol that is a keyword name is taken as a putative hit at a time determined from the Viterbi start and end times of the symbol. The output score was calculated by taking the difference between the end state log likelihood and the

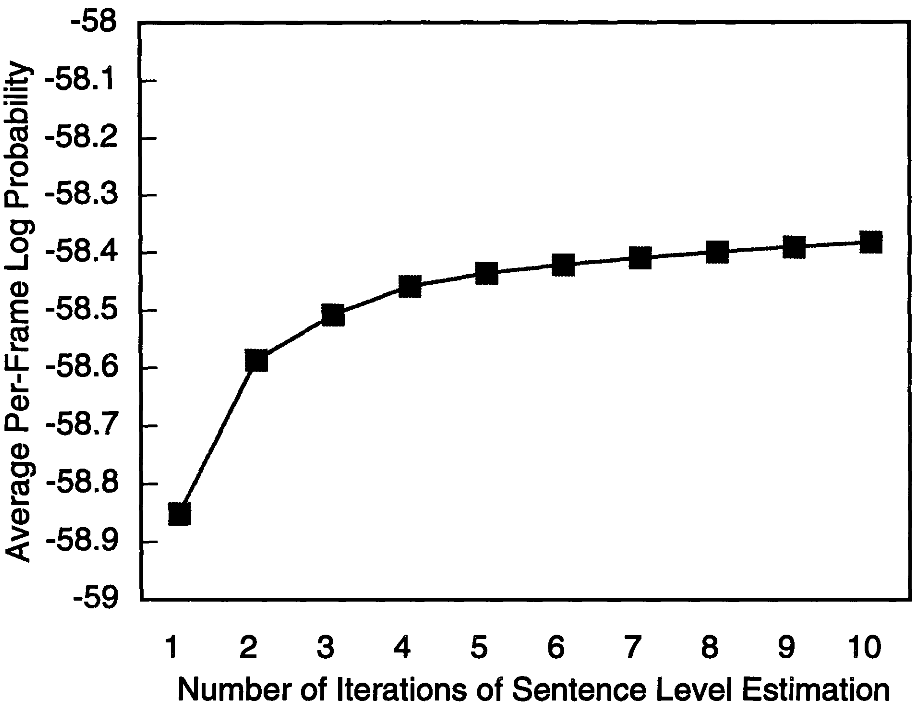


Figure 4-2: The average log probability per frame through ten iterations of sentence level estimation.

beginning state log likelihood of the keyword model and normalizing it by dividing by the length of the putative hit in frames. The performance of this initial baseline HMM wordspotter was poor. It provided an overall FOM of 16.9% when trained using the 48 training conversations and tested on the remaining 22 conversations in the testing set.

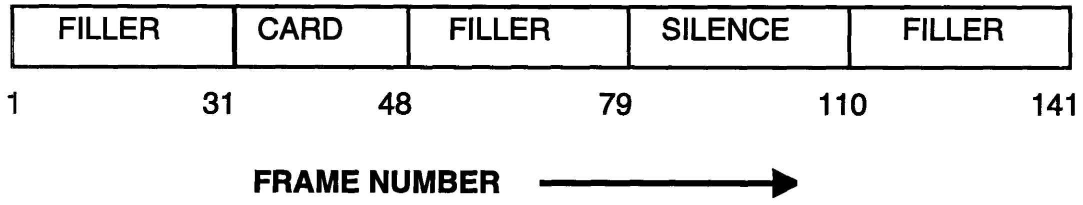


Figure 4-3: Transcription generated through performing Viterbi decoding and matching keyword models to the input speech.

4.5.2 Peak-Picking

The putative hits generated using the Viterbi scoring method suffer from the fact that only one keyword hypothesis can exist at any given time. For example, suppose that the word *charged* occurs in the conversation. Using Viterbi decoding, either the word *charge* or the word *card* may be in the output transcript, but not both at the same time period. Another method of detecting keywords is to independently calculate the probability of having a keyword occurring at a certain time. Let $P(o_t|\lambda_{keyword})$ be the probability of generating the observation o from the model $\lambda_{keyword}$ at time t and let $P(o_t|\lambda_{filler})$ be the probability of generating the observation o from the model λ_{filler} at time t. Using the Bayes rule, the probability of a keyword's occurrence at time t, $P(keyword|o_t)$ is the following:

$$P(keyword|o_t) = \frac{P(o_t|keyword)P(keyword)}{P(o_t)} \tag{4.4}$$

To simplify calculation, the Viterbi end state likelihood $P(o_t|\lambda_{keyword})$ is used to approximate $P(o_t|keyword)$ and the filler end state likelihood $P(o_t|\lambda_{filler})$ is used to approximate $P(o_t)$. Since $P(keyword)$ is assumed to be constant through out the conversation, it is dropped out. The probability of keyword occurrence $P(keyword|o_t)$ is then calculated by:

$$P(keyword|o_t) = \frac{P(o_t|\lambda_{keyword})}{P(o_t|\lambda_{filler})}. \tag{4.5}$$

Since the probabilities are very small, they are represented logarithmically during computation. Equation 4.5 written in log is:

$$log(P(keyword|o_t)) = log(P(o_t|\lambda_{keyword})) - log(P(o_t|\lambda_{filler})). \tag{4.6}$$

The putative hits are detected using frame-by-frame word scores. The score for each keyword model is calculated as the Viterbi log likelihood normalized by subtracting the Viterbi log likelihood of the filler model. Putative hits are generated by independently detecting peaks from the normalized output of each word model. Figure 4-4 shows a diagram of the architecture of the baseline HMM wordspotter. In this figure, only one keyword model is shown. The end state Viterbi log likelihood of the filler model is subtracted from the end state Viterbi log likelihood of the keyword model. Figure 4-5 shows the end state log likelihood for the *filler* model and the *card* model. The end state log likelihood is the probability of observing the particular input frame x being generated by the end state of the HMM model. In the wordspotter, all 20 keyword models have outputs normalized in the manner described above.

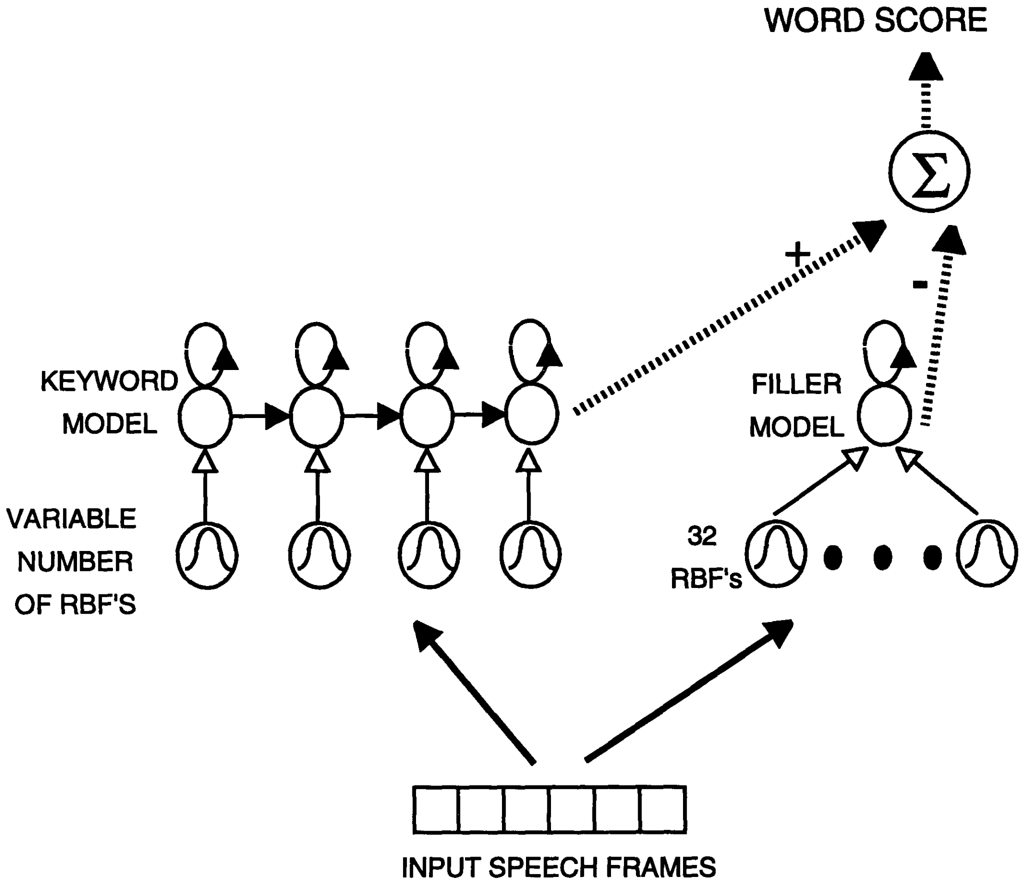


Figure 4-4: The generation of keyword score using the hybrid wordspotter composed of radial basis functions and hidden Markov states.

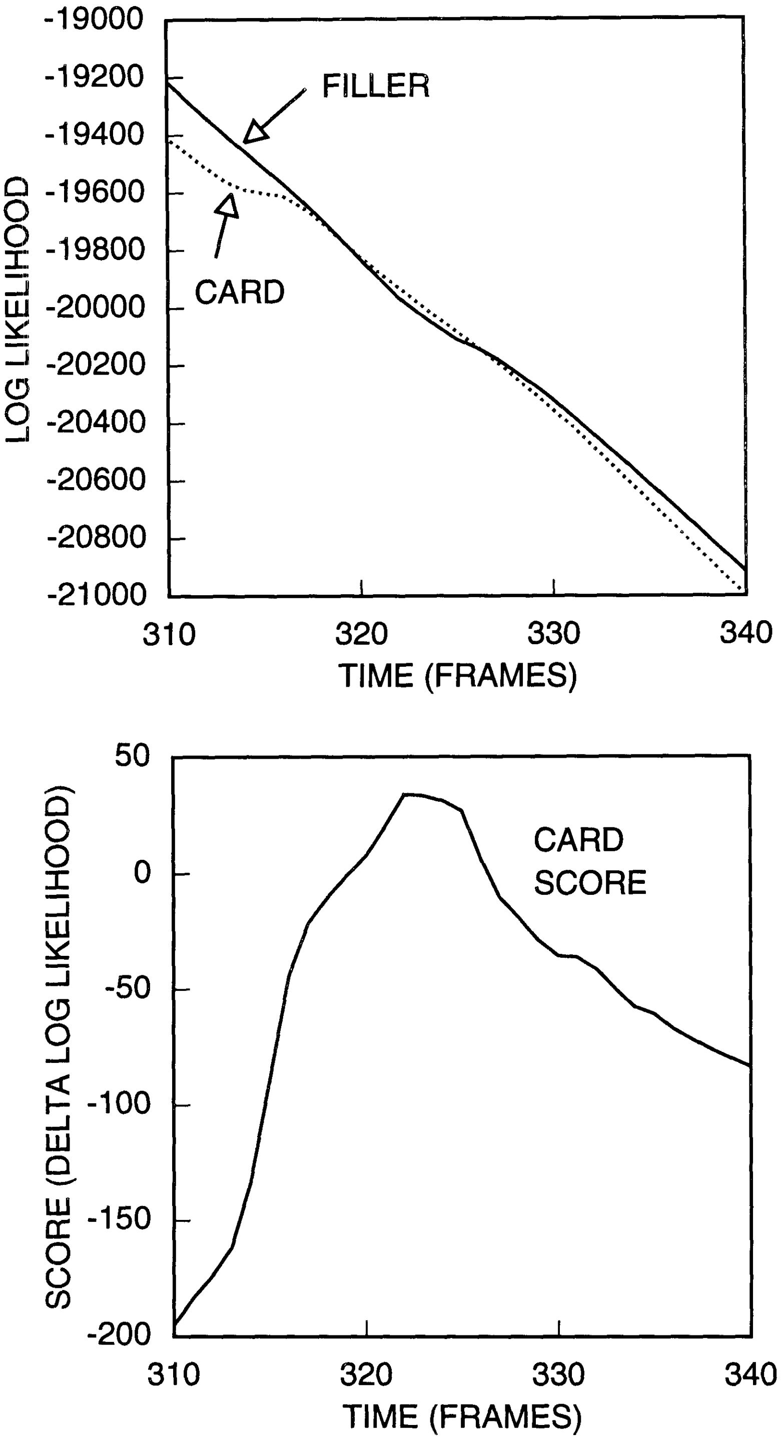


Figure 4-5: End state log likelihood of the *card* HMM model and the *filler* model are shown in the top plot, the difference between the two log likelihoods is the output score for the word *card* and is shown in the bottom plot.

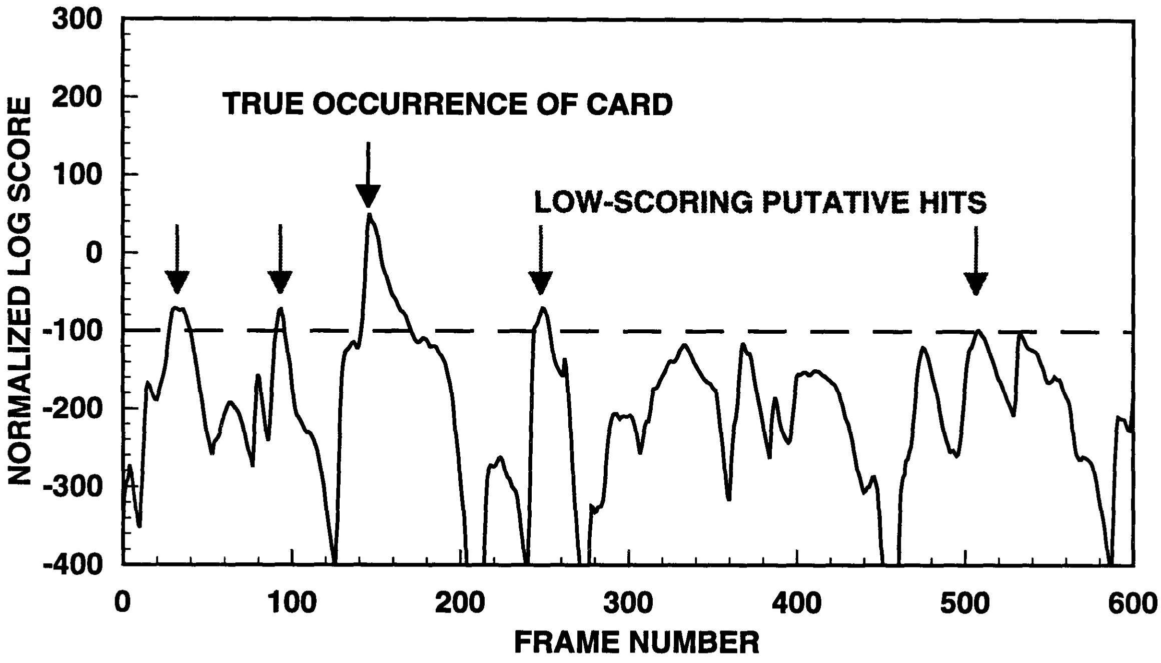


Figure 4-6: End state log likelihood of the *card* HMM model normalized by the end state log likelihood of the *filler* model.

Figure 4-6 shows an example of the normalized scores for the word*card* over six seconds of speech. Peaks in these traces above a threshold of -100 were taken as putative hits for this word and used to generate per-word and overall FOM scores. In this trace, the true occurrence of *card* ending at frame 146 generates a strong high-scoring putative hit and other words generate four low-scoring putative hits. This peak-picking approach increased the overall FOM on the testing set to 42.3%.

4.6 Improved Duration Modelling

The next improvement in FOM was obtained by using a better duration model in the keyword models. This approach was first suggested by Gish et al [24]. Every HMM state in each keyword model was duplicated to form a *twin* state and the recurrent HMM connection was eliminated in *twin* states. A twin-state word model formed using this approach is illustrated in Figure 4-7. Adding twin state models makes it difficult for a HMM model to stay in a particular state for most of the keyword's duration and skip quickly through frames that it does not match well. The improved duration modelling is important because it reduces the chance of the wordspotter generating putative hits at certain time location

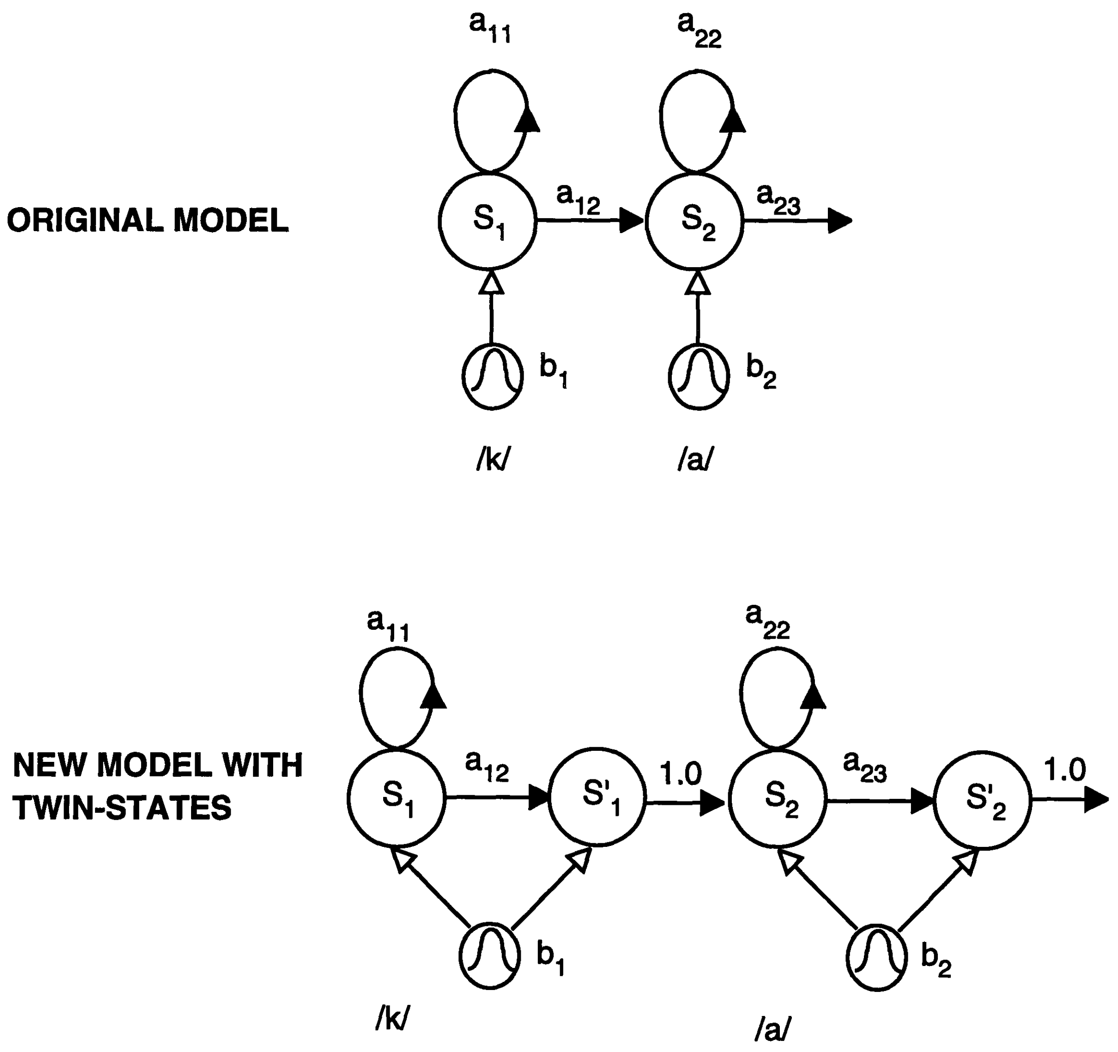


Figure 4-7: Enhancing the modelling of duration in the HMM keyword model.

solely due to a keyword state being matched very well. The better duration modelling increases the FOM from 42.3% to 47.4%.

4.7 Postprocessing

The telephone speech database used in this study contains many instances of crosstalk during which one side's recording contains the other side of the conversation as well, i.e., a single side conversation actually contains voices from both sides of the conversation. Since the crosstalk signals are strong and contain keywords, the wordspotting system picks out keywords in the crosstalk regions as well. According to NIST's FOM calculation method, if the wordspotting system detects a keyword from the opposite side of the conversation, then the putative hit should be ignored and counts neither as a hit nor as a false alarm. But if the keyword is not actually in the other side of the conversation, then the putative hit is counted as a false alarm.

To avoid the problem of spotting false alarms within crosstalk portions, a program called *rmxtalk* was implemented by the author. The program receives the labels generated by *xtalk* in the two-channel mode of the segment of speech in question. If fewer than 20% of the frames in the segment have energy greater than the corresponding frames from the other channel, then the putative hit is removed. This program has the ability to remove putative hits that are clearly from crosstalk because speech frames in crosstalk normally have lower energy than frames in the opposite channel. The threshold of 20% was set conservatively so that no true hit detections are inadvertently thrown out. After the putative hits with the number of frames below the threshold were left out of putative hits, the overall FOM improved from 47.4% to 50.5%.

4.8 Gender Detection

Male and female speech differ significantly in many aspects such as pitch, formant frequencies, and intonation. Many speech recognizers with separate male and female models have been successfully developed [37, 67]. There are three ways of using separate models for male and female speakers:

1. Determine the gender of the incoming speaker and use the appropriate model.

2. Combine the male and female models by running both in parallel .

3. Run both models separately and then combine the generated labels based on the likelihood of each label.

In this thesis, both the first and the second approach were explored. In a set of preliminary experiments on a different split of the database, the first approach resulted in combined FOM of 69.7% while the second approach resulted in a FOM of 69.3%. The FOM on the male set and the female set are listed in Table 4.3.

Although the FOMs from the two approaches are very close, the first approach was chosen because it has the advantage of lower computation requirement. Once the gender is determined, only one set of models is matched against the incoming speech. The system

Table 4.3: FOM using separate gender models vs. using parallel gender models.

Condition	Male Test	Female Test	Combined Testing
Separate Gender	64.3%	76.4%	69.7%
Parallel Gender	63.7%	75.4%	69.3%

described in this thesis consists of two sets of models, one trained on male data and another trained on female data. During actual spotting, the gender of the speaker is first identified, and then the appropriate set of models is used. A gender identification system that can reliably make the hard decision before actual wordspotting begins needs to be designed.

Gaussian mixture classifiers and Radial Basis Function classifiers can be used to classify a sequence of patterns. Gender identification experiments were carried out using Gaussian mixture classifiers and Radial Basis Function classifiers. Different algorithm parameters such as the number of inputs and number of clusters were explored in optimizing classifier performance. The best RBF classifier achieved an average of 1-2 errors out of 68 conversations and is sufficient for performing gender identification before wordspotting.

The input features are the mel-scaled cepstra values starting from $mcc(1)$. The conversations are first processed with the program *xtalk* to determine whether the speaker is speaking at any given frame. As described previously, *xtalk* compares the energy level in the two channels of the recording of the telephone conversation and assigns the active flag to the channel having the higher average energy. Only the active frames of a given speaker are used in the training data of that speaker. This procedure reduces the danger of having crosstalk sound patterns mixed with the data from the training conversation.

Three programs that have been recently developed under the *LNKseq* framework were used to conduct the study [44]. All frames in the designated sequences are treated as independent from each other and stored as examples of individual classes. The program *kmeans_seq* clusters the input sequence of frames by class using the k-means algorithm [50]. The program *gmix_seq* and *rbf_seq* then use the cluster parameter file as the starting point to train sequence classifiers. The RBF classifier is trained using matrix inversion [73] to minimize the mean squared error while the Gaussian mixture classifier is trained with maximum likelihood estimation [50]. The trained RBF and Gaussian mixture sequence classifiers are then used to test the ability of each classifier to correctly identify the class of an input sequence. During classification, the output of the classifier at each frame is treated as independent from other frames and multiplied with the network output of the previous frames. The final network output of the classifier is calculated using the equations below:

$$score\_male = \prod_{i=1}^{nframe} output_{i,male}, \tag{4.7}$$

and

$$score\_female = \prod_{i=1}^{nframe} output_{i,female}, \tag{4.8}$$

where i is the frame index, $nframe$ is the total number of frames in the input speech, $output_{i,male}$ is the output of the male classifier on frame i, and $output_{i,female}$ is the output of the female classifier on frame i.

4.8.1 Experimental Results

A series of experiments was conducted with both the Gaussian mixture classifier and the RBF classifier. When this set of experiments was conducted, two conversations were not used due to incorrect labelling supplied on the NIST CD-ROM. To obtain general experimental results with the limited data, the remaining 68 conversations were divided into four different splits, with each split containing 51 conversations for training the classifier and 17 conversations for evaluation. The total number of errors after all four splits of experiments were performed was summed and the results are shown in Figure 4-8. The chart shows that the best results were obtained with RBF classifiers with 128 centers. The number of errors was the same whether the first twelve feature values or the first eighteen feature values are used. For this set of data, the RBF sequence classifiers generally perform better than the Gaussian mixture classifiers.

4.9 Complexity of the Baseline Wordspotter

The number of states and the number of mixtures in the baseline wordspotter was set to reflect the amount of data available. The complexity of each keyword model is dependent on the amount of data available to train that model. Thus commonly occurring words such as *card* have 4 mixtures per state while less frequently occurring words such as *discover* have 1 mixture per state. Since many examples of filler speech and silence are available, 32

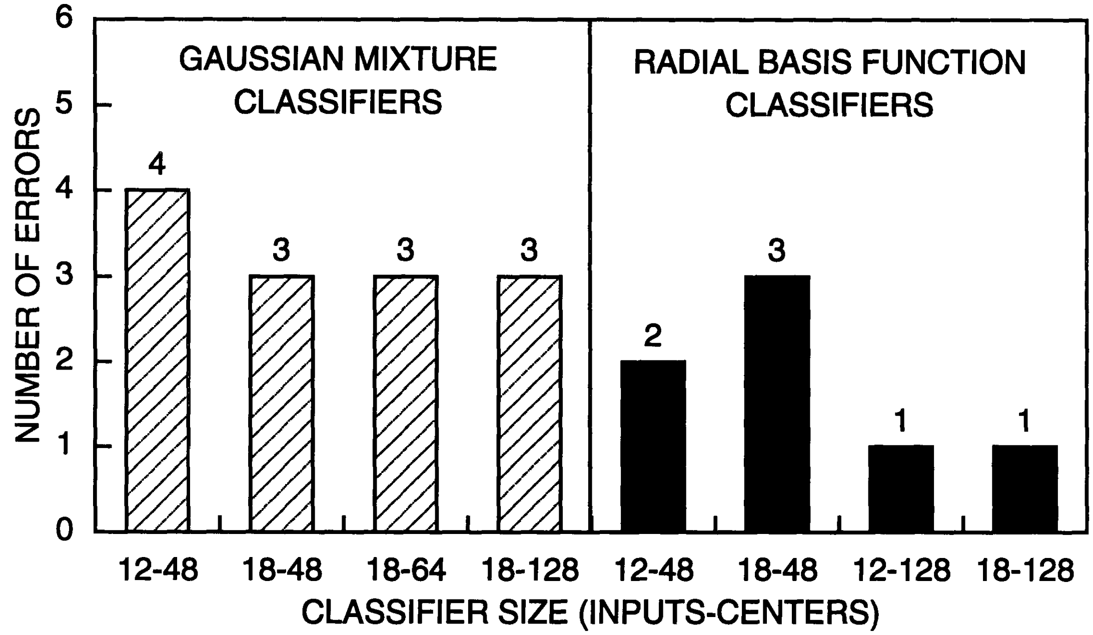


Figure 4-8: Comparison of the number of errors committed by Gaussian mixture based gender classifier vs. radial basis function based gender classifier.

mixtures are used in the filler and silence model. The filler model and the silence model each has one state due to the fact that, unlike the keyword speech patterns, filler and silence speech patterns are not expected to have a particular temporal sequence of sounds. The number of states and mixtures for all models are listed in Table 4.4. All subsequent experiments used the HMM topologies specified in this table. Covariance matrices were diagonal and variances were estimated separately for all states. An initial set of models was trained during 16 passes through the training data using word level estimation on only the excised words from the training conversations.

4.10 Performance of the Baseline Wordspotter

To ensure that the new algorithms developed in this thesis have real benefits, the high performance baseline wordspotter described above was developed and used as a basis for comparison. The sequential increase in the performance of the baseline wordspotter is shown in Table 4.5. The FOM is also plotted in Figure 4-9. The baseline wordspotter was trained using the 48 conversations in the training set and tested on the remaining 22 conversations in the testing set. The initial wordspotter relied on the likelihood generated from the Viterbi path to score the keywords. The overall testing set FOM was 16.9%

Table 4.4: The number of states and the number of RBF's per state used to model each keyword, filler speech, and silence. More RBF's are used per state when the number of training tokens is large.

Keyword	Number of States	Number of RBF's/State	Twin State?	Number of Training Tokens
account	10	1	Yes	25
american_express	30	1	Yes	37
balance	12	1	Yes	27
bank	8	1	Yes	31
card	8	4	Yes	435
cash	8	2	Yes	72
charge	8	2	Yes	98
check	8	2	Yes	79
credit	12	2	Yes	316
credit_card	20	4	Yes	250
discover	14	1	Yes	19
dollar	8	2	Yes	74
hundred	14	1	Yes	24
interest	12	2	Yes	72
limit	10	1	Yes	22
money	8	2	Yes	86
month	8	2	Yes	83
percent	12	1	Yes	33
twenty	14	1	Yes	15
visa	8	2	Yes	50
filler	1	32	No	Not Applicable
sil	1	32	No	Not Applicable

Table 4.5: The testing set FOM of the baseline wordspotter after stages of improvement.

Condition	Test FOM		
	Male	Female	Overall
Viterbi Scoring	24.0%	16.2%	16.9%
Peak-Picking Scoring	47.9%	39.8%	42.3%
Twin State	51.1%	45.0%	47.4%
Crosstalk Removal	54.5%	47.7%	50.5%
Separate Gender	56.0%	52.7%	52.9%
Embedded Reestimation	64.8%	61.5%	62.5%

(as described in 4.5.1). By using the peak-picking scoring method described in 4.5.2 to generate putative hits, the overall FOM was raised to 42.3%. The next improvement was using twin states to force the duration of all putative hits to be more than one frame on all states, which increased the overall FOM to 47.4%. After realizing that much crosstalk occurs in the database and such crosstalk can generate false alarms, the crosstalk removal algorithm described in Section 4.7 was implemented. This addition improved the overall FOM to 50.5%. The separate FOM for male and female testing speakers clearly shows that the female speakers are more difficult to spot. By training separate gender models and using the correct gender model to spot each speaker, the FOM was improved to 52.9%. Notice that the gap between FOM on the female test speakers and the male test speakers narrowed after separate models were trained for each gender. Lastly, by training the HMM models with embedded reestimation, the overall FOM increased to 62.5%. As a point of reference, the whole-word based wordspotter developed by Rohlicek et. al. at BBN has a FOM of 56.6% on this same database [67]. There are two major differences between the BBN system and the baseline wordspotter. First, the baseline wordspotter has a filler model that is trained separately from the keyword models while the BBN system uses the combination of all keyword models to model filler speech. Second, the keyword and filler model in the baseline system was trained with sentence level estimation while the BBN system was not trained with sentence level estimation. The baseline wordspotter is among the best whole-word based wordspotters tested on this database. With the baseline firmly established, novel techniques in training can be explored.

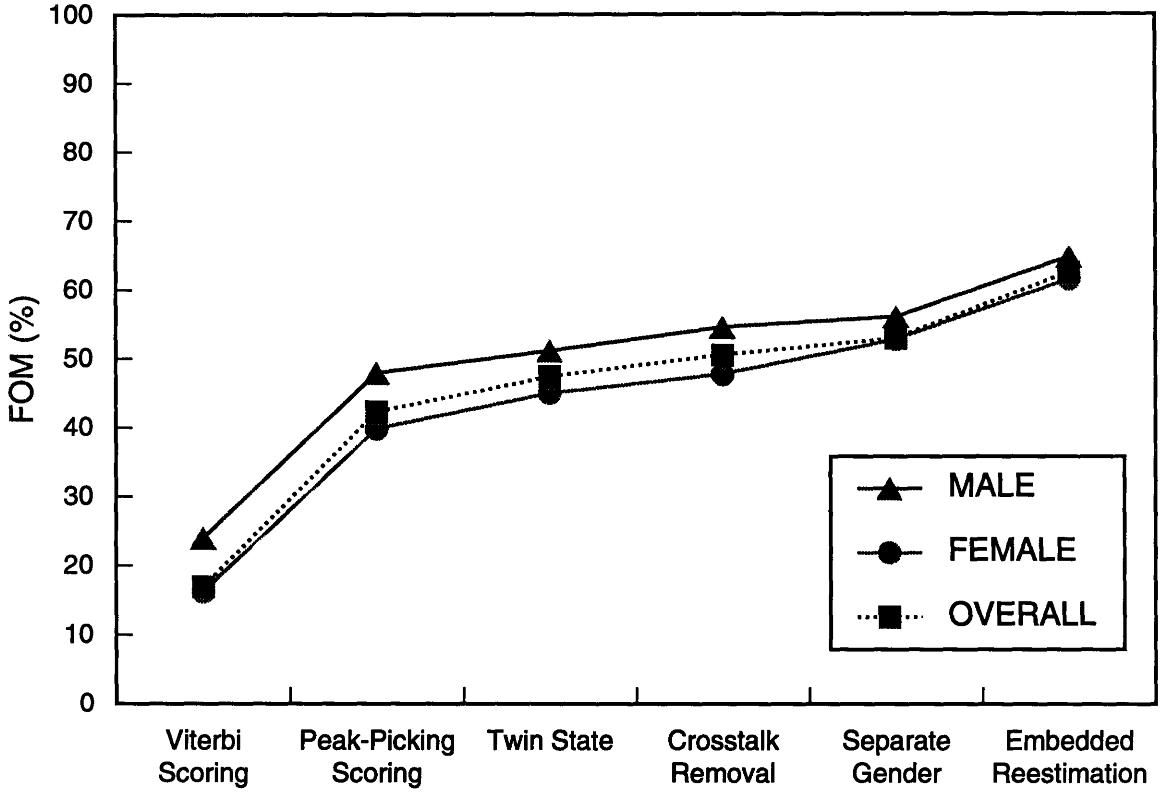


Figure 4-9: The change in overall FOM on the testing set after improvements are added to the baseline wordspotter.

4.11 Chapter Summary

This chapter described a whole-word based wordspotter that has been used as the baseline in this thesis. Section 4.2 described the steps that were used to convert a digitized speech signal of 8,000 samples per second to 2 streams of mel-scaled cepstra feature vectors with 25 values per frame and 100 frames per second. The reduction in data rate from 8,000 values per second to 2,500 values per second is an example of extracting useful information from the raw speech signal. Experiments were performed to determine a set of pre-processing steps that are suitable for removing variabilities inherent in the database. The preprocessing steps explored are crosstalk detection, RASTA filtering, and spectral normalization.

The baseline HMM based wordspotter was described in Section 4.3. A whole-word based HMM discrete word recognizer was used as the basis of the wordspotter. The training procedure of the baseline wordspotter was presented in Section 4.4. Two different methods of generating putative hits from the output of the wordspotter were described in Section 4.5. Section 4.6 presented a change to the wordspotter structure to ensure that all states contribute to the spotting task. Section 4.7 illustrated the steps that are performed to the output of the wordspotter to remove unwanted artifacts. Section 4.8 described the design and implementation of a gender detection system.

To improve the performance of the wordspotter, separate improvements such as gender separation and embedded reestimation were tried. The sequence of modifications increased the overall FOM on the testing set from 16.9% to 62.5%. The resulting whole-word based wordspotting system is one of the best wordspotters of similar complexity. The high performance provides a reasonable baseline for measuring improvements that result from applying two new approaches which will be introduced in the next two chapters.

Chapter 5

Figure of Merit Training

Spotting tasks require detection of target patterns from a background of richly varied non-target inputs. The performance measure of interest for these tasks, called the Figure of Merit (FOM), is the detection rate for target patterns when the false alarm rate is in an acceptable range. A new approach to training spotters which computes the FOM gradient for each input pattern and then directly maximizes the FOM using backpropagation is presented in this chapter. This approach eliminates the need for thresholds during training. It also uses network resources to model Bayesian *a posteriori* probability functions accurately only for patterns which have a significant effect on the detection accuracy over the false alarm rate of interest. FOM training increased detection accuracy by up to four percentage points for a hybrid radial basis function (RBF) hidden Markov model (HMM) wordspotter on the credit-card speech corpus.

5.1 Introduction

Spotting tasks require accurate detection of target patterns from a background of varied non-target inputs. Such problems share three common characteristics. First, the number of instances of target patterns is unknown. Second, patterns from background, non-target, classes are varied and often difficult to model accurately. Third, the performance measurement is the detection rate for target patterns when the false alarm rate is over a specified range. Neural network classifiers are often used for detection problems by training on target and background classes, optionally normalizing target outputs using the background output, and thresholding the resulting score to generate putative hits.

5.2 Prior Approaches of Discriminative Training

Discriminative training has been applied to the wordspotting task by other researchers. Rose has previously presented an approach to training a HMM based wordspotter using a discriminative training criterion [68]. In his approach, both the filler and the keyword models share a mixture of 128 Gaussians to represent the output probability of the HMM states, and the HMM models differ in the mixture weights of each state. During discriminative training, only the mixture weights of the keyword models are changed to maximize the difference between the log keyword probability and the log filler probability. One problem with this training approach is that false alarms are not used in modifying the model parameters. Also, the gradient that is used depends on the difference between the log keyword likelihood and the log filler likelihood. Section 5.4 will present Figure of Merit gradient charts which demonstrate why different keywords have different gradients with respect to the overall Figure of Merit. Rose reported an average increase of 30 percentage points in detection rate at low false alarm rates (from 16% at 0 false alarm per keyword per hour to 46% at 0 false alarm per keyword per hour). However, Rose used the Viterbi scoring method described in 4.5.1, which accounts for the very low detection rate before discriminative training. If the peak-picking algorithm described in 4.5.2 had been used, the amount of improvement might have been smaller.

Another approach of training a Multi-State Time-Delayed Neural Network (MS-TDNN) to perform wordspotting has been proposed by Zeppenfeld et al. [81, 83]. The Time-Delayed Neural Network (TDNN) classifier is a neural network classifier which receives an input composed of a matrix of spectral energy values that extend both in time and frequency. The TDNN estimates the likelihood of the input being a certain phoneme, and the likelihoods are integrated on a top layer to model the occurrence of a sequence of phonemes and passed through a sigmoid output unit to obtain an output between 0.0 and 1.0. During training, the TDNN has a target value of 1.0 on true keyword occurrences and 0.0 on false alarms. The training algorithm seeks to train the output of the wordspotter to be 0.0 on false alarms and 1.0 on true occurrences of the keywords. The system achieved a FOM of 51.04% on the official September 1992 testing set of the Switchboard credit card database [82]. Some problems with using the sigmoid output are discussed in Section 5.3.

Recently de la Torre and Acero introduced a discriminative training approach designed for wordspotters [13]. The error function is of the following form:

$$E = \frac{L_r}{N_k} \sum_{N_k} f(P_g - P_k) + \frac{L_f}{N_g} \sum_{N_g} f(P_k - P_g), \tag{5.1}$$

where L_r is the constant used to weight the importance of wrongly rejecting keywords, N_k is the number of keyword utterances, P_g is the log-probability of the garbage model, P_k is the log-probability of the keyword model, L_f is the constant used to weigh the importance of false alarm, and N_g is the number of non-keyword utterances. The function f is a sigmoid function of the following form:

$$f(x) = \frac{1}{(1 + exp(-Tx))}. \tag{5.2}$$

The constant T is used to determine the threshold over which the model is no longer trained. In [13] the method of choosing T was not specified. Using this methodology to train the parameters of the garbage model, they successfully reduced the predefined error function of a wordspotting system from 5.50 to 5.07. The word error rate was reduced from 9.8% to 6.9%, but the rejection rate on true keywords rose from 0.8% to 12.0%. The approach described in this thesis has the following differences:

- Different keywords have different gradients.
- Keyword models are trained instead of the garbage model.
- Setting of threshold T is not necessary.

5.3 Problems with Simple Backpropagation Training

Neural network classifiers used for spotting tasks can be trained using conventional backpropagation procedures with 1 of N desired outputs and a squared error cost function. This approach to training does not maximize the FOM because it attempts to estimate Bayesian *a posteriori* probability functions accurately for all inputs even if a particular input has little effect on detection accuracy at false alarm rates of interest. Excessive network resources may be allocated to modelling the distribution of common background inputs dissimilar from targets and of high-scoring target inputs which are easily detected. This problem can be addressed by training only when network outputs are above thresholds. Problems with this approach are threefold:

1. It is difficult to set the threshold for different keywords. For example, for certain long words such as *discover*, there may be many more true hits than false alarms, and the network will be trained with many more positive examples than negative examples, even though the negative examples change the final Figure of Merit (FOM) much more than positive examples.
2. Target values must be set for all putative hits to calculate the gradient that is used to modify model parameters. Using fixed target values of 1.0 and 0.0 requires careful normalization of network output scores to prevent saturation and to maintain back-propagation effectiveness.
3. The gradient calculated from a fixed target value does not reflect the actual impact on the FOM.

Figure 5-1 shows the gradient of true hits and false alarms when target values are set to be 1.0 for true hits and 0.0 for false alarms, the output unit is sigmoidal, and the threshold for a putative hit is set to roughly 0.6. The gradient is the derivative of the squared error cost with respect to the input of the sigmoidal output unit. As can be seen, low-scoring hits or false alarms that may affect the FOM are ignored, the gradient is discontinuous at the threshold, the gradient does not fall to zero fast enough at high values, and the relative sizes of the hit and false alarm gradients are fixed for all words and do not reflect the true effect of a hit or a false alarm on the FOM.

5.4 Figure of Merit Training

A new approach to training a spotter system called *Figure of Merit Training* is to directly compute the FOM and its derivative. This derivative is the change in FOM over the change in the output score of a putative hit and can be used instead of the derivative of a squared-error or other cost functions during training.

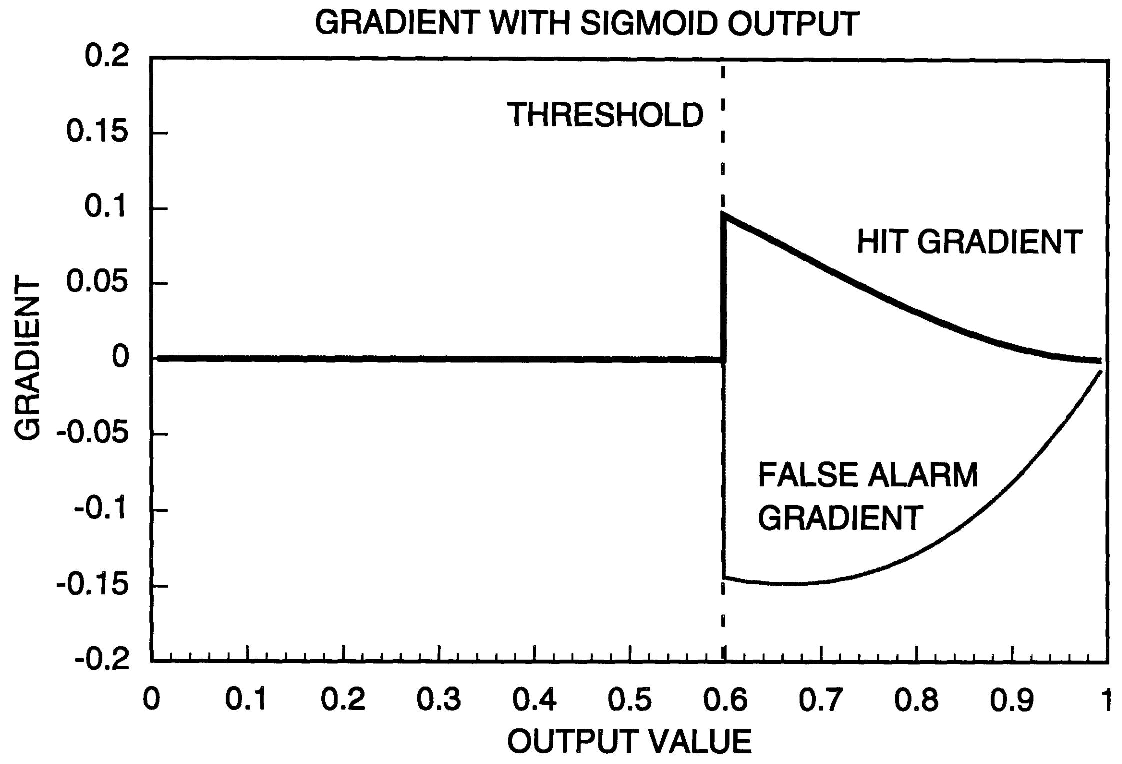


Figure 5-1: The gradient for a sigmoid output unit when the target value for true hits is set to 1.0 and the target value for false alarms is set to 0.0.

5.4.1 Calculation of the Figure of Merit Gradient

Since the FOM is calculated by sorting true hits and false alarms separately for each target class and forming detection versus false alarm curves, these measures and their derivatives can not be computed analytically. Instead, the FOM and its derivative are computed using fast sort routines. These routines insert a new putative hit into an already sorted list and calculate the change in the FOM caused by that insertion. The running putative hit list used to compute the FOM is updated after every new putative hit is observed and it must contain all putative hits observed during the most recent past training cycle through all training patterns. The gradient estimate is smoothed over nearby putative hit scores to account for the quantized nature of the change in FOM. The derivation of the Figure of Merit gradient for the keyword *account* on the training set is illustrated in Figure 5-2. The top plot in the figure shows the impact of a true hit for the word *account* as its score is changed from -100 to 300. When the score is very low (below 0), the putative hit has no impact because it is ranked below many false alarms. When the score is very high, there are no longer any false alarms with scores above the putative hit's score and thus there is no change in FOM. If the score falls in the middle region, when the putative hit's score surpasses that of a false alarm, the overall FOM is increased. The middle plot in Figure 5-2 illustrates the change in FOM (delta FOM) when the score of the putative hit is varied. The delta FOM is discontinuous because the distribution of false alarms is random. Smoothing is performed by using a least squared slope approximation on 19 samples from a range around the score of the input putative hit. The range of each keyword is calculated by taking the difference between the scores of the 20th percentile putative hit and the 80th percentile putative hit for the keyword. For example, suppose that the word *card* has 100 putative hits that are sorted in descending order. Let $score[n]$ represent the score of the nth putative hit. The score range is calculated with the formula:

$$score\ range = score[20] - score[80] \tag{5.3}$$

Let $FOM(score)$ be the overall FOM when a putative hit with the score $score$ is inserted in the putative hit list. The FOM gradient for a putative hit with the score hit_score is estimated by performing a least square slope estimation on the following 19 pairs of data points $(score, FOM)$:

$$FOM(score), \quad score = \begin{array}{l} -\frac{9}{8} \cdot score\ range + hit\_score, \\ -\frac{8}{8} \cdot score\ range + hit\_score, \\ \qquad\vdots \\ \frac{9}{8} \cdot score\ range + hit\_score \end{array}$$

Figure 5-3 shows plots of linearly scaled gradients for the 20-word hybrid wordspotter. Each value on the curve represents the smoothed change in the FOM that occurs when a single hit or false alarm with the specified normalized log output score is inserted into the current putative hit list. Gradients are positive for putative hits corresponding to true hits and negative for false alarms. They also fall off to zero for putative hits with extremely high or low scores. Shapes of these curves vary across words: the relative importance of a hit or a false alarm, the normalized output score which results in high gradient values, and the shape of the gradient curve varies. Use of a squared error or other cost function with sigmoid output nodes would not generate this variety of gradients or automatically

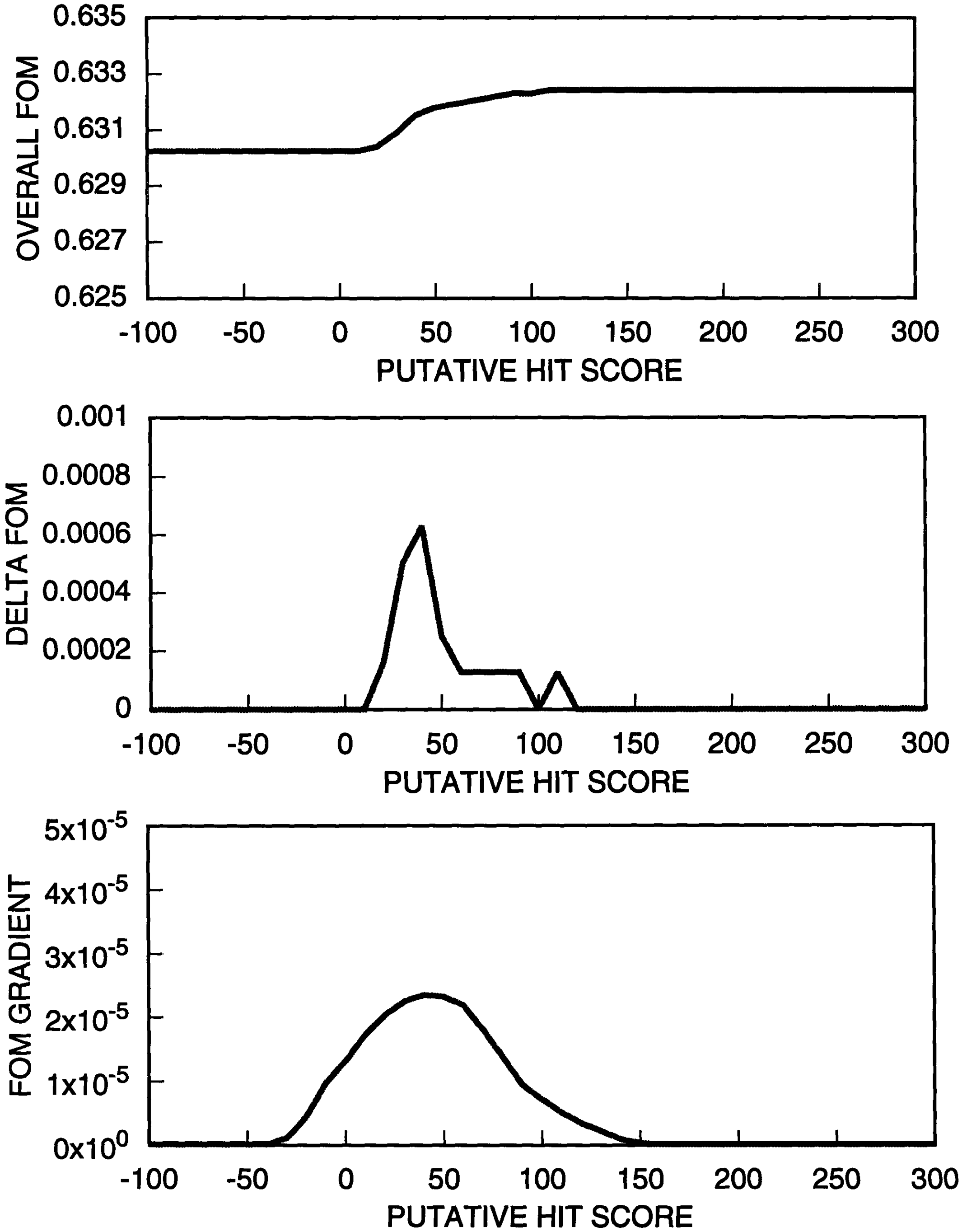


Figure 5-2: The generation of the Figure of Merit gradient. The score for a true hit of the word *account* is varied from -100 to 300. The resulting FOM is shown in the top chart. The change in FOM (delta FOM) across the same score range is shown in the middle chart. Smoothing is applied to the values in the middle chart to derive the FOM gradient used for training, shown in the bottom chart.

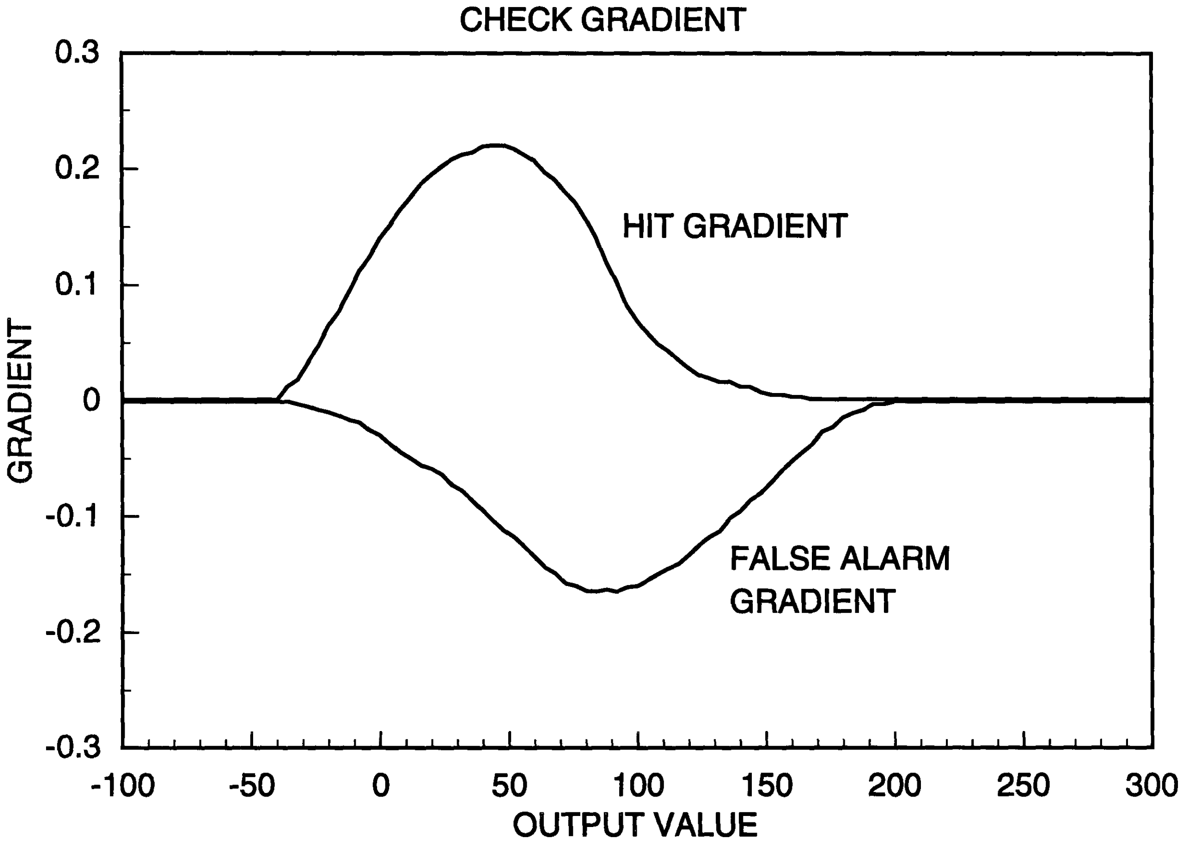


Figure 5-3: Figure of Merit gradients computed for true hits (HIT) and false alarms (FA) with scores ranging from -100 to 300 for the keyword *check*.

identify the range of putative hit scores where gradients should be high. Application of FOM training requires only the gradients shown in these curves with no supplementary thresholds. Patterns with low and high inputs will have a minimal effect during training without using thresholds because they produce gradients near zero.

5.4.2 Different Figure of Merit Gradients for Words of Different Difficulty

Different keywords have dramatically different gradients. For example, the keyphrase *credit card* is long and the detection rate is high. The overall FOM thus does not change much if more true hits are found. A high scoring false alarm, however, decreases the FOM drastically. There is thus a large negative gradient for false alarms for *credit card.* The keywords *account* and *check* are usually short in duration and thus more difficult to detect, and thus any increase in number of true hits strongly increases the overall FOM. On the other hand, since in this database, the words *account* and *check* occur much less frequently than *credit card,* a high scoring false alarm for the words *account* and *check* has less impact on the overall FOM. The gradient for false alarms for these words is thus correspondingly smaller. Comparing the curves in Figure 5-4 with the fixed prototypical curve in Figure 5-1 demonstrates the dramatic differences in gradients that occur when the gradient is calculated to maximize the FOM directly instead of using a threshold with sigmoid output nodes.

5.5 Implementation

The FOM training algorithm was applied to train the high performance baseline wordspotter described in Chapter 4. A new addition to the hybrid wordspotter, called *State Weights*, was added to allow for weighting the contribution of different states in a keyword model. The training methodology uses the high performance wordspotter developed in Chapter 4 to spot training conversations and improve the performance of the wordspotter on the putative hits generated by the baseline wordspotter.

5.5.1 State Weights

The state weight is a penalty added for each frame assigned to a state. The weight for each individual state is adjusted according to how important each state is to the detection of the keyword. For example, many false alarms for the word *card* are words that sound like

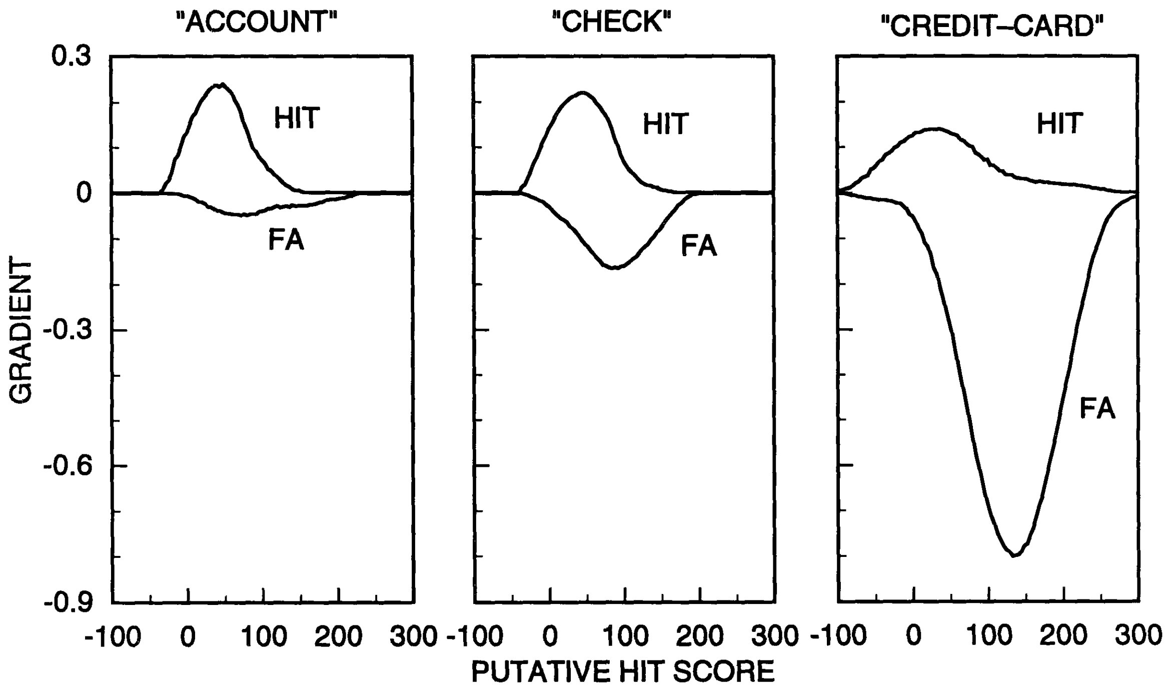


Figure 5-4: Figure of Merit gradients computed for true hits (HIT) and false alarms (FA) with scores ranging from -100 to 300 for the keywords *account*, *check*, and *credit-card*.

part of the keyword such as *hard* or *far*. The first few states of the *card* model represent the phoneme /k/ and false alarms stay in these front states only a short time. If the state weight of the first few states of the card model is large, then a true hit has a larger score than false alarms.

5.5.2 Training Methodology

Figure 5-5 describes the methodology used for applying FOM training. FOM training is applied to the high-performance HMM wordspotter after sentence level estimation is completed. Word models in the HMM wordspotter are first used to spot keywords in training conversations. The true hits, the false alarms, and the misses are then used as training tokens for Figure of Merit training. The FOM gradient of each putative hit is calculated when the hit is inserted into the putative hit list. In the case of misses, a fixed score of 100.0 was arbitrarily chosen to be used as the score for generating the FOM gradient. The speech segment corresponding to a putative hit is excised from the conversation speech file and the corresponding keyword model is used to match each frame with a particular state in the model using a Viterbi backtrace (shown in Figure 5-6.) The gradient is then

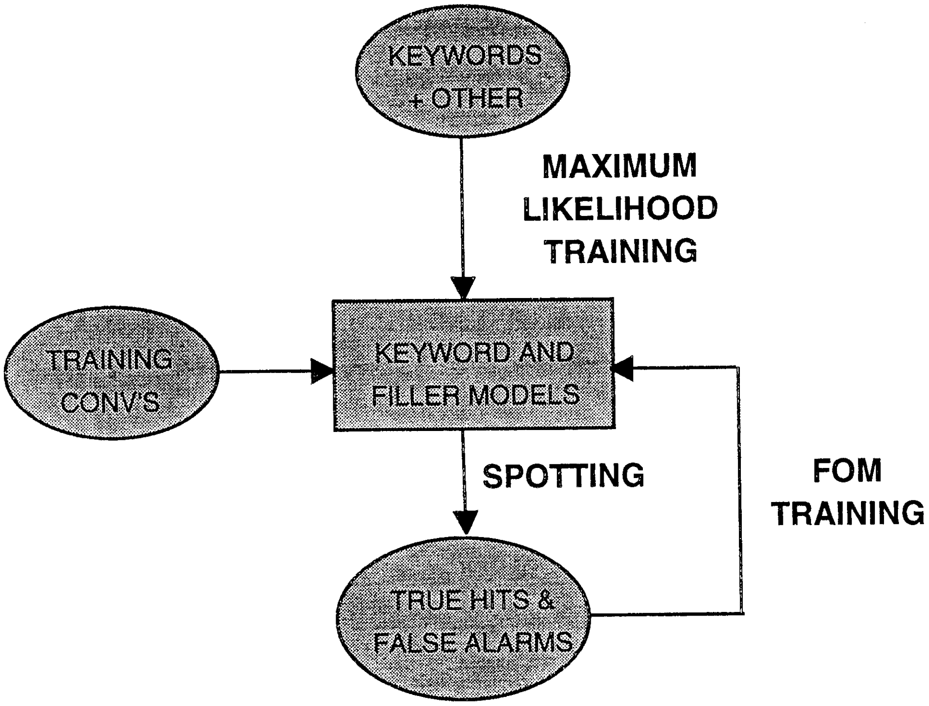


Figure 5-5: The training methodology for Figure of Merit training. The models are first trained using Baum-Welch training. Then the models are used to spot conversations and the resulting true hits and false alarms are used to train the models discriminatively.

used to adjust the location of each Gaussian component in a node as in RBF classifiers [40] and also the state weight of each state.

The putative hit score which is used to detect peaks representing putative hits is generated according to

$$S_{total} = S_{keyword} - S_{filler}. \tag{5.4}$$

In this equation, S_{total} is the putative hit score, $S_{keyword}$ is the log Viterbi score in the last state of a specific keyword model computed using the Viterbi algorithm from the beginning of the conversation to the frame where the putative hit ended, and S_{filler} is the log Viterbi score in the last state of the filler model. The filler score is used to normalize the keyword score and approximate the *a posterior* probability. The keyword score is calculated using a modified form of the Viterbi algorithm

$$\alpha_i(t+1) = max[\alpha_i(t) + a_{i,j}, \alpha_{i-1}(t) + a_{i-1,j}] + b_i(t,x) + w_i. \tag{5.5}$$

This equation is identical to the normal Viterbi recursion for left-to-right linear word models after initialization, except that the extra state weight w_i is added. In this equation, $\alpha_i(t)$ is the log Viterbi score in state i at time t, $a_{i,j}$ is the log of the transition probability from state i to state j, and $b_i(t, x)$ is the log likelihood for state i for the input feature vector x at time t.

Word scores are computed and a peak-picking algorithm looks for maxima above a low threshold. After a peak representing a putative hit is detected, frames of a putative hit are aligned with the states in the keyword model using the Viterbi backtrace and both the means of Gaussians in each state and state weights of the keyword model are modified. State weights are modified according to the following equation:

$$w_i(t+1) = w_i(t) + gradient \times \eta_{state} \times duration. \tag{5.6}$$

In this equation, $w_i(t)$ is the state weight in state i at time t, *gradient* is the FOM gradient for the putative hit, η_{state} is the step size for state weight adaptation, and *duration* is the number of frames aligned to state i. If a true hit occurs and the gradient is positive, the state weight is increased in proportion to the number of frames assigned to a state. If a false alarm occurs, the state weight is reduced in proportion to the number of frames assigned to a state. The state weight will thus be strongly positive if there are many more frames from true hits than from false alarms. It will be strongly negative if there are more frames from false alarms than from true hits. Using state weight values should thus improve discrimination between true hits and false alarms.

The center of the Gaussian components within each state, which are similar to Gaussians in radial basis function networks, are modified according to the following equation:

$$m_{ijk}(t+1) = m_{ijk}(t) + gradient \times \eta_{center} \times \frac{w_{ij} N(x_k; m_{ijk}, \sigma_{ijk})}{\sum_{j=1}^{ncenter} w_{ij} N(x_k; m_{ijk}, \sigma_{ijk})} \times \frac{x_k(t) - m_{ijk}(t)}{\sigma_{ijk}}. \tag{5.7}$$

In this equation, $m_{ijk}(t)$ is the kth component of the mean vector for the jth Gaussian mixture in HMM state i at time t, *gradient* is the FOM gradient, η_{center} is the step size for moving Gaussian centers, $x_k(t)$ is the value of the kth component of the input feature vector at time t, σ_{ijk} is the standard deviation of the kth component of the jth Gaussian

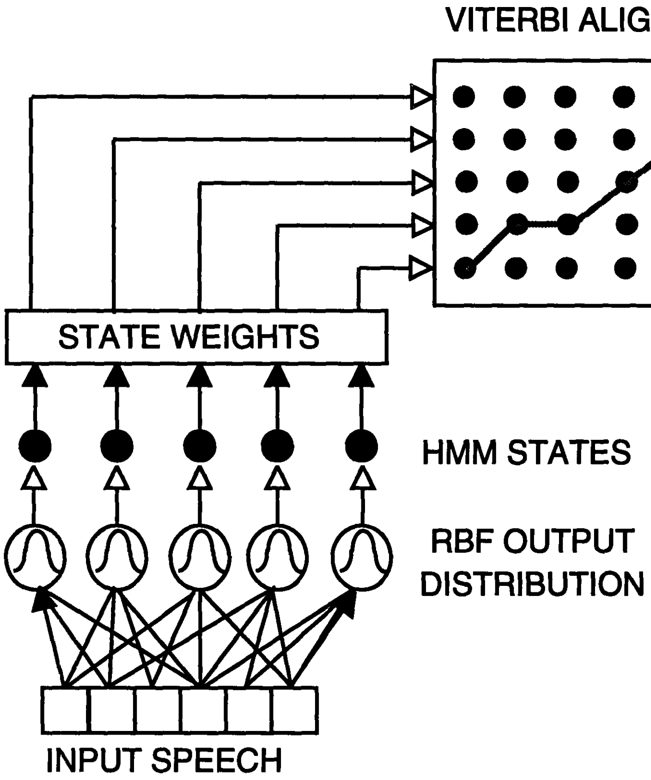


Figure 5-6: During training, frames of input speech are matched to states of the prospective keyword model through Viterbi alignment.

mixture in HMM state i, *ncenter* is the number of Gaussian mixtures, w_{ij} is the mixture weight of each Gaussian mixture, and $N(x_k; m_{ijk}, \sigma_{ijk})$ is the Gaussian function. For each true hit, the centers of Gaussian mixtures in a state move toward the observation vectors of frames assigned to a particular state. For a false alarm, the centers move away from the observation vectors that are assigned to a particular state. Over time, the centers move closer to the true hit observation vectors and further away from false alarm observation vectors. Note that when a Gaussian mixture is more likely compared to other Gaussian mixtures, then the following terms in Equation 5.7 becomes large and larger changes are made to the Gaussian mixture:

$$\frac{w_{ij} N(x_k; m_{ijk}, \sigma_{ijk})}{\sum_{j=1}^{ncenter} w_{ij} N(x_k; m_{ijk}, \sigma_{ijk})}. \tag{5.8}$$

Thus, Gaussian centers that are closer to the input pattern $x(t)$ get changed more heavily. Equation 5.7 is different from Equation 3.10 in that the term $\sum_{j=1}^{ncenter} w_{ij} N(x_k; m_{ijk}, \sigma_{ijk})$ is added in the denominator. This term is the result of using the log of the output of the radial basis functions in the wordspotting system. When the derivative of a log function $ln(f(x))$ is taken, the derivative takes the form of:

$$\frac{\partial ln(f(x))}{\partial x} = \frac{1}{f(x)} \cdot \frac{\partial f(x)}{\partial x}. \tag{5.9}$$

5.6 Experimental Results

Experiments were performed using the baseline HMM wordspotters that were trained using the maximum likelihood algorithm as described in Chapter 4. FOM training was then performed for five iterations through the training data. On each pass, conversations were presented in a new random order. The changes in FOM for the training set and the testing set are shown in Figure 5-7 and Figure 5-8. The FOM on the training data for both male and female speakers increased by more than six percentage points after five epochs of training. The FOM on the male testing set increased by 2.6 percentage points (64.8% to 67.4%) after five passes through the training data, but then decreased with further training (for example, the FOM is 66.5% after 10 epochs of training). The male training set FOM increased from 83.4% to 95.5% after 10 epochs of FOM training. The FOM on the female testing set increased by 4.1 percentage points (61.5% to 65.6%) after five passes through

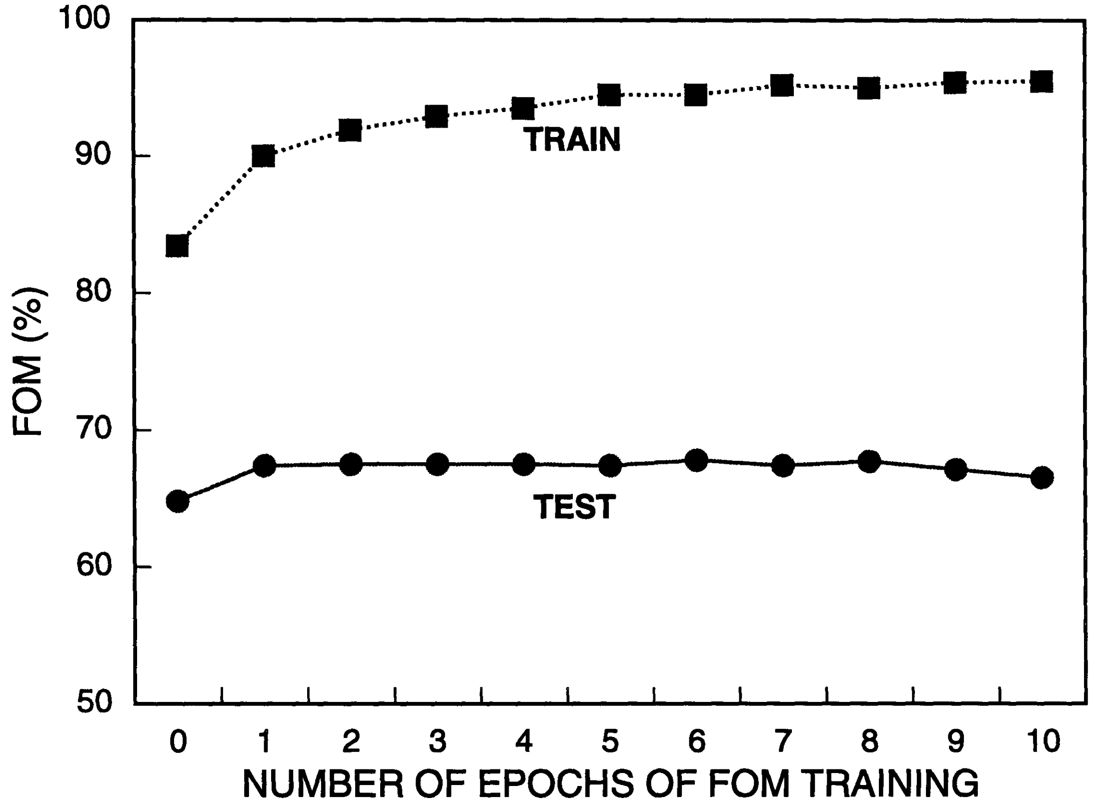


Figure 5-7: The FOM for the male training and testing set versus the number of epochs of FOM training.

the training data, but then decreased with further training (for example, the FOM is 65.2% after 10 epochs of training). The female training set FOM increased from 88.5% to 95.7% after 10 epochs of FOM training.

This result suggests that the structure learned during the final five training epochs is overfitting the training data so that the performance on the training set improves while the performance on the testing set deteriorates.

A graphical illustration of the effect of FOM back-propagation training can be seen in Figure 5-9. The upper plot shows a histogram of true hits and false alarms for all putative hits with positive normalized scores from an earlier experiment for the word *card*. Bars that count true hits are filled and bars that count false alarms are empty. The separation between true hits and false alarms for high-scoring putative hits is relatively poor. There are only nine high-scoring true hits before the highest scoring false alarm is encountered with a score of roughly 150 and the FOM is 54.3%. The lower histograms in Figure 5-9 shows the effect of FOM back-propagation training. The separation between true hits and false

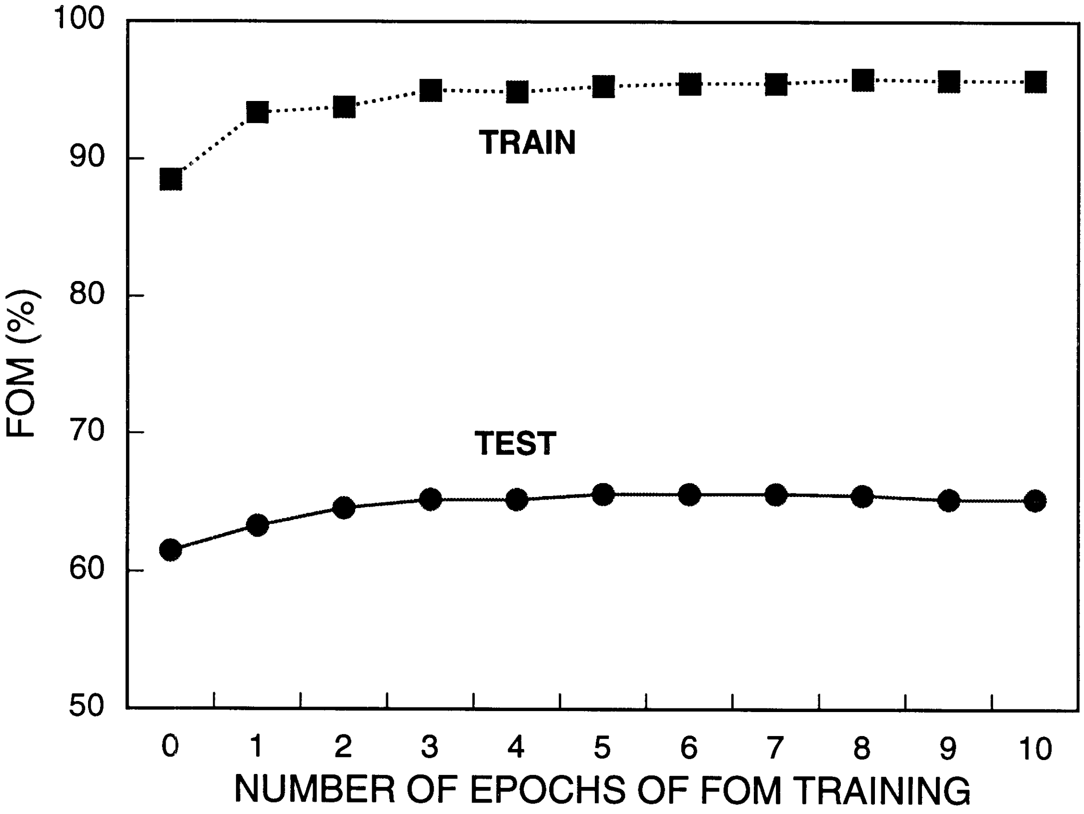


Figure 5-8: The FOM for the female training and testing set versus the number of epochs of FOM training.

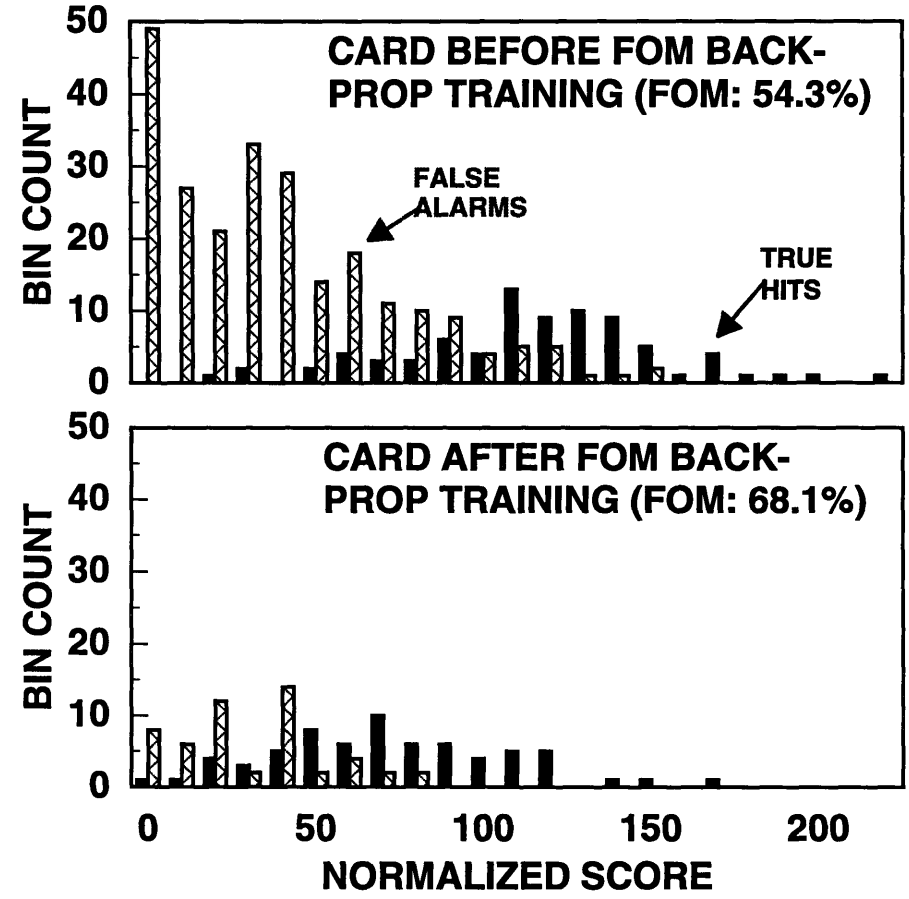


Figure 5-9: Histograms for true hits (solid bars) and false alarms (hollow bars) for the word *card* before and after FOM back-propagation training.

alarms is much better for the highest-scoring putative hits. There are now 23 high-scoring true hits before the highest scoring false alarm is encountered with a score of roughly 80 and FOM has increased to 68.1%. FOM back-propagation has thus reduced the score of the high-scoring false alarms while maintaining high scores for the true hits.

The per-word performances before and after FOM training for the male speakers are listed in Table 5.1 and Table 5.2. The overall FOM improved by 2.6 percentage points, from 64.8% to 67.4%. One strong effect of FOM training is the number of putative hits generated by the wordspotters. Before FOM training, the wordspotter generated a total of 2,978 putative hits, 326 of which are true hits. After five iterations of FOM training, the wordspotter generated only 1,190 putative hits, 319 of which are true hits. While the number of true hits was reduced by only 2.2%, the number of false alarms was reduced by 40%. Notice that performance on the keyword *discover* is poor. Both the number of training tokens (9) and the number of training speakers (4) for *discover* are small (see Tables 2.4 and 2.5), resulting in keyword models that do not generalize well. In Chapter 6 voice transformation techniques are introduced which improve performance on this keyword.

For the female speakers, the results before and after FOM training are listed in Table 5.4 and Table 5.5 respectively. Again, the number of false alarms has been drastically reduced from 2,598 to 1,207, while the number of hits detected has only been slightly reduced from 233 to 227. The overall FOM increased by 4.1 percentage points from 61.5% to 65.6%. Notice that performance on the keywords *hundred* and *twenty* are poor. Both the number of training tokens and the number of training speakers for these two words are small (see Tables 2.4 and 2.5), resulting in keyword models that do not generalize well. In Chapter 6 voice transformation techniques are introduced which improve performance on these keywords. Tables 5.3 and 5.6 present the change in FOM after 5 iterations of FOM training. The largest individual FOM increases were 17.4 percentage points on the word *cash* for the male set and 15.1 percentage points on the word *card* for the female set. Tables 5.7 and 5.8 lists the performance of the wordspotter on the combined male/female testing set.

Table 5.1: The per-keyword and overall FOM for the male testing set before FOM training. (The two numbers under True Hits represent the number of true hits detected over the number of true hits present in the conversations. The two numbers under False Alarms represent the number of false alarms over the number of putative hits generated.)

Keyword	FOM	True Hits	Det %	False Alarms	FA %
account	55.6%	7/8	87.5%	117/124	94.4%
american_express	87.5%	7/8	87.5%	0/7	0.0%
balance	73.3%	6/7	85.7%	18/24	75.0%
bank	58.4%	17/19	89.5%	212/229	92.6%
card	52.3%	49/49	100.0%	243/292	83.2%
cash	67.5%	15/15	100.0%	241/256	94.1%
charge	76.7%	20/20	100.0%	129/149	86.6%
check	70.5%	27/27	100.0%	227/254	89.4%
credit	62.1%	11/13	84.6%	67/78	85.9%
credit_card	95.0%	70/73	95.9%	12/82	14.6%
discover	0.0%	0/2	0.0%	0/0	0.0%
dollar	62.6%	14/14	100.0%	314/328	95.7%
hundred	43.2%	8/14	57.1%	38/46	82.6%
interest	64.5%	10/12	83.3%	209/219	95.4%
limit	80.2%	4/4	100.0%	12/16	75.0%
money	45.7%	11/11	100.0%	329/340	96.8%
month	52.4%	22/22	100.0%	306/328	93.3%
percent	43.6%	8/10	80.0%	51/59	86.4%
twenty	36.0%	5/10	50.0%	44/49	89.8%
visa	26.0%	15/17	88.2%	83/98	84.7%
Overall	64.8%	326/355	91.8%	2652/2978	89.1%

Table 5.2: The per-keyword and overall FOM for the male testing set after 5 iterations of FOM training. (The two numbers under True Hits represent the number of true hits detected over the number of true hits present in the conversations. The two numbers under False Alarms represent the number of false alarms over the number of putative hits generated.)

Keyword	FOM	True Hits	Det %	False Alarms	FA %
account	57.2%	7/8	87.5%	50/57	87.7%
american_express	87.5%	7/8	87.5%	0/7	0.0%
balance	68.8%	6/7	85.7%	21/27	77.8%
bank	56.4%	15/19	78.9%	83/98	84.7%
card	48.4%	47/49	95.9%	50/97	51.5%
cash	84.9%	14/15	93.3%	62/76	81.6%
charge	70.6%	20/20	100.0%	57/77	74.0%
check	73.4%	26/27	96.3%	78/104	75.0%
credit	67.0%	11/13	84.6%	33/44	75.0%
credit_card	95.8%	71/73	97.3%	34/105	32.4%
discover	0.0%	0/2	0.0%	0/0	0.0%
dollar	78.9%	14/14	100.0%	75/89	84.3%
hundred	44.9%	8/14	57.1%	30/38	78.9%
interest	71.7%	9/12	75.0%	23/32	71.9%
limit	74.3%	4/4	100.0%	12/16	75.0%
money	47.8%	10/11	90.9%	72/82	87.8%
month	67.1%	22/22	100.0%	61/83	73.5%
percent	42.6%	8/10	80.0%	57/65	87.7%
twenty	36.8%	6/10	60.0%	38/44	86.4%
visa	35.7%	14/17	82.4%	35/49	71.4%
Overall	67.4%	319/355	89.9%	871/1190	73.2%

Table 5.3: The per-keyword and overall FOM for the male testing set before and after 5 iterations of FOM training and the change in FOM.

Keyword	FOM Before	FOM After	Change in FOM
account	55.6%	57.2%	0.6%
american_express	87.5%	87.5%	0.0%
balance	73.3%	68.8%	-4.5%
bank	58.4%	56.4%	-2.0%
card	52.3%	48.4%	-3.9%
cash	67.5%	84.9%	17.4%
charge	76.7%	70.6%	-6.1%
check	70.5%	73.4%	2.9%
credit	62.1%	67.0%	4.9%
credit_card	95.0%	95.8%	0.8%
discover	0.0%	0.0%	0.0%
dollar	62.6%	78.9%	16.3%
hundred	43.2%	44.9%	1.7%
interest	64.5%	71.7%	7.2%
limit	80.2%	74.3%	-5.9%
money	45.7%	47.8%	2.1%
month	52.4%	67.1%	14.7%
percent	43.6%	42.6%	-1.0%
twenty	36.0%	36.8%	0.8%
visa	26.0%	35.7%	9.7%
Overall	64.8%	67.4%	2.6%

Table 5.4: The per-keyword and overall FOM for the female testing set before FOM training. (The two numbers under True Hits represent the number of true hits detected over the number of true hits present in the conversations. The two numbers under False Alarms represent the number of false alarms over the number of putative hits generated.)

Keyword	FOM	True Hits	Det %	False Alarms	FA %
account	35.1%	3/4	75.0%	6/9	66.7%
american_express	75.0%	3/4	75.0%	0/3	0.0%
balance	62.9%	6/7	85.7%	23/29	79.3%
bank	63.0%	6/6	100.0%	174/180	96.7%
card	42.7%	33/33	100.0%	208/241	86.3%
cash	58.7%	15/15	100.0%	274/289	94.8%
charge	76.6%	17/17	100.0%	136/153	88.9%
check	31.2%	7/8	87.5%	147/154	95.5%
credit	66.5%	20/21	95.2%	49/69	71.0%
credit_card	96.5%	38/39	97.4%	36/74	48.6%
discover	87.5%	7/8	87.5%	5/12	41.7%
dollar	49.4%	10/11	90.9%	334/344	97.1%
hundred	0.0%	0/3	0.0%	0/0	0.0%
interest	77.7%	19/20	95.0%	57/76	75.0%
limit	45.1%	4/7	57.1%	86/90	95.6%
money	27.9%	13/14	92.9%	458/471	97.2%
month	55.9%	15/17	88.2%	402/417	96.4%
percent	84.7%	8/9	88.9%	57/65	87.7%
twenty	0.0%	0/6	0.0%	18/18	100.0%
visa	44.8%	9/9	100.0%	128/137	93.4%
Overall	61.5%	233/258	90.3%	2598/2831	91.8%

Table 5.5: The per-keyword and overall FOM for the female testing set after 5 iterations of FOM training. (The two numbers under True Hits represent the number of true hits detected over the number of true hits present in the conversations. The two numbers under False Alarms represent the number of false alarms over the number of putative hits generated.)

Keyword	FOM	True Hits	Det %	False Alarms	FA %
account	39.8%	3/4	75.0%	5/8	62.5%
american_express	75.0%	3/4	75.0%	0/3	0.0%
balance	69.6%	6/7	85.7%	23/29	79.3%
bank	58.3%	6/6	100.0%	154/160	96.2%
card	57.8%	31/33	93.9%	34/65	52.3%
cash	60.8%	15/15	100.0%	50/65	76.9%
charge	77.4%	16/17	94.1%	43/59	72.9%
check	32.8%	6/8	75.0%	69/75	92.0%
credit	68.7%	21/21	100.0%	47/68	69.1%
credit_card	94.5%	39/39	100.0%	36/75	48.0%
discover	87.5%	7/8	87.5%	6/13	46.2%
dollar	63.1%	10/11	90.9%	58/68	85.3%
hundred	0.0%	0/3	0.0%	0/0	0.0%
interest	72.0%	18/20	90.0%	30/48	62.5%
limit	51.8%	4/7	57.1%	62/66	93.9%
money	39.3%	10/14	71.4%	122/132	92.4%
month	61.9%	15/17	88.2%	117/132	88.6%
percent	87.8%	8/9	88.9%	52/60	86.7%
twenty	0.0%	0/6	0.0%	18/18	100.0%
visa	55.5%	9/9	100.0%	54/63	85.7%
Overall	65.6%	227/258	88.0%	980/1207	81.2%

Table 5.6: The per-keyword and overall FOM for the female testing set before and after 5 iterations of FOM training and the change in FOM.

Keyword	FOM Before	FOM After	Change in FOM
account	35.1%	39.8%	4.7%
american_express	75.0%	75.0%	0.0%
balance	62.9%	69.6%	6.7%
bank	63.0%	58.3%	-4.7%
card	42.7%	57.8%	15.1%
cash	58.7%	60.8%	2.1%
charge	76.6%	77.4%	0.8%
check	31.2%	32.8%	1.6%
credit	66.5%	68.7%	2.2%
credit_card	96.5%	94.5%	-2.0%
discover	87.5%	87.5%	0.0%
dollar	49.4%	63.1%	13.7%
hundred	0.0%	0.0%	0.0%
interest	77.7%	72.0%	-5.7%
limit	45.1%	51.8%	6.7%
money	27.9%	39.3%	11.4%
month	55.9%	61.9%	6.0%
percent	84.7%	87.8%	3.1%
twenty	0.0%	0.0%	0.0%
visa	44.8%	55.5%	10.7%
Overall	61.5%	65.6%	4.1%

Table 5.7: The per-keyword and overall FOM for the combined testing set. (The two numbers under True Hits represent the number of true hits detected over the number of true hits present in the conversations. The two numbers under False Alarms represent the number of false alarms over the number of putative hits generated.)

Keyword	FOM	True Hits	Det %	False Alarms	FA %
account	42.1%	10/12	83.3%	123/133	92.5%
american_express	83.3%	10/12	83.3%	0/10	0.0%
balance	62.7%	12/14	85.7%	41/53	77.4%
bank	59.6%	23/25	92.0%	386/409	94.4%
card	50.6%	82/82	100.0%	451/533	84.6%
cash	62.7%	30/30	100.0%	515/545	94.5%
charge	73.7%	37/37	100.0%	265/302	87.7%
check	62.2%	34/35	97.1%	374/408	91.7%
credit	63.9%	31/34	91.2%	116/147	78.9%
credit_card	94.9%	108/112	96.4%	48/156	30.8%
discover	70.0%	7/10	70.0%	5/12	41.7%
dollar	56.4%	24/25	96.0%	648/672	96.4%
hundred	39.0%	8/17	47.1%	38/46	82.6%
interest	67.2%	29/32	90.6%	266/295	90.2%
limit	44.5%	8/11	72.7%	98/106	92.5%
money	35.8%	24/25	96.0%	787/811	97.0%
month	53.1%	37/39	94.9%	708/745	95.0%
percent	61.2%	16/19	84.2%	108/124	87.1%
twenty	25.1%	5/16	31.2%	62/67	92.5%
visa	29.9%	24/26	92.3%	211/235	89.8%
Overall	62.5%	559/613	91.2%	5250/5809	90.4%

Table 5.8: The per-keyword and overall FOM for the combined testing set after 5 iterations of FOM training. (The two numbers under True Hits represent the number of true hits detected over the number of true hits present in the conversations. The two numbers under False Alarms represent the number of false alarms over the number of putative hits generated.)

Keyword	FOM	True Hits	Det %	False Alarms	FA %
account	43.2%	10/12	83.3%	55/65	84.6%
american_express	83.3%	10/12	83.3%	0/10	0.0%
balance	69.1%	12/14	85.7%	44/56	78.6%
bank	56.3%	21/25	84.0%	237/258	91.9%
card	52.8%	78/82	95.1%	84/162	51.9%
cash	73.3%	29/30	96.7%	112/141	79.4%
charge	69.6%	36/37	97.3%	100/136	73.5%
check	64.4%	32/35	91.4%	147/179	82.1%
credit	67.6%	32/34	94.1%	80/112	71.4%
credit_card	95.6%	110/112	98.2%	70/180	38.9%
discover	70.0%	7/10	70.0%	6/13	46.2%
dollar	72.6%	24/25	96.0%	133/157	84.7%
hundred	40.0%	8/17	47.1%	30/38	78.9%
interest	72.6%	27/32	84.4%	53/80	66.2%
limit	48.8%	8/11	72.7%	74/82	90.2%
money	41.9%	20/25	80.0%	194/214	90.7%
month	62.9%	37/39	94.9%	178/215	82.8%
percent	61.0%	16/19	84.2%	109/125	87.2%
twenty	24.8%	6/16	37.5%	56/62	90.3%
visa	38.0%	23/26	88.5%	89/112	79.5%
Overall	65.8%	546/613	89.1%	1851/2397	77.2%

5.6.1 Comparison Between FOM Gradient and Constant Gradient

Experiments were performed to determine those components of FOM training that contribute to the improved performance. The state weights and the centers of the radial basis functions for each state of the keyword models were separately trained. Small improvements are obtained if either only the state weights alone (2.6 percentage points) or if the keyword centers are adapted (3.1 percentage points). In addition, the FOM degrades whenever parameters in the filler model are adapted. In the result shown in Figure 5-10, an additional filler model was added for each keyword and radial basis function centers for these filler models were adapted. This reduced the FOM by 1.1 percentage points. Other experiments resulted in no improvement or reduced FOM when the filler was adapted.

To separately evaluate the effectiveness of using the FOM gradient versus the effectiveness of performing gradient backpropagation, a simpler version of FOM back-propagation training was also explored. Instead of computing FOM gradients as in Figure 5-3, constant gradients were used for true hits and false alarms along with a threshold which adapted keyword centers and state weights only when a putative hit score was sufficiently high. After adjusting the threshold to -50 and the constant gradient values to 0.001 for true hits and -0.001 for false alarms, it was possible to increase the FOM by 2.9 percentage points. This is less than the improvement provided by FOM back-propagation training and required more experimentation to find good threshold and gradient values. It is, however, a simpler alternative to FOM back-propagation training that may be useful in some situations.

5.6.2 The Effect of Using Different Learning Rate Constants

The choice of learning rate constant η is usually determined though performing some initial experiments. If the learning rate is too small, then the models are not being trained as rapidly as possible. If the learning rate is too large, then the amount of change in model parameters may be too large at each step and the model parameters will diverge from optimality. A series of experiments was performed in which four different learning rates were used: 0.125, 0.25, 0.5, and 1.0. The results are shown in Figure 5-11. Notice that no significant differences can be found between using the learning rate of 0.25 and 0.5. However, an overly large learning rate ($\eta = 1.0$) caused the wordspotter performance to degrade. In this thesis, all learning rates in FOM training experiments were set to 0.25.

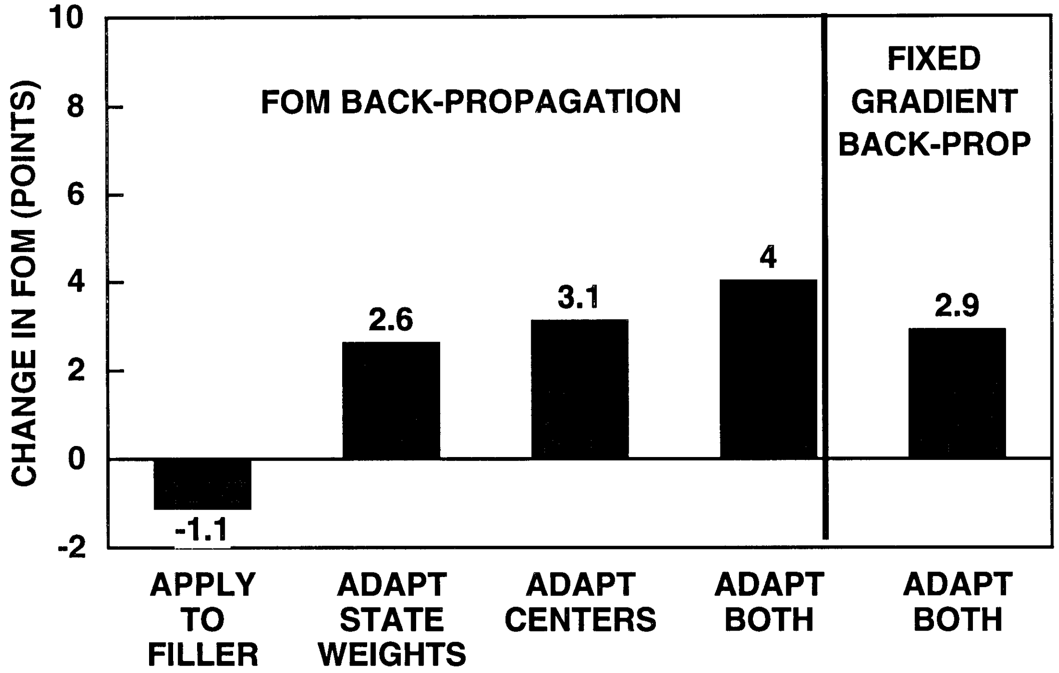


Figure 5-10: Change in FOM after training 1) the filler centers, 2) the keyword centers, 3) the keyword states, 4) the keyword centers and states, and 5) the keyword centers and states using constant gradient.

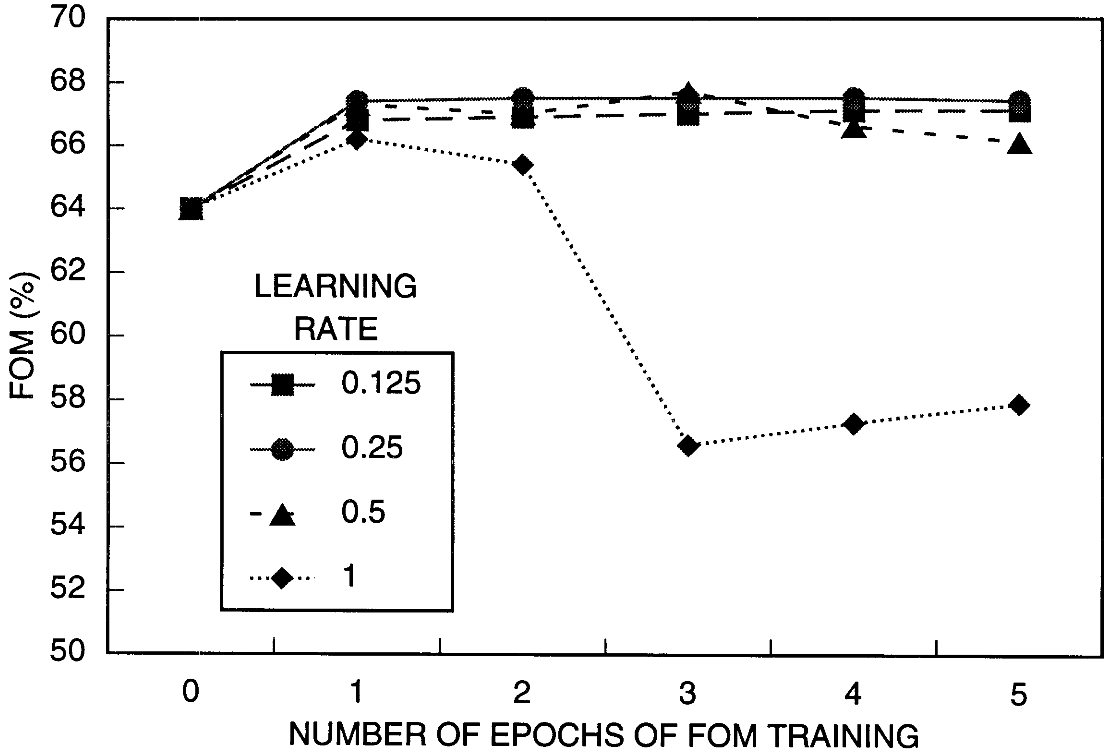


Figure 5-11: The change in FOM through five epochs of FOM training using different learning rate constants. Notice that using too large a learning rate (1.0) caused the wordspotter performance to degrade.

5.7 Chapter Summary

Detection of target signals embedded in a noisy background is a common and difficult problem distinct from the task of classification. The evaluation metric of a spotting system, called Figure of Merit (FOM), is also different from the classification accuracy used to evaluate classification systems. FOM training uses a gradient which directly reflects the impact of a putative hit on the FOM to modify the parameters of the spotting system and directly maximizes FOM. FOM training does not require careful adjustment of thresholds and target values and has been applied to improve a wordspotter's FOM from 62.5% to 65.8%.

Figure of Merit discriminative training provided a 2.6 percentage point improvement for the male speakers (64.8% to 67.4%) and a 4.1 percentage point increase in the female testing set FOM (61.5% to 65.6%). Best performance was obtained when Figure of Merit training was used to move radial basis function centers and to adjust state weights. Figure of Merit training is a general technique that can applied to any spotting task where a set of targets must be discriminated from background inputs. Such tasks including detecting faults in large systems, detecting abnormal heart beats, and detecting targets in satellite images.

Chapter 6

Voice Transformations

Speech recognizers provide good performance for most users but the error rate is often large for a small percentage of speakers who are different from those speakers used for training. One expensive solution to this problem is to gather more training data in an attempt to sample these outlier users. A second solution, explored in this thesis, is to artificially enlarge the number of training speakers by transforming the speech of existing training speakers. This approach is similar to enlarging the training set for optical character recognition by warping the training character images [17], but is more difficult because continuous speech has a much larger number of dimensions (e.g. linguistic, phonetic, temporal, spectral) that differ across speakers. In this thesis, the use of simple linear spectral warping to enlarge the 48-speaker training database used for wordspotting is explored. The Figure of Merit was increased by 6.4 percentage points (from 67.4% to 73.8%) for male speakers and 4.9 percentage points (from 65.6% to 70.5%) for female speakers.

6.1 Introduction

Speech recognizers, optical character recognizers, and other types of pattern classifiers used for human interface applications often provide good performance for most users. Performance is often, however, low and unacceptable for a small percentage of *outlier* users who are presumably not represented in the training data. One expensive solution to this problem is to obtain more training data in the hope of including users from these outlier classes. Other approaches already used for speech recognition are to use input features and distance metrics that are relatively invariant to linguistically unimportant differences between speakers and to adapt a recognizer for individual speakers. Speaker adaptation is difficult for wordspotting and with poor outlier users because the recognition error rate is high and speakers often can not be prompted to recite standard phrases that can be used for adaptation. An alternative approach, that has not been fully explored for speech recognition, is to artificially expand the number of training speakers using voice transformations. Transforming the speech of one speaker to make it sound like that of another is difficult because speech varies across many difficult-to-measure dimensions including linguistic, phonetic, duration, spectra, style, and accent. The transformation task is thus more difficult than in optical character recognition where a small set of warping functions can be successfully applied to character images to enlarge the number of training images [17]. This chapter demonstrates how a transformation accomplished by warping the spectra of training speakers to create more training data can improve the performance of a whole-word wordspotter on a large spontaneous-speech database.

6.2 Speaker Adaptation Approaches

Traditional methods of speaker adaptation are divided into two different types: active speaker adaptive methods or passive speaker selection methods. In active adaptation, the model parameters of the speech recognition system are adjusted to better match the input from the current speaker. In passive speaker selection method, a model is selected from a set of models based on the similarity between the current speaker and the training data used to train the selected speaker model. For this particular database, the limited number of training tokens per speaker makes training separate models for different clusters of speakers impractical. Another possibility is to estimate some parameters related to the voice characteristics of the incoming speaker and adapt the model parameters to the incoming speaker's voice. In this thesis, a spectral scaling approach is explored. It is well known that formant frequencies are related to a speaker's vocal tract length [77]. In fact, if the vocal tract is modelled as a series of acoustic tubes, then the formant frequencies are inversely proportional to the vocal tract length.

6.2.1 Changing the Frequency Scale

A method of transforming a speaker's voice which utilizes the Sinusoidal Transform Analysis/Synthesis System (STC) described in [61] has been implemented. This technique attempts to transform one speaker's speech pattern to that of a different speaker. The STC generates a 512-point spectral envelope of the input speech 100 times a second and also separates pitch and voicing information. Separation of vocal tract characteristic and pitch information has allowed the implementation of pitch and time transformations in previous work [61]. The system has been modified to generate and accept a spectral envelope file from an input speech sample. We informally explored different techniques to transform the spectral envelope to generate more varied training examples by listening to transformed speech. This resulted in the following algorithm that transforms a speaker's voice by scaling the spectral envelope of training speakers.

An outline of the steps in the procedure is presented below:

1. The original credit card database is upsampled to 10,000 Hz sampling rate from the original 8,000 Hz sampling rate since the STC system requires speech input with the sampling rate of 10,000 Hz.
2. The STC system processes the upsampled sample files and generates a 512 point spectral envelope of the input speech waveform at a frame rate of 100 frames a second and with a window length of approximately 2.5 times the length of the pitch period.
3. A new spectral envelope is generated by linearly expanding or compressing the spectral axis. Each spectral point is identified by its index, ranging from 0 to 511. To transform a spectral profile by a factor of 2, the new spectral value at frequency f is generated by taking a local average of the spectral values around the original spectral profile at the frequency of $0.5f$. The transformation process is illustrated in Figure 6-1.
4. The transformed spectral value is then used to resynthesize a speech waveform using the vocal tract excitation information extracted from the original file.
5. The new sample files are downsampled from 10,000 Hz. to 8,000 Hz. The new samples files are then converted into the mel-scaled cepstra values as described in Figure 4-1.

The procedure outlined above enables the generation of conversation recordings that have been spectrally scaled. Through informal listening tests, it was determined that ratios from 0.9 to 1.1 were reasonable for compressing or expanding the frequency scale. However, as an on-line adaptation approach, the procedure outlined above is too slow. The STC algorithm takes approximately 5 times real time to process input speech waveforms

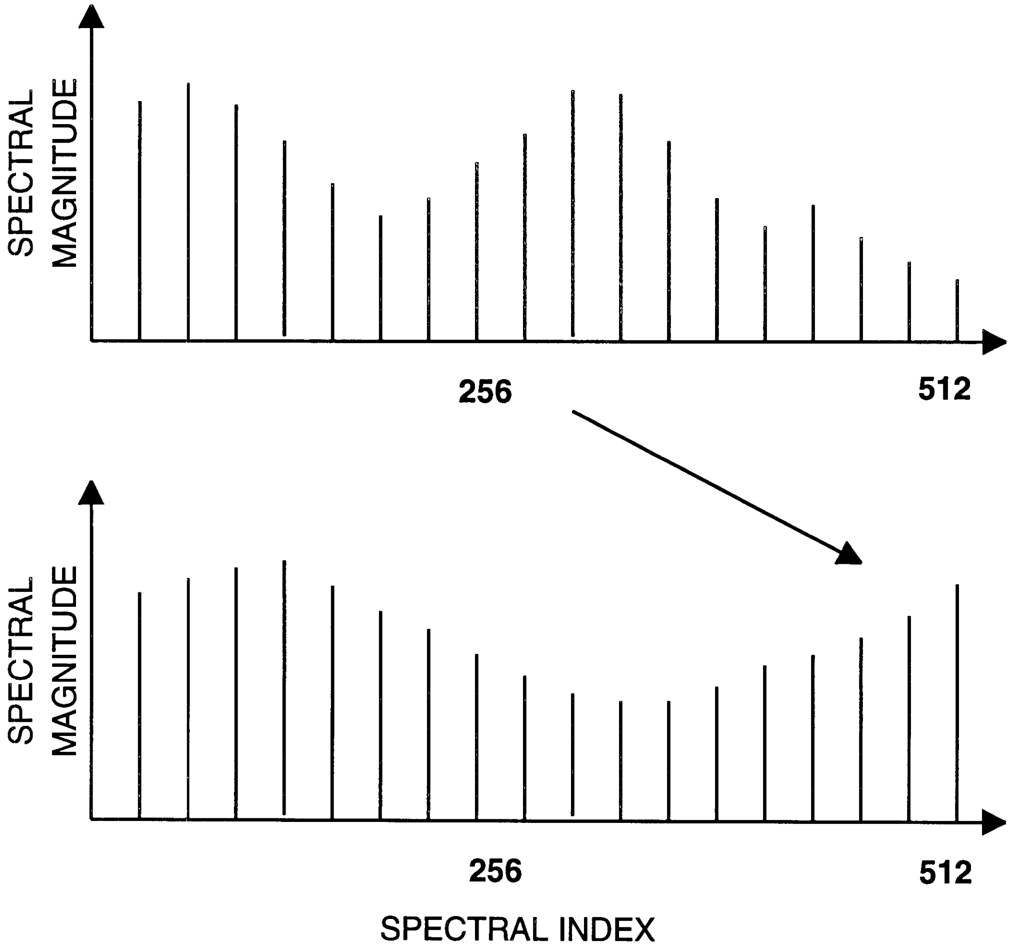


Figure 6-1: An example of the spectral transformation algorithm where the original spectral envelope frequency scale is expanded by 2.

on a Sparc 10. To allow for faster transformation, the preprocessing procedure described in Figure 4-1 was modified to rescale the frequency scale. The mapping procedure similar to Step 3 listed above was used to remap the FFT coefficients that are calculated during the mel-scaled filter bank stage of the preprocessing process. All other steps in preprocessing remain the same. This procedure is much faster than relying on the STC transform algorithm, taking approximately 1/10th real time on a Sparc 10. Thus, on-line transformation of speech would be possible with this procedure. However, the STC approach has the advantage of allowing the transformed speech to be listened to and thus is useful for determining the appropriate ratios for the rescaling of the frequency scale.

6.2.2 Impact of Spectral Transformation on FOM

After the spectral transformation algorithm was implemented as a part of the preprocessing procedure, a set of experiments which varied the spectral transform ratio over a range to determine the effect of spectral transformation ratios on individual FOM was performed. Figure 6-2 shows the variations in FOM over a range of spectral transform ratios for a male speaker and a female speaker respectively. For each speaker, a suitable range of transformation ratios exists that results in a high FOM. The FOM falls off as the spectral transform ratio is moved away from this range. While the magnitude of change in FOM varies among the test speakers, most speakers exhibited a tendency to be spotted better at a particular spectral transform ratio range.

When the spectral transform ratios with the highest FOM for each speaker are chosen *a posteriori*, the FOM improves from 71.3% to 73.7% for the male testing set and 67.7% to 69.5% for the female testing set. Such a gain is significant, although such a gain can only be achieved in real life if the appropriate ratio can be determined for each individual speaker. Another approach is to spot each conversation several times, each time using a different scaling ratio. The resulting putative hits can then be combined. Two sets of experiments in combining the putative hits from multiple runs of the spotting algorithm using different spectral ratios were conducted. In the first method, the putative hits of the same word at the same time in the conversation from wordspotters spotting at different frequency scales were grouped and the putative hit with the highest score was selected. With this method, the FOM for the male speakers decreased from 71.3% to 70.8% and the female speakers' FOM changed from 67.7% to 67.9%. In a second set of experiments, the putative hits that

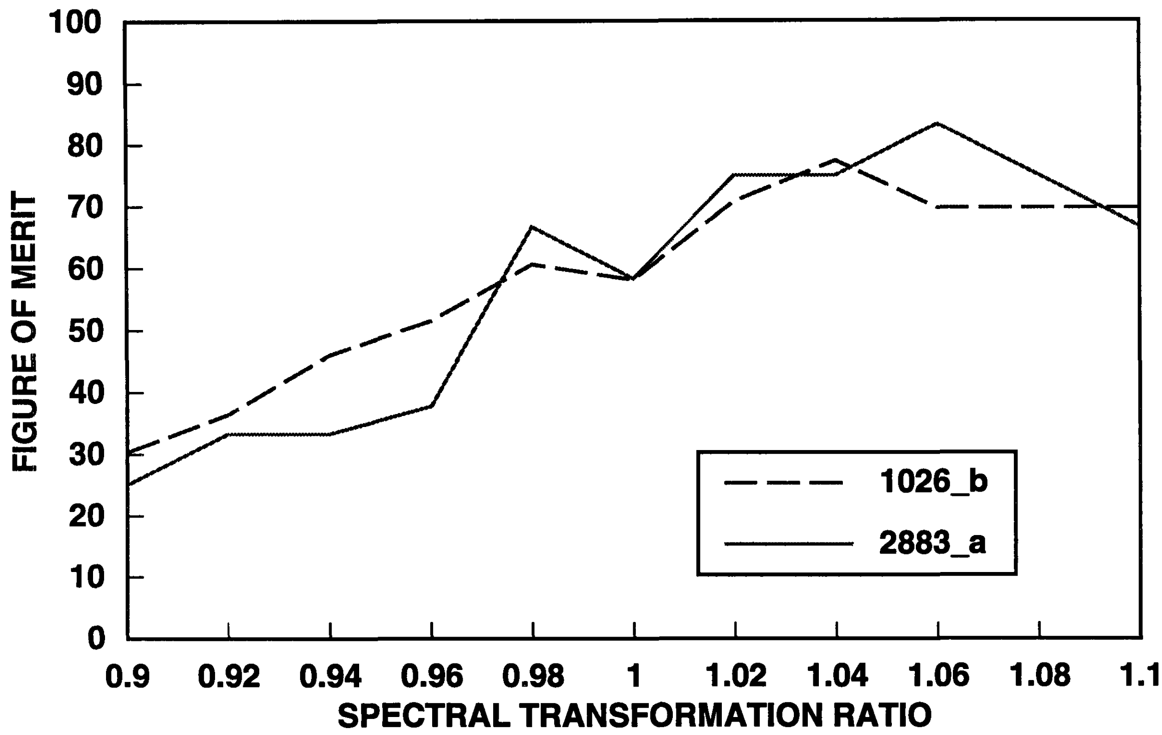


Figure 6-2: Variation of Figure of Merit for one male speaker (1026_b) and one female speaker(2883_a) over a range of spectral transform ratios.

were in the same time period in the conversation were grouped together and the average score was used as the putative hit score. The FOM for the male speakers increased from 71.3% to 71.9% and the female speaker's FOM changed from 67.7% to 68.0%. Both methods failed to significantly improve the FOM, indicating the difficulty of combining the putative hits generated from several frequency scales.

6.2.3 Determining a Speaker's Frequency Scale

An alternative approach is to estimate the appropriate frequency scale for an input speaker and to use the estimated scale in processing the speech from the input speaker. A method of estimating the proper frequency scale for each speaker was developed. The active speech segments of each speaker are used for analysis. Each segment is processed by the program *formant* from Entropic Research Laboratory Incorporated. The program *formant* generates the formant frequencies, bandwidths, pitch, voicing probability, and a cross-correlation measure [70]. Only strongly voiced portions of the segments are used by choosing frames in which both the voicing probability and the cross-correlation measures are greater than 0.8.

The formant frequencies of all the voiced portions are then averaged to derive the average formant frequencies for each speaker.

After formant frequencies have been calculated for each speaker, the appropriate frequency scale can be estimated in two ways. One way is to calculate the average F3 frequency of all the training speakers of the same gender, and then calculate the ratio α which satisfies the following equation:

$$\bar{F_3} = \alpha \cdot F_3, \tag{6.1}$$

where $\bar{F_3}$ is the average $F3$ frequency over the population and F_3 is the average formant frequency for a test speaker. This method was based on the observation that while formant one and formant two vary a great deal from vowel to vowel, formant three is relatively stable among different vowels. Thus, the variation in F3 between speakers can be more characteristic of the speaker than the vowel and thus can be used as a measure of each speaker's vocal tract characteristics. In the second method, the ratio α which minimizes the following expression:

$$\sum_{i=1}^{3} (\alpha \frac{F_i}{\bar{F_i}} - 1)^2 \tag{6.2}$$

is derived through least square estimation.

In a preliminary set of experiments, using the ratio derived from F3 alone, the FOM increased from 71.3% to 73.4% on the male testing set. Using the ratio α derived from all three formants as described in Equation 6.2, the FOM increased from 71.3% to 73.3% on the male testing set. Although the increase in FOM is significant, this on-line adaptation of the frequency scale approach suffers from the necessity of estimating the frequency scale. During the experiments, all strong vowel segments from a given speaker were used to calculate the average formant frequencies. The formant frequencies can not be estimated robustly with a small number of frames and thus this approach is not suitable for improving performance rapidly for a novel speaker without supervised training and saying adaptation phrases.

Another approach to improve robustness toward speaker variability that does not require any on-line adaptation at all was tried. The approach utilizes *a priori* knowledge about the source of variability in speech to artificially enrich the training set. This approach of *voice transformation* is introduced in the next section.

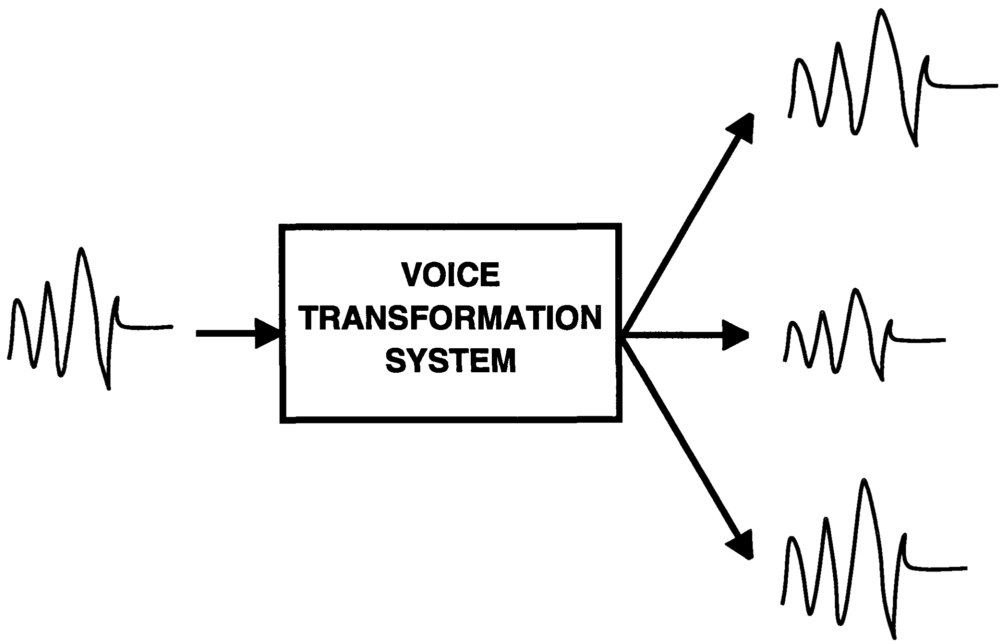


Figure 6-3: Generating more training data by artificially transforming original speech training data.

6.3 Creating More Training Data with Voice Transformations

6.3.1 Introduction

Speaker adaptation is difficult for wordspotting because error rates are high and speakers often can not be prompted to verify adaptation phrases. This thesis introduces a new approach of increasing performance across speakers by using voice transformation techniques to generate more varied training examples of keywords as shown in Figure 6-3. Other researchers have used speaker transformation techniques to produce more natural synthesized speech [31, 47], but using speaker transformations to generate more training data is novel.

Linear warping in the spectral domain is correct when the vocal tract is modelled as a series of lossless acoustic tubes and the excitation source is at one end of the vocal tract [77]. Wakita showed that if the vocal tract is modelled as a series of equal length, lossless, and concatenated acoustic tubes, then the ratio of the areas between the tubes determines the relative resonant frequencies of the vocal tract, while the overall length of the vocal tract is inversely proportional to the formant frequencies. Linear warping of the frequency scale is used as the voice transformation method in this thesis.

6.3.2 Experimental Setup

Preliminary research was conducted using linear scaling with spectral ratios ranging from 0.6 to 1.8 to alter test utterances. After listening to the STC transformed speech and observing spectrograms of the transformed speech, it was found that speech transformed using ratios between 0.9 and 1.1 is reasonably natural and can represent speech without introducing artifacts. Using discriminative training techniques such as FOM training carries the risk of overtraining the wordspotter on the training set and obtaining poor results on the testing set. To delay the onset of overtraining, each training set conversation was artificially transformed during each epoch using a different random linear transformation ratio. The transformation ratio used for each conversation is calculated using the following formula:

$$ratio \equiv \alpha + N(0, 0.06), \tag{6.3}$$

where α is the transformation ratio that matches each training speaker to the average of the training set speakers, and N is a normally distributed random variable with a mean of 0.0 and standard deviation of 0.06.

For each training conversation, the long term averages of formant frequencies for formants 1, 2, and 3 are calculated. A least square estimation is then performed to match the formant frequencies of each training set conversation to the group average formant frequencies. The transformation calculation is performed as described in Equation 6.2. The transform ratio for each individual conversation is calculated to improve the naturalness of the transformed speech. In preliminary experiments, each conversation was transformed with fixed ratios of 0.9, 0.95, 1.05, and 1.1. However, for a speaker with already high formant frequencies, pushing the formant frequencies higher may make the transformed speech sound unnatural. By incorporating the individual formant matching ratio into the transformation ratio, speakers with high formant frequencies are not transformed to very high frequencies and speakers with low formant frequencies are not transformed to even lower formant frequency ranges.

Male and female conversations from the NIST credit card database were used separately to train separate wordspotters. Both the male and the female partition of data used 24 conversations for training and 11 conversations for testing. Keyword occurrences were extracted from each training conversation and used as the data for initialization of the neural network wordspotter. Also, each training conversation was broken up into sentence length segments to be used for embedded reestimation in which the keyword models are joined with the filler models and the parameters of all the models are jointly estimated. After embedded reestimation, Figure of Merit training as described in Chapter 5 was performed for up to 10 epochs. During each epoch, each training conversation is transformed using a transform ratio randomly generated as described above. The performance of the wordspotter after each iteration of training is evaluated on both the training set and the testing set.

6.3.3 Wordspotting Results

Training and testing set FOM scores for the male speakers and the female speakers are shown in Figure 6-4 and Figure 6-5 respectively. The x axis plots the number of epochs of FOM training where each epoch represents presenting all 24 training conversations once. The FOM for wordspotters trained with the normal training conversations and wordspotters trained with artificially expanded training conversations are shown in each plot. After the first epoch, the FOM improves significantly. With only the original training conversations (normal), the testing set FOM rapidly levels off while the training set FOM keeps on improving.

When the training conversations are artificially expanded, the training set FOM is below the training set FOM from the normal training set due to more difficult training data. However, the testing set FOM continues to improve as more epochs of FOM training are performed. When comparing the FOM of wordspotters trained on the two sets of data after ten epochs of training, the FOM for the expanded set was 2.9 percentage points above the normal FOM for male speakers and 2.5 percentage points above the normal FOM for female speakers.

6.3.4 Artificial Variation in Initialization

The improved result obtained by using artificially transformed speech during FOM training was followed by a set of experiments in using artificially transformed speech during initialization. The distribution of keywords among the training speakers is uneven and there are many words that are only spoken by a few speakers in the training set. For example, the word *limit* was spoken by only 5 out of 24 male training set speakers. Similarly, the word *account* was spoken by only 3 out of 24 female training speakers. Table 2.5 shows

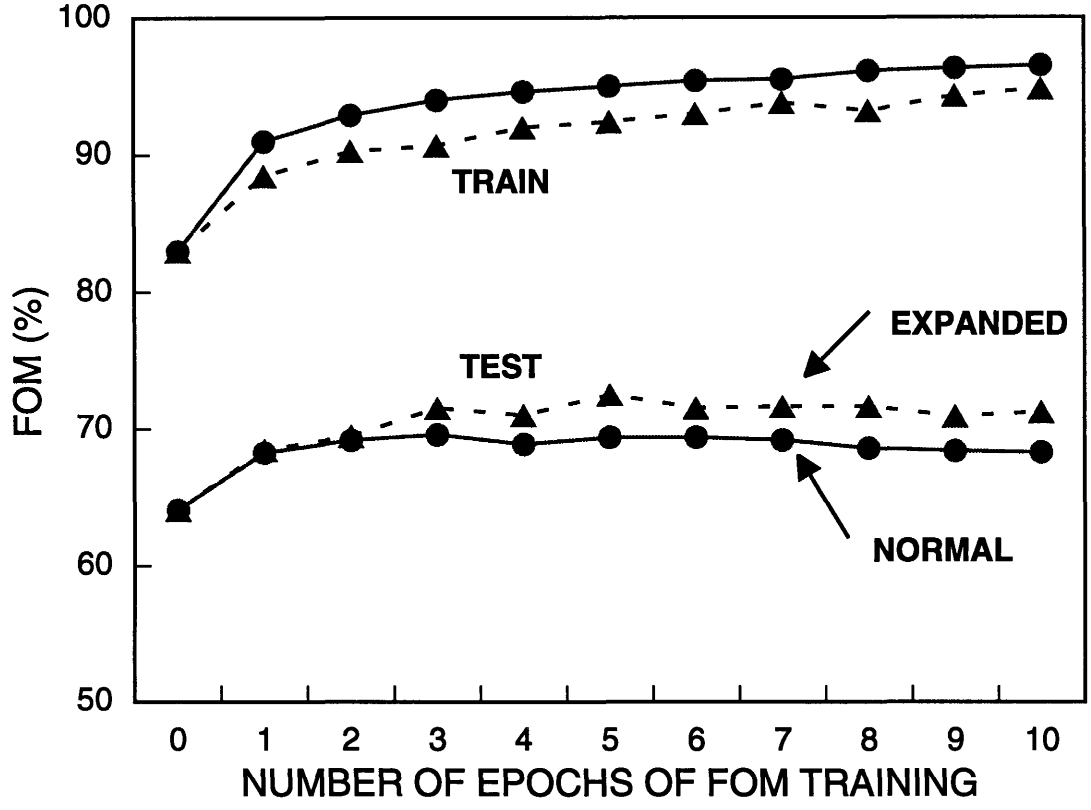


Figure 6-4: Average detection accuracy (FOM) for the male training and testing set versus the number of epochs of FOM training.

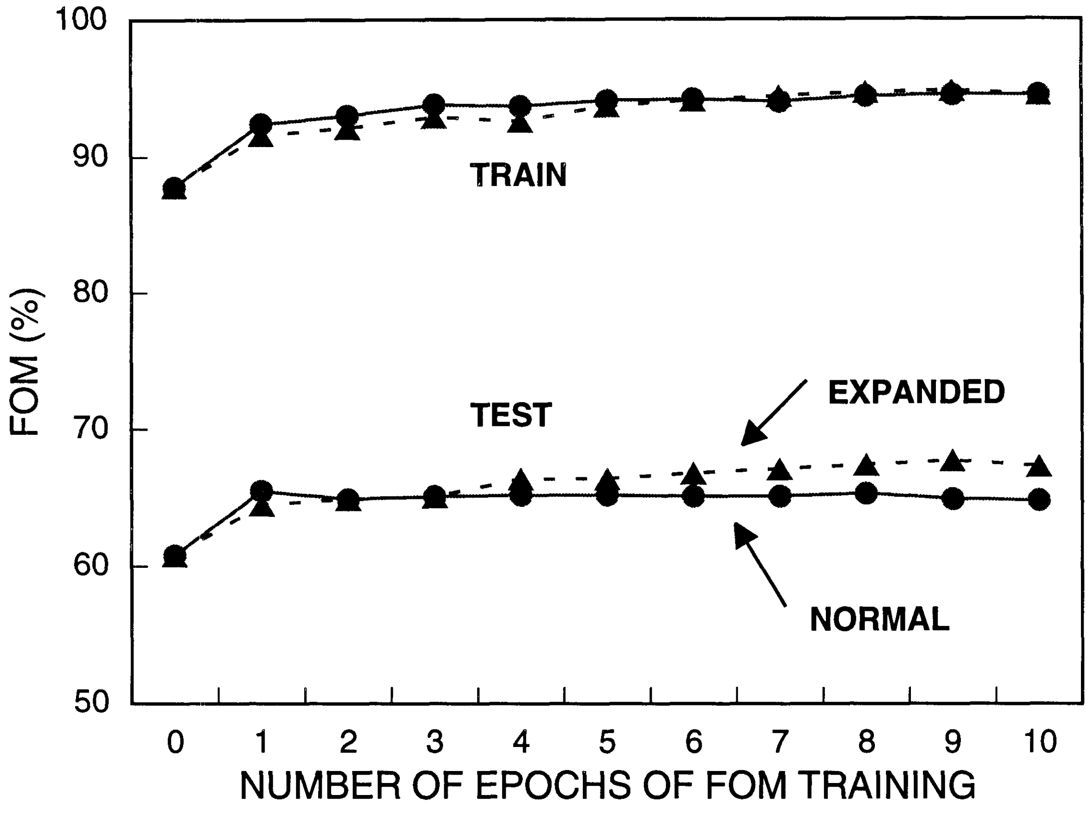


Figure 6-5: Average detection accuracy (FOM) for the female training and testing set versus the number of epochs of FOM training.

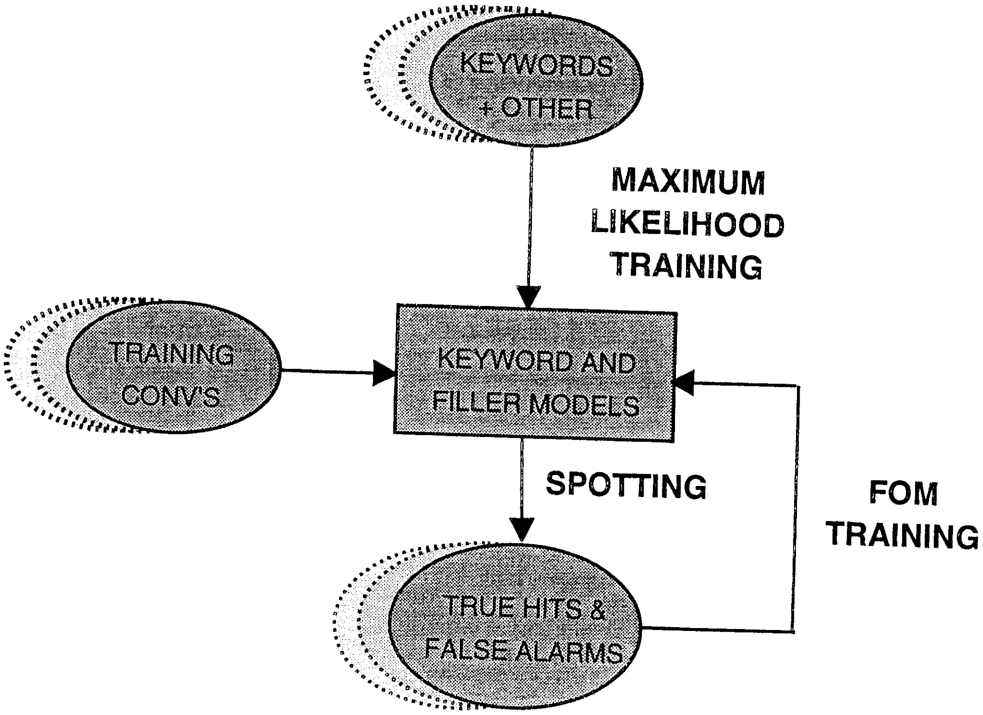


Figure 6-6: The training methodology for Figure of Merit training in combination with voice transformations. The models are first trained using maximum likelihood estimation on original data and artificially transformed data. Then the models are used to spot artificially transformed conversations and the resulting true hits and false alarms are used to train the models discriminatively.

that certain keywords were spoken by very few speakers. Thus, the keyword models which are initialized with the excised keyword examples may not be sufficiently general to spot the same keyword spoken by new speakers. To improve the generalization of the keyword models, it was decided that artificially transformed speech would be used starting at the first stage of maximum likelihood training. The new training methodology is described in Figure 6-6.

Experiments were performed in which the amount of artificially transformed speech was varied to determine the effect of adding artificial speech. In addition to the baseline experimental results from Chapter 5, 3 sets of experiments were conducted using varying amounts of artificially transformed speech. The experimental setup for each experiment is described in Table 6.1. The transformation ratios were chosen so that with no transformed speech during initialization, no transformed speech is used during FOM training either. When a large amount of transformed speech are used during initialization, then speech transformed with greater variability are used during FOM training.

Table 6.1: The amount of artificial speech added to the training set and the standard deviation of the normal function used to transform conversations during FOM training.

Amount of Speech	Std. Deviation of Normal Function during FOM training
0X	0.0
1X	0.0125
3X	0.0375
5X	0.06

The experimental results are shown in Table 6.2 and Table 6.3 for the male testing set and the female testing set respectively. The results are also plotted in Figure 6-7 and Figure 6-8 to allow visualization. For the male testing set, one trend which emerges is that adding artificial training data does not improve FOM during the maximum likelihood estimation training stage (Estimation and Embedded Estimation). In fact, adding more artificial training data tends to decrease FOM during the maximum likelihood training stage. But after five iterations of Figure of Merit Training, it is clear that the more artificial data added, the better the FOM. This result is due to the fact that adding more artificial training data during maximum likelihood training makes the output distributions of the keyword state models broader. These keyword models thus are more likely to pick up false alarms. Since only the keyword examples are used to train each keyword model during maximum likelihood training, such an effect is understandable.

However, during Figure of Merit training, since both true hits and false alarms are used to adjust the parameters of the keyword models, the keyword models are trained to discriminate between true hits and false alarms. Thus the keyword models become more selective and do not generate as many false alarms, meanwhile, the better generalization ability is still retained. For example, by comparing Table 5.2 and Table 6.4, it becomes clear that the detection rates of most keywords are increased by adding 5 transformed copies of the original data into the training set. Similar trends can be observed for the female speakers in Table 6.5. Furthermore, keywords for which very few speakers are represented saw big increases in FOM. For example, the word *discover* has only four male speakers in the training set. The FOM improved from 0.0% (Table 5.2) to 50.0% (Table 6.4) by performing voice transformations in addition to Figure of Merit training. Similarly, the word *account* has only three female speakers in the training set. The FOM improved from

Table 6.2: Male testing set FOM with varying amount of artificial training data.

	Training Stage		
Amt. of Artificial Data	Estimation	Embedded Estimation	FOM Training (5 Iter.)
0X	56.0%	64.8%	67.4%
1X	54.4%	63.8%	69.6%
3X	52.1%	61.7%	71.4%
5X	53.4%	61.6%	73.8%

Table 6.3: Female testing set FOM with varying amount of artificial training data.

	Training Stage		
Amt. of Artificial Data	Estimation	Embedded Estimation	FOM Training (5 Iter.)
0X	52.7%	61.5%	65.6%
1X	55.5%	61.7%	70.1%
3X	53.9%	63.9%	69.6%
5X	51.4%	61.1%	70.5%

39.8% (Table 5.5) to 72.7% (Table 6.5) by performing voice transformation in addition to Figure of Merit training. Such big gains demonstrate the particular effectiveness of using voice transformations to increase variability for infrequently occurring keywords.

The overall FOM for the combined male/female testing set is shown in Table 6.6. The Figure of Merit is now 71.9%. In addition, the added variability improved the effect of Figure of Merit training. The overall results of combining Figure of Merit training and voice transformation is shown in Table 6.7. The overall FOM improved from 62.5% to 71.9%. Also, beside the word *american_express* which was easy to spot, the FOMs for all other keywords were improved. The word *account* improved the most with a 26.5 percentage point increase.

6.4 Training a Wordspotter with Synthetic Speech

An alternative approach to enlarging the training data is to synthesize speech. Blomberg et al. have tried using synthetic speech to create examples in a speech recognition sys-

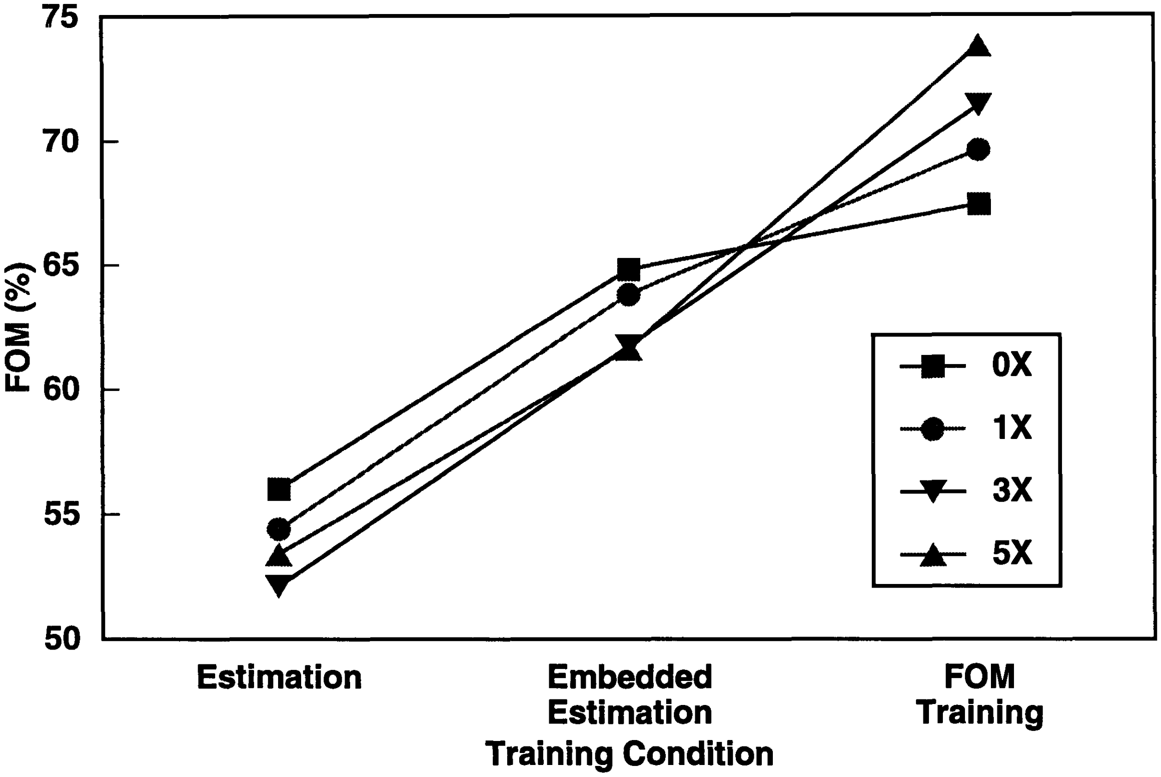


Figure 6-7: FOM after estimation, embedded estimation, and FOM training with 0X, 1X, 3X, and 5X of artificial data added to the original male training data.

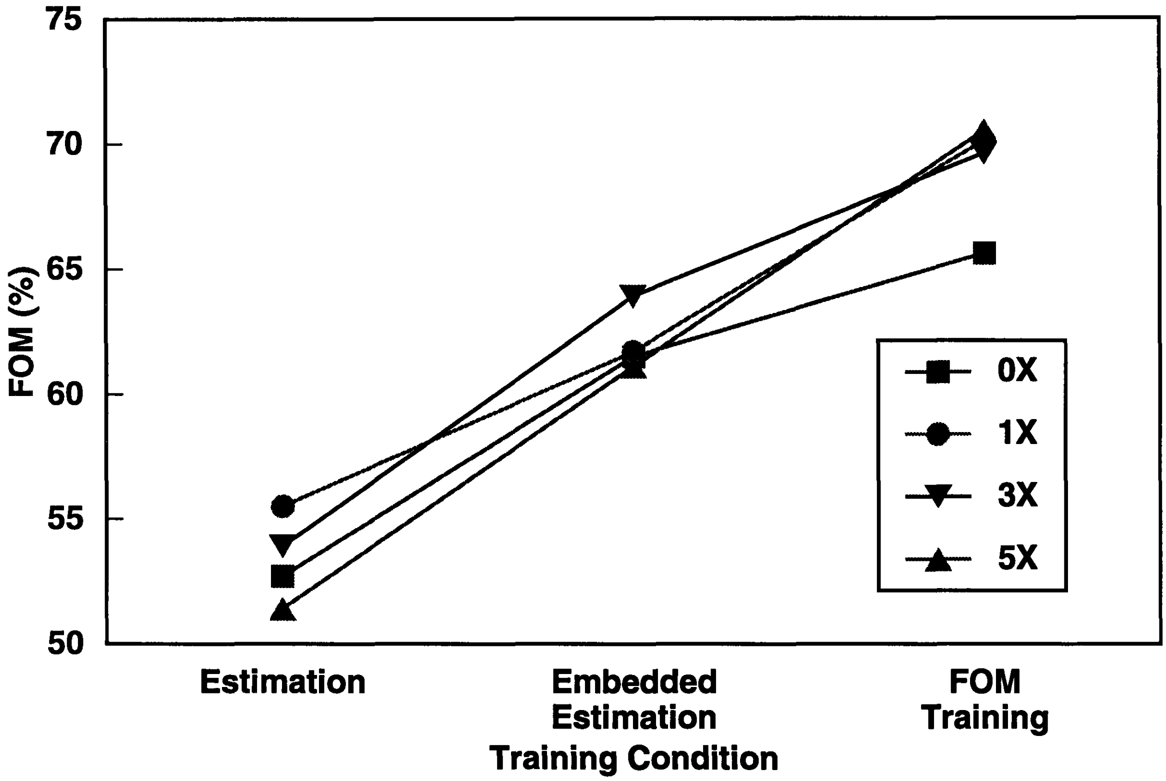


Figure 6-8: FOM after estimation, embedded estimation, and FOM training with 0X, 1X, 3X, and 5X of artificial data added to the original female training data.

Table 6.4: The per-keyword and overall FOM for the male testing set after 5 iterations of FOM training with an artificially enlarged training set. (The two numbers under True Hits represent the number of true hits detected over the number of true hits present in the conversations. The two numbers under False Alarms represent the number of false alarms over the number of putative hits generated.)

Keyword	FOM	True Hits	Det %	False Alarms	FA %
account	69.1%	7/8	87.5%	59/66	89.4%
american_express	87.5%	7/8	87.5%	0/7	0.0%
balance	66.9%	7/7	100.0%	32/39	82.1%
bank	69.4%	17/19	89.5%	109/126	86.5%
card	64.6%	48/49	98.0%	143/191	74.9%
cash	92.8%	15/15	100.0%	93/108	86.1%
charge	81.3%	20/20	100.0%	75/95	78.9%
check	79.6%	27/27	100.0%	95/122	77.9%
credit	66.0%	12/13	92.3%	55/67	82.1%
credit_card	96.3%	72/73	98.6%	35/107	32.7%
discover	50.0%	1/2	50.0%	2/3	66.7%
dollar	86.6%	14/14	100.0%	116/130	89.2%
hundred	51.9%	10/14	71.4%	53/63	84.1%
interest	86.4%	11/12	91.7%	97/108	89.8%
limit	46.0%	4/4	100.0%	66/70	94.3%
money	53.3%	11/11	100.0%	137/148	92.6%
month	73.2%	22/22	100.0%	103/125	82.4%
percent	54.7%	10/10	100.0%	93/103	90.3%
twenty	40.5%	7/10	70.0%	99/106	93.4%
visa	31.9%	14/17	82.4%	79/93	84.9%
Overall	73.8%	336/355	94.6%	1541/1877	82.1%

Table 6.5: The per-keyword and overall FOM for the female testing set after 5 iterations of training with an artificially enlarged training set. (The two numbers under True Hits represent the number of true hits detected over the number of true hits present in the conversations. The two numbers under False Alarms represent the number of false alarms over the number of putative hits generated.)

Keyword	FOM	True Hits	Det %	False Alarms	FA %
account	72.7%	3/4	75.0%	10/13	76.9%
american_express	75.0%	3/4	75.0%	1/4	25.0%
balance	62.0%	6/7	85.7%	27/33	81.8%
bank	82.8%	6/6	100.0%	62/68	91.2%
card	75.2%	33/33	100.0%	67/100	67.0%
cash	54.4%	15/15	100.0%	81/96	84.4%
charge	79.0%	16/17	94.1%	42/58	72.4%
check	24.6%	7/8	87.5%	67/74	90.5%
credit	86.5%	21/21	100.0%	66/87	75.9%
credit_card	95.7%	39/39	100.0%	40/79	50.6%
discover	87.5%	7/8	87.5%	17/24	70.8%
dollar	67.3%	10/11	90.9%	61/71	85.9%
hundred	33.3%	1/3	33.3%	1/2	50.0%
interest	84.8%	19/20	95.0%	23/42	54.8%
limit	51.8%	5/7	71.4%	135/140	96.4%
money	29.2%	13/14	92.9%	209/222	94.1%
month	49.4%	15/17	88.2%	129/144	89.6%
percent	86.8%	8/9	88.9%	44/52	84.6%
twenty	22.4%	5/6	83.3%	88/93	94.6%
visa	58.0%	9/9	100.0%	118/127	92.9%
Overall	70.5%	241/258	93.4%	1288/1529	84.2%

Table 6.6: The per-keyword and overall FOM for the combined testing set after 5 iterations of training with an artificially enlarged training set. (The two numbers under True Hits represent the number of true hits detected over the number of true hits present in the conversations. The two numbers under False Alarms represent the number of false alarms over the number of putative hits generated.)

Keyword	FOM	True Hits	Det %	False Alarms	FA %
account	68.6%	10/12	83.3%	69/79	87.3%
american_express	83.3%	10/12	83.3%	1/11	9.1%
balance	65.0%	13/14	92.9%	59/72	81.9%
bank	71.3%	23/25	92.0%	171/194	88.1%
card	68.8%	81/82	98.8%	210/291	72.2%
cash	76.4%	30/30	100.0%	174/204	85.3%
charge	76.8%	36/37	97.3%	117/153	76.5%
check	68.6%	34/35	97.1%	162/196	82.7%
credit	78.2%	33/34	97.1%	121/154	78.6%
credit_card	96.3%	111/112	99.1%	75/186	40.3%
discover	80.0%	8/10	80.0%	19/27	70.4%
dollar	77.7%	24/25	96.0%	177/201	88.1%
hundred	52.8%	11/17	64.7%	54/65	83.1%
interest	79.8%	30/32	93.8%	120/150	80.0%
limit	50.0%	9/11	81.8%	201/210	95.7%
money	39.6%	24/25	96.0%	346/370	93.5%
month	63.1%	37/39	94.9%	232/269	86.2%
percent	69.2%	18/19	94.7%	137/155	88.4%
twenty	31.2%	12/16	75.0%	187/199	94.0%
visa	36.8%	23/26	88.5%	197/220	89.5%
Overall	71.9%	577/613	94.1%	2829/3406	83.1%

Table 6.7: The per-keyword and overall FOM for the combined testing set for the baseline system, the system trained with 5 iterations of FOM training on expanded data, and the change in FOM.

Keyword	FOM Before	FOM After	Change in FOM
account	42.1%	68.6%	26.5%
american_express	83.3%	83.3%	0.0%
balance	62.7%	65.0%	2.3%
bank	59.6%	71.3%	11.7%
card	50.6%	68.8%	18.2%
cash	62.7%	76.4%	13.7%
charge	73.7%	76.8%	3.1%
check	62.2%	68.6%	6.4%
credit	63.9%	78.2%	14.3%
credit_card	94.9%	96.3%	1.4%
discover	70.0%	80.0%	10.0%
dollar	56.4%	77.7%	21.3%
hundred	39.0%	52.8%	13.8%
interest	67.2%	79.8%	12.6%
limit	44.5%	50.0%	5.5%
money	35.8%	39.6%	3.8%
month	53.1%	63.1%	10.0%
percent	61.2%	69.2%	8.0%
twenty	25.1%	31.2%	6.1%
visa	29.9%	36.8%	6.9%
Overall	62.5%	71.9%	9.4%

tems [4, 3]. In one experiment, synthesized speech samples of 26 Swedish words were used as templates in a dynamic time warping recognition system. It was found that the synthetic word templates performed adequately, although worse than any of the human speech templates even when some adaptation is performed [4]. Since the task in the experiment was isolated word recognition, the variability which exists between different words in the vocabulary may be large enough so that even synthetic speech examples can model the differences between different words.

Experiments were conducted in which keyword examples were generated with an artificial voice synthesizer [21]. Since four male voices are currently available in this synthesizer, four examples of each keyword were generated. Telephone line simulators were used to generate the spectral distortions and noises that occur in telephone transmission [36]. Three different types of channel conditions were used on each keyword sample, resulting in a total of 12 examples for each keyword. These twelve keyword examples and samples of random speech generated from the synthesizer are used to train keyword models and the filler model respectively using maximum likelihood estimation.

6.4.1 Experimental Results

The wordspotter trained with synthetic speech had a FOM of 6% on the male testing set. Since this level of performance is much lower than that of the baseline wordspotter, training with synthetic speech was not pursued further. Upon examining the per-word result of the wordspotter, it was found that the detection rate for all words was only 18.6%. Apparently the keyword models trained on synthetic speech from this particular speech synthesizer do not model the keywords in natural speech well enough. Other synthesizers such as DECtalk could have been explored but this line of research was not pursued since it would fall outside the scope of this thesis [72].

6.5 Summary

Lack of training data has always been a constraint in training speech recognizers. This research presents a voice transformation technique which increases the variety among training samples. The resulting more varied training set provided up to 6.1 percentage points of improvement in the Figure of Merit of a high performance wordspotter. Much bigger improvements occurred on words with limited number of training speakers. For example, the FOM for the word *discover* increased from 0.0% to 50.0% on the male testing set and the FOM for the word *account* increased from 39.8% to 72.7% on the female testing set. This technique can also be applied to other speech recognition systems such as continuous speech recognition, speaker identification, and isolated word recognition.

Chapter 7

Human Wordspotting Experiments

7.1 Introduction

One question which frequently arises in speech recognition research is the performance of the machine versus the performance of human beings. Human beings have so far shown much higher performance than machines in speech recognition and dictation. For example, Ebel and Picone recently tested human listening performances on high quality read speech with unconstrained vocabulary [19]. They played the original speech segments and speech segments corrupted with signal to noise ratios of 22 dB, 16 dB and 10 dB and found that human performance is an order of magnitude above the machine performance. On average, the human word error rate was in the range of 2%. The best performance achieved by continuous speech recognizers was in the range of 7 to 12% [57].

There has been no published report on the performance of human beings on the task of wordspotting for the Switchboard credit card database. To analyze the effectiveness of the wordspotting system, excised segments of the speech corpus were extracted from the testing set and played to human subjects. All the true occurrences of the keywords were selected. The ones that the wordspotter successfully picked out (*true hits*) were excised with the boundary labels provided by the wordspotter. The true occurrences that the wordspotter missed (*misses*) were extracted using the boundary provided by the NIST word transcripts. To prevent the human listener from biasing the decision process based on the frequency of occurrence of false alarms vs. true hits, an equal number of false alarms and true occurrences were selected from the wordspotter output. The highest scoring false alarms were selected first and the selection stopped either when the number of false alarms equaled to the combination of true hits and misses, or when the pool of false alarms are exhausted. The boundary labels that the wordspotter generated for the false alarms were used as the basis to excise the speech segments from the evaluation conversations.

A complete wordspotting experiment, in which each subject listens through all the testing conversations and marks the occurrence of the keywords was ruled out because it would be too time consuming. The conducted experiments require less of the human subjects' time and also allow evaluating the benefit of context in discriminating between true hits and false alarms.

7.2 Experimental Setup

The human subject sits in an enclosed quiet room in which outside noises are attenuated. Speech segments have a sampling rate of 8000 Hz and are played through a Digital/Analog conversion board system connected to a Sun 4 workstation. The human subject wears a Sennheiser headphone and the playback volume is adjusted to suit each human subject's comfort level.

During the experiment, the human subject hears each segment played once and answers whether or not the pre-specified keyword occurs in the segment. The segments for each keyword are played successively in a randomized sequence. The whole test consists of 1,157 utterances and the number of true hits and false alarms played are shown in Table 7.1. Three sessions of listening experiments were performed with each human subject. In the first session, the time boundaries that are generated from the wordspotter are used to excise the speech segments. In the second session, the boundaries of the speech segments are extended by 0.1 second at both the beginning and the end. In the third session, the boundaries of the speech segments are extended by two seconds at both the beginning and the end. The first set of speech segments are what the wordspotter system hypothesized to be keyword occurrences. In preliminary experiment, it was found that the wordspotter frequently detected keywords based on strong vowel sounds and the surrounding consonants are frequently left out of the hypothesized occurrence. As a result, many words that were

Table 7.1: Distribution of True Hits, Misses, and False Alarms of 20 keywords used in human listening experiment

Keyword	# of True Hits	# of Miss	# of False Alarms
account	8	4	12
american express	10	2	0
balance	13	1	14
bank	22	3	25
card	81	1	82
cash	30	0	30
charge	36	1	37
check	32	3	35
credit	31	3	34
credit card	111	1	59
discover	7	3	6
dollar	24	1	25
hundred	10	7	17
interest	29	3	32
limit	9	2	11
money	21	4	25
month	37	2	39
percent	16	3	19
twenty	6	10	16
visa	22	4	26
Total	555	58	544

excised based on the output of the wordspotter are only portions of a word. The second set of experiments extended the boundaries long enough so that a word is completely represented. However, the human subjects still do not have enough context to allow the use of grammar and prosody. In the third set of speech segments, the boundaries are extended for two seconds in each direction. Thus, the speech segments contain partial or full sentences and the human subject can use grammar, prosody, and short term speaker adaptation to improve his detection of the keywords.

7.3 Experimental Results

Experiments were conducted in which the human listener's judgment was used to filter out putative hits as shown in Figure 7-1. If a putative hit from the wordspotter was classified as

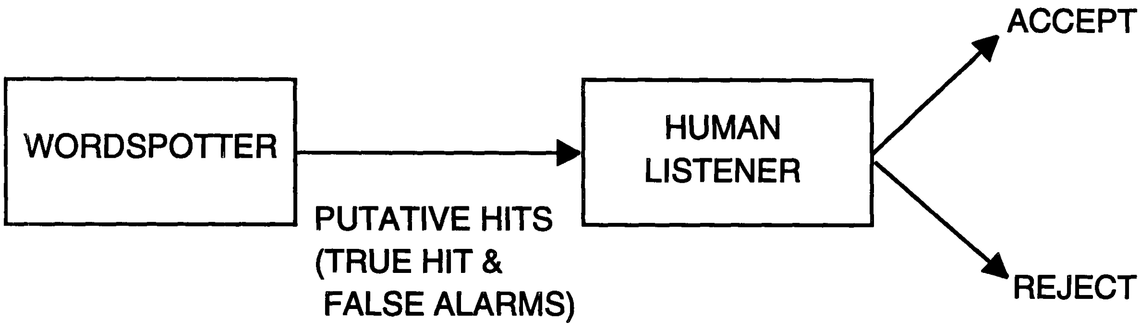


Figure 7-1: Using human listener to filter the putative hits generated by the wordspotter.

a false alarm by the human subject, then that putative hit was removed from the putative hit list. The result of this methodology is that a human subject's performance on the words missed by the wordspotter does not affect the final result since in a real system that uses human beings to perform second level filtering, the misses would not be made available to the human listener. The improvements in Figure of Merit through having a human filtering the putative hits are quite substantial for certain keywords, as shown in Table 7.2. The overall Figure of Merit increased by more than 10 percentage points. For the word *visa*, the increase was more than 20 percentage points. The human subjects suggested that it was frequently possible to reject a putative hit after hearing portions of the putative hit that could not have occurred in the true keyword. For example, the top 15 false alarms for the word *visa* were all successfully filtered out when the boundary was 0.1 seconds around both sides of each putative hit. Table 7.3 lists the transcription of the top 15 false alarms for *visa*. One immediately sees that most of the top scoring false alarms have the phoneme /iy/ or /i/ in them, such as the words *give, me, easy, early, steve,* and *peace*. Although these false alarms have vowels that sound similar to the strong vowel in the keyword *visa*, the consonants are different enough to allow the human subjects to recognize that they are false alarms. The wordspotter system apparently does not enforce as strong a constraint.

In general, the current wordspotter has the weakness of not placing enough emphasis on consonants to discriminate between the keywords and the false alarms. For example, *gas* was among the top scoring false alarms for the keyword *cash*. While both words consist of a sequence of /stop/ /vowel/ /fricative/, the beginning stop of the word *gas* was a good enough cue for the human subject to discriminate between the two words. Such examples are found for other keywords as well. For example, the word *charge* is frequently picked

Table 7.2: The per-keyword and overall FOM for the combined testing set for the baseline system and the system using Subject 2 to filter the top scoring putative hits and with misses not counted.

Keyword	FOM Before	FOM After	Change in FOM	High Scoring False Alarms
account	41.3%	41.4%	0.1%	*are counting, found*
american_express	83.3%	83.3%	0.0%	None
balance	62.1%	81.8%	19.7%	*account, down*
bank	59.8%	63.5%	13.7%	*back, things*
card	64.4%	80.9%	16.5%	*car, far, charge*
cash	74.7%	82.2%	7.5%	*passion, cats, gas*
charge	71.8%	89.7%	17.9%	*sure, cards*
check	69.4%	79.8%	10.4%	*protection, kept*
credit	71.1%	86.8%	15.7%	*crazy, describing*
credit_card	94.2%	94.2%	0.0%	*credit when, that card*
discover	70.0%	70.0%	0.0%	*especially, go*
dollar	74.5%	83.8%	9.3%	*without, follow*
hundred	51.0%	57.6%	6.6%	*can deduct, another*
interest	77.8%	86.0%	8.2%	*when it's, just*
limit	46.7%	48.3%	1.6%	*might, that one*
money	39.0%	53.4%	14.4%	*myself, month*
month	70.2%	84.4%	14.2%	*mom's, not*
percent	62.7%	82.2%	19.5%	*never send, sinclair*
twenty	25.2%	34.6%	9.4%	*once, of these*
visa	40.1%	66.0%	25.9%	*give, me*
Overall	68.9%	79.0%	10.1%	-

Table 7.3: The false alarm tokens that are successfully filtered by Subject 2 and the actual transcriptions.

Conv	Start	Duration	Score	Transcription
sw2710_b	39.84	0.31	48.6	give
sw2710_b	583.83	0.39	40.2	me
sw2710_b	42.51	0.33	35.47	easy
sw3409_a	216.58	0.35	34.41	early
sw2987_b	51.71	0.35	29.94	steve
sw2682_a	457.55	0.29	27.6	you something
sw2987_b	31.47	0.50	25.80	use the
sw2682_a	165.18	0.28	25.40	you see
sw2987_b	105.59	0.26	23.88	to be that
sw3751_a	29.69	0.36	23.43	you use a
sw1026_b	6.26	0.39	22.19	really
sw2987_b	158.77	0.38	21.78	got to use
sw2987_b	266.57	0.30	21.12	teeth
sw2710_b	413.94	0.27	17.78	peace of
sw2710_b	374.35	0.30	17.10	fifteenth of

up by the wordspotter as a false alarm of the word *card*. While the wordspotter picks out keywords primarily by the longer duration vowels in each keyword, human subjects were able to pick up discriminating cues which separate the keywords from the false alarms. The inability to pick up subtle differences between similar sounding phonemes is a common failing of modern speech recognition systems. For example, Duchnowski's results showed that the phoneme /g/ was most frequently confused with the phoneme /k/ [18].

The overall performance of the human subjects in filtering out false alarms is shown in Table 7.4 and plotted in Figure 7-2. When the exact boundary from the wordspotter was used, Subject 1 improved the FOM by 4.5 percentage points (68.9% to 73.4%). Subject 2 actually decreased the FOM by 0.9 percentage point (68.9% to 68.0%) because many word segments contained only portions of the keywords and these segments were rejected. When window size around the speech segments is extended by 0.1 second, Subject 1 improved the FOM by 8.6 percentage points (68.9% to 77.5%) while Subject 2 improved FOM by 10.1 percentage points (68.9% to 79.0%). In this session, the word in question occurs completely in the excised segments, but not enough surrounding words are included to allow the use of grammar and prosody. In session 3, the window size around each excised speech segment was

Table 7.4: FOM after using humans to filter putative hits.

	Window Size		
Listener	Exact Boundary	0.1 Sec. Window	2.0 Sec. Window
original	68.9%	68.9%	68.9%
Subject 1	73.4%	77.5%	82.8%
Subject 2	68.0%	79.0%	83.3%
perfect	83.3%	83.3%	83.3%

extended to 2 seconds in each end. With enough words in the speech segments to provide some context, Subject 1 increased FOM by 13.9 percentage points (68.9% to 82.8%). Notice that the performance of Subject 1 is very close to perfect performance on this test. The perfect performance is calculated by rejecting all false alarms in the listening tests while accepting all true hits. The FOM for perfect rejection is not 100.0% because there are still misses that were not detected. Also, there are lower scoring false alarms that were not listened to by the human subjects and thus could not have been rejected.

The gap in human performance between the second session and the third session is up to 5.3 percentage points. Interestingly, such a gap correlates with the difference in performance between a large vocabulary wordspotter running with grammar and a large vocabulary wordspotter running without grammar. Carlson has performed experiments in which she used a null grammar instead of the estimated bigram in a large vocabulary continuous speech wordspotter and the FOM dropped from 75.0% to 69.4% [6]. With a null grammar, all words are equally likely to follow any given word. With an estimated bigram, the probability of one word following another word is estimated from the Switchboard credit card transcripts provided by NIST. Since in reality words do frequently follow other related words, such as *card* following *credit*, using the bigram provides more constraint on the occurrence of words and improves wordspotting performance. Human beings rely on the same phenomenon to discriminate between similarly sounding words. For example, the /k a r/ sound in the phrase *"I have two cars in my garage"* and the phrase *"I have two cards in my wallet"* would be very confusable if the surrounding words are not used to determine whether *cars* or *cards* is being said.

In Table 7.4, human subjects' performance on the misses are not counted because the misses would not be presented if human subjects were used as a secondary filter to the

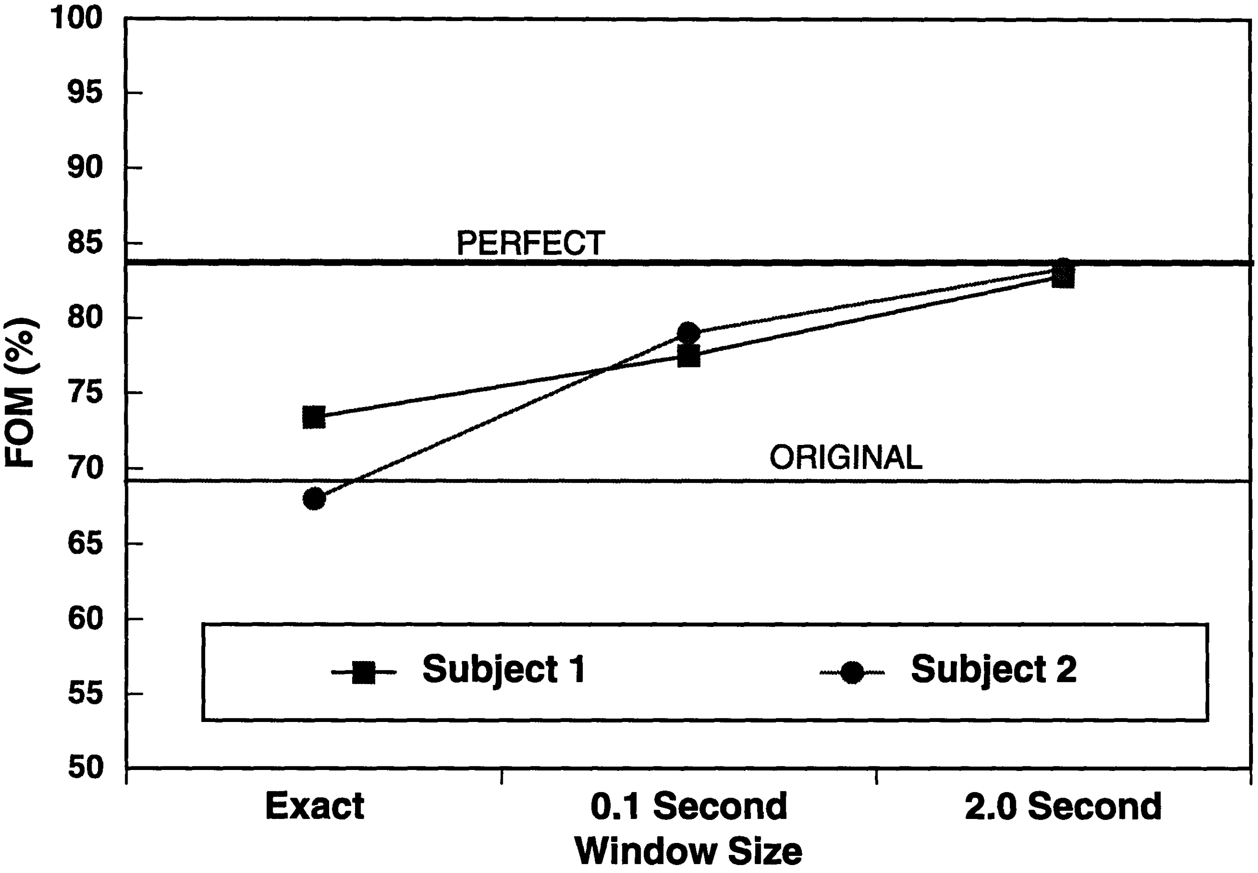


Figure 7-2: Results from using human subjects to filter the putative hits generated by the wordspotter.

Table 7.5: FOM after using humans to filter putative hits and counting human detection of misses.

	Window Size		
Listener	Exact Boundary	0.1 Sec. Window	2.0 Sec. Window
original	68.9%	68.9%	68.9%
Subject1	82.7%	86.9%	92.3%
Subject2	77.3%	87.4%	92.8%
perfect	92.8%	92.8%	92.8%

output of the wordspotter. However, it would be interesting to know how well human subjects can detect keyword occurrences that were missed by the wordspotter. Table 7.5 lists the FOM when the misses that human subjects detected correctly are counted and added to the putative hit list with the highest score possible. Figure 7-3 plots the same result graphically. As one can see, the FOM improved substantially when the misses are counted. Even with the exact boundary, Subject 1 improved the FOM from 73.4% to 82.7% when the misses detected by Subject 1 are counted. For session 3, the FOM from Subject 1 is 92.3%, very close to 92.5% from a perfect performance. The perfect performance is not 100.0% because there are still lower scoring false alarms that were not played to the human subjects and thus could not have been removed. After listening to the words that the wordspotter missed, no clear rules could be derived which define why some speakers are easier to comprehend. When the input conversation contains a new variability that the wordspotter had not been exposed to before, the wordspotter performs inadequately. For example, only one example for the keyphrase *american express* was missed. In this particular case, the speaking rate is very slow and the wordspotter was not able to cope with the large number of frames.

7.4 Previous Listening Experiments

While there have not been other human listening experiments for the wordspotting task, there have been experiments comparing the performance of human versus machines in other tasks. Recently ARPA organized a benchmark test on large vocabulary speech transcription tasks. In one set of experiments, sentences taken from Wall Street Journal articles are read and used as input for a transcription system. Results are presented by Pallett et al. on the

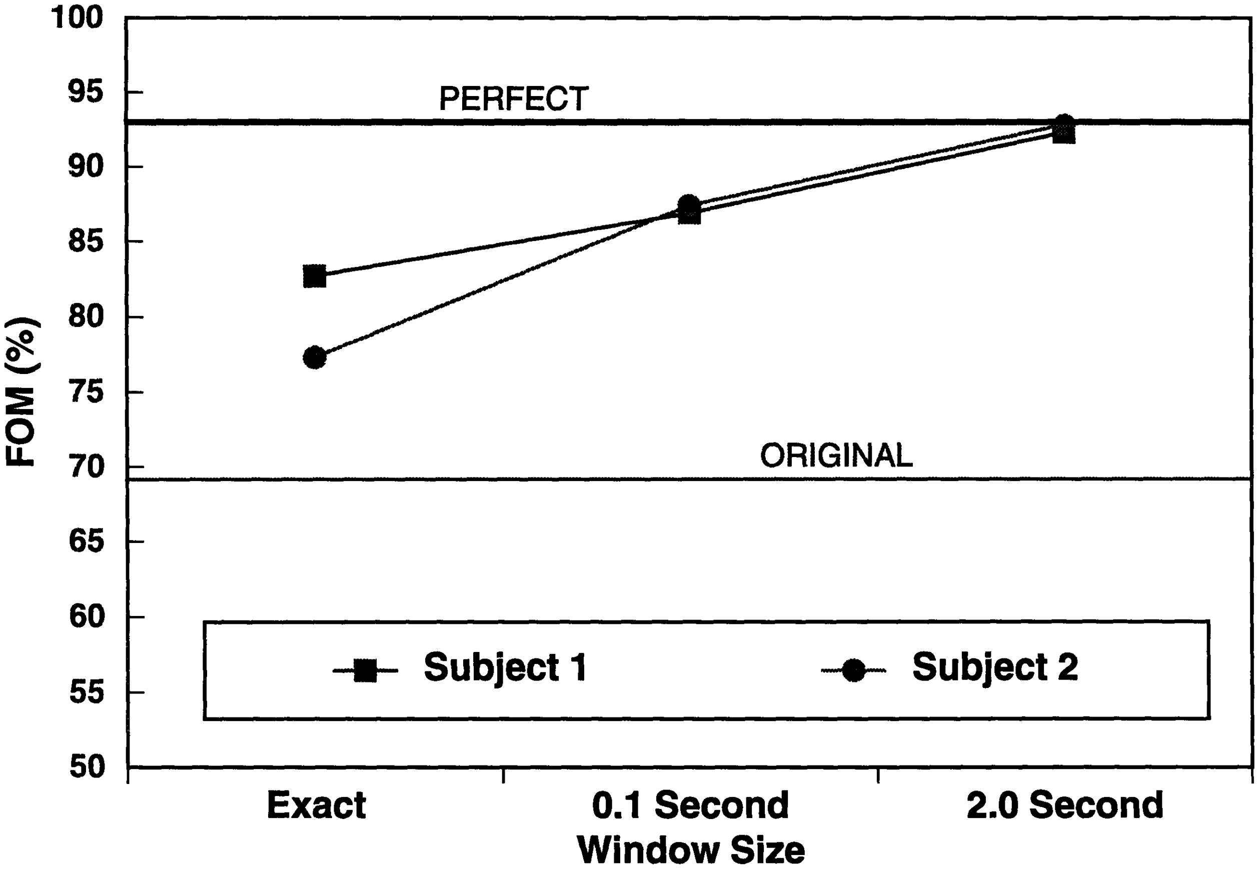


Figure 7-3: Results from using human subjects to filter the putative hits generated by the wordspotter and to detect the keyword occurrences missed by the wordspotter.

Table 7.6: Word error rates on continuous speech, human vs. machines.

	Noise Compensation			No Compensation			
	22 dB	16 dB	10 dB	22 dB	16 dB	10 dB	inf
System	W.E.	W.E.	W.E.	W.E.	W.E.	W.E.	W.E.
cu-htk4	9.4%	13.4%	19.8%	41.9%	59.4%	84.7%	7.2%
ibm2	8.4%	10.0%	12.8%	15.4%	42.2%	77.4 %	7.2%
sri3	8.4%	9.8%	12.2%	11.1%	18.4%	35.4 %	6.7%
human (open)	-	-	-	2.1%	2.1%	2.1%	2.0%
human (closed)	-	-	-	0.9%	1.0%	1.1%	0.9%

performance of different sites on a continuous speech recognition task [57]. Performances of human subjects on the same task are presented by Ebel et al [19]. A summary of machine performance versus human performance is shown in Table 7.6. In this experiment, different levels of signal to noise ratio (SNR) were created by adding a noise recorded in a car traveling on the highway to the original clean speech recording. The magnitude of the noise added is adjusted to create the appropriate SNR. The machine performance was evaluated under three conditions:

1. The system uses samples of noise for compensation.
2. The system is not adapted to the noise source at all.
3. The system used an input that was clean speech.

The human performance is evaluated under two conditions:

1. The human's transcription is directly compared with the true labels (open condition).
2. The human's transcription was processed by replacing words that are out of the vocabulary with similar sounding words in the vocabulary (closed condition).

It is clear that human performance is still much better than machine performance, especially when the input is noisy. With a SNR of 10 dB, the error rates of machines are ten times that of human beings under the closed condition. Even when the input consists of clean speech, the error rates of machines are still about 7 times worse than that of human beings.

7.4.1 Comparison Between the Wordspotting Tests and the Transcription Tests

Upon seeing the relatively low error rates in Table 7.6, one may wonder why the current wordspotting system is performing only with a FOM of about 70%. While FOM is not strictly comparable to the word error rate, the large gap does bring questions.

One reason for the discrepancy is the difficulty of the databases. The Switchboard credit card database is collected over the telephone while the Wall Street Journal is high quality speech recorded with a microphone. Also, the Switchboard credit card database consists of two way conversations, while the Wall Street Journal database consists of read speech, and therefore the problem of crosstalk does not occur in the Wall Street Journal database.

Another factor is the difference between spontaneous speech and read speech. Recently Daly has performed an analysis of a database of spontaneous commands to a city guide system [11]. Her results show that in spontaneous speech people speak with much more disfluencies. In the credit card database, people often interject and hesitate by saying words such as *uh-hum, sure, yeah.* Such disfluencies make the task more difficult for the speech recognizer.

Lastly, even when the disfluencies are taken into account, the speaking styles of unconstrained, regular conversation and read speech are very different. The researchers at BBN have performed experiments in which sentences from the Switchboard database, including the disfluencies, and Wall Street Journal articles are read by human subjects and recorded with a high quality microphone [20]. The new speech database was used as input to a speech recognizer. There was still a gap of about twenty percentage points (27.5% versus 8.8%) in word error rate between the Switchboard sentences and the Wall Street Journal sentences. There are proportionally more function words and short words in the Switchboard sentences. But even for words of similar length, the word error rate on the Switchboard sentences is still higher.

Given these differences between the Switchboard database and the Wall Street Journal database, it is not surprising that continuous speech recognition performance on the Switchboard database is much worse, with word error rates in the range of 55% [15]. Thus the FOM of around 70% is not unreasonable.

7.5 Chapter Summary

This chapter describes experiments in which two human subjects tried to detect occurrences of keywords in excised segments from the Switchboard database. Three types of segments, those containing true hits, those containing false alarms, and those containing misses were used in the experiment. Human subjects were able to discriminate between true hits and false alarms when the complete words were played out. After an analysis of the high scoring false alarms, it was found that the current wordspotting system detects keywords mostly by spotting for the occurrence of vowels that are part of the keyword. Consonants which discriminate between true hits and false alarms are not used by the wordspotter sufficiently. The wordspotter system is currently still about 20 percentage points below human performance and much future work remains to be done.

Chapter 8

Discussion

The algorithms developed in this thesis can be evaluated along two different dimensions:

1. How well do they improve a wordspotting system?

2. What lessons have been learned that can be applied elsewhere?

Section 8.1 describes the results from using the algorithms presented in this thesis to train wordspotters that were used to spot the NIST September 1992 official testing set. Because results from other sites are available for this database, some comparison can be made between the whole-word based wordspotter and those of other sites. The performance of the whole-word based system presented in this thesis also compares favorably to large vocabulary continuous speech recognition systems. Section 8.2 compares the characteristics of the whole-word wordspotting system described in this thesis to a large vocabulary continuous speech recognition (LVCSR) system developed at Lincoln Laboratory. The main results are that the whole-word system uses orders of magnitude less storage and computation resources and performs very closely to a LVCSR system. In Section 8.3, other possible extensions and applications of the results of this thesis are suggested.

8.1 Results on the Official Test Set

At the conclusion of research in this thesis, the methodology described throughout this thesis was used to train a wordspotter. All 35 conversations for each gender in the NIST database were used to train a wordspotter for the corresponding gender. The wordspotters were trained with five sets of transformed database plus the original database. Training included maximum likelihood estimation and Figure of Merit Training with the artificially enlarged training set. The resulting wordspotters were used to spot the conversations in the NIST official testing set of September 1992. The results from other sites are provided by NIST and show the performance of each site in Fall, 1992. The performance of the system developed in this thesis was obtained in Spring, 1995.

Figure 8-1 plots the FOM of systems developed by the various sites. The Lincoln whole-word wordspotter with FOM training and voice transformation provided the highest FOM of all whole-word wordspotters. On this testing set, the wordspotter described in this thesis has a FOM of 64.2%. The next highest performing whole-word wordspotter is from BBN (obtained in Fall, 1992) with a FOM that is more than 10 percentage points lower (53.54%). The CMU system, based on discriminative training of a TDNN system, achieved a FOM of 51.04% (obtained in Fall, 1992). The best performing wordspotter among all the entrants is the BBN-LVCSR system with a FOM of 69.2% (obtained in Fall, 1992) [67]. In the summer of 1993, BBN reported improvements in their LVCSR system and a new FOM of 75.4% on the NIST official testing set [34]. LVCSR systems, as shown in the Section 8.2, are orders of magnitude more complex than a whole-word system. A whole-word system can easily be implemented on single DSP chips or personal computers and work with relatively rapid training. Large vocabulary recognizers require fast processors with rapid access to large memories and long complex training procedures. The other LVCSR system was from SRI International with a FOM of 59.88% (obtained in Fall, 1992). This system is based on the SRI DECIPHER continuous speech recognition system and utilizes a lexicon of up to 6900 words. However, it generates putative hits by using the Viterbi decoding approach described in Section 4.5.1. As explained in Section 4.5.2, the Viterbi decoding method is limited to generate only one putative hit at any given time and does not perform as well as the peak-picking method. The BBN-LVCSR system also utilizes the peak-picking method and this difference may explain the gap between the performance of the SRI system versus the BBN system. The Figure of Merit summarizes the information contained in a Receiver Operating Characteristic (ROC) curve, which is a plot of true detection rate versus the false alarm rate. Figure 8-2 shows the ROC curves from the BBN-LVCSR, the system developed in this thesis (Lincoln-WW), the whole-word based system from BBN (BBN-WW), and the whole-word based system from CMU (CMU-WW). The system developed in this thesis clearly surpasses the performance of the other whole-word based systems that were tested

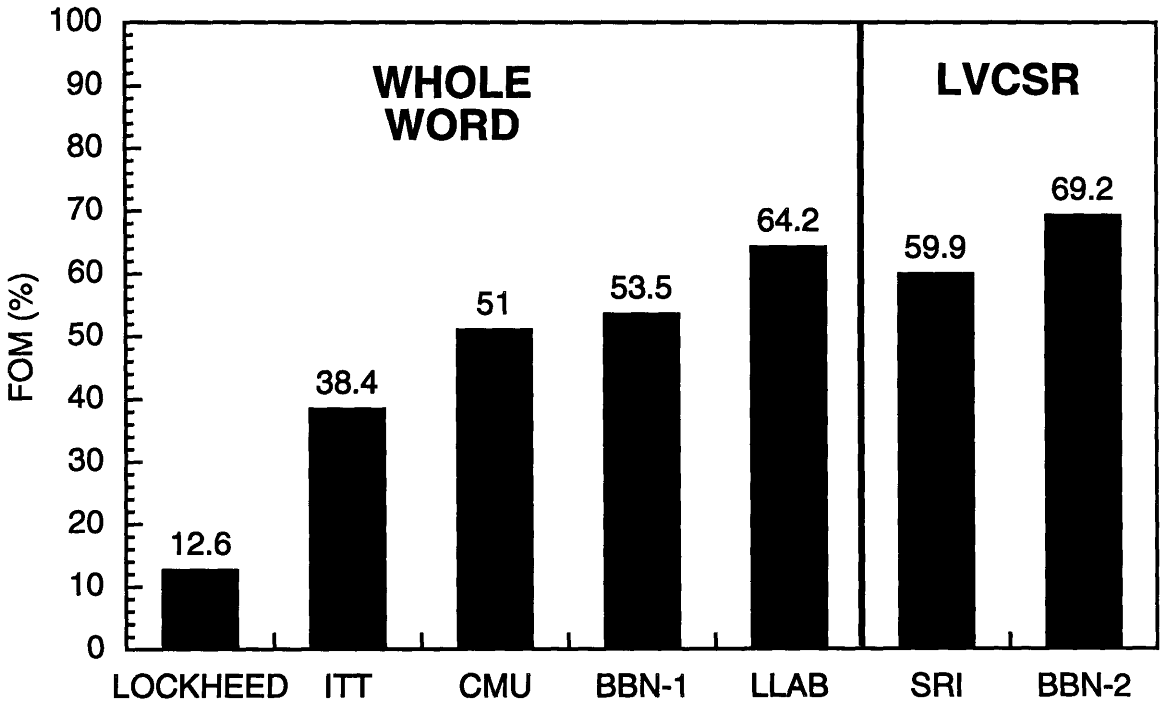


Figure 8-1: Figure of Merit of different sites on the September 1992 Official Testing Set.

in Fall, 1992. The group at BBN has not worked on its whole-word based system since then [51].

The other low ranking systems are from Lockheed and ITT. The Lockheed system was developed in 1987 by Daniel Griffin and utilizes filter bank energies as input features and dynamic time warping to match the input pattern to keyword templates [49]. The system was submitted mainly to evaluate how much wordspotting performance has improved since 1987. The ITT system is also based on a dynamic time warping system and the performance is clearly not as good as HMM based systems.

8.2 Comparison to Large Vocabulary Continuous Speech Recognition Based Systems

Results from the whole-word wordspotter described in this thesis and results obtained with a large vocabulary continuous speech recognition (LVCSR) system developed at Lincoln Laboratory [7] have allowed a comparison of the performance of the whole-word based wordspotter to the LVCSR wordspotter in terms of Figure of Merit, memory requirements, and computation requirements.

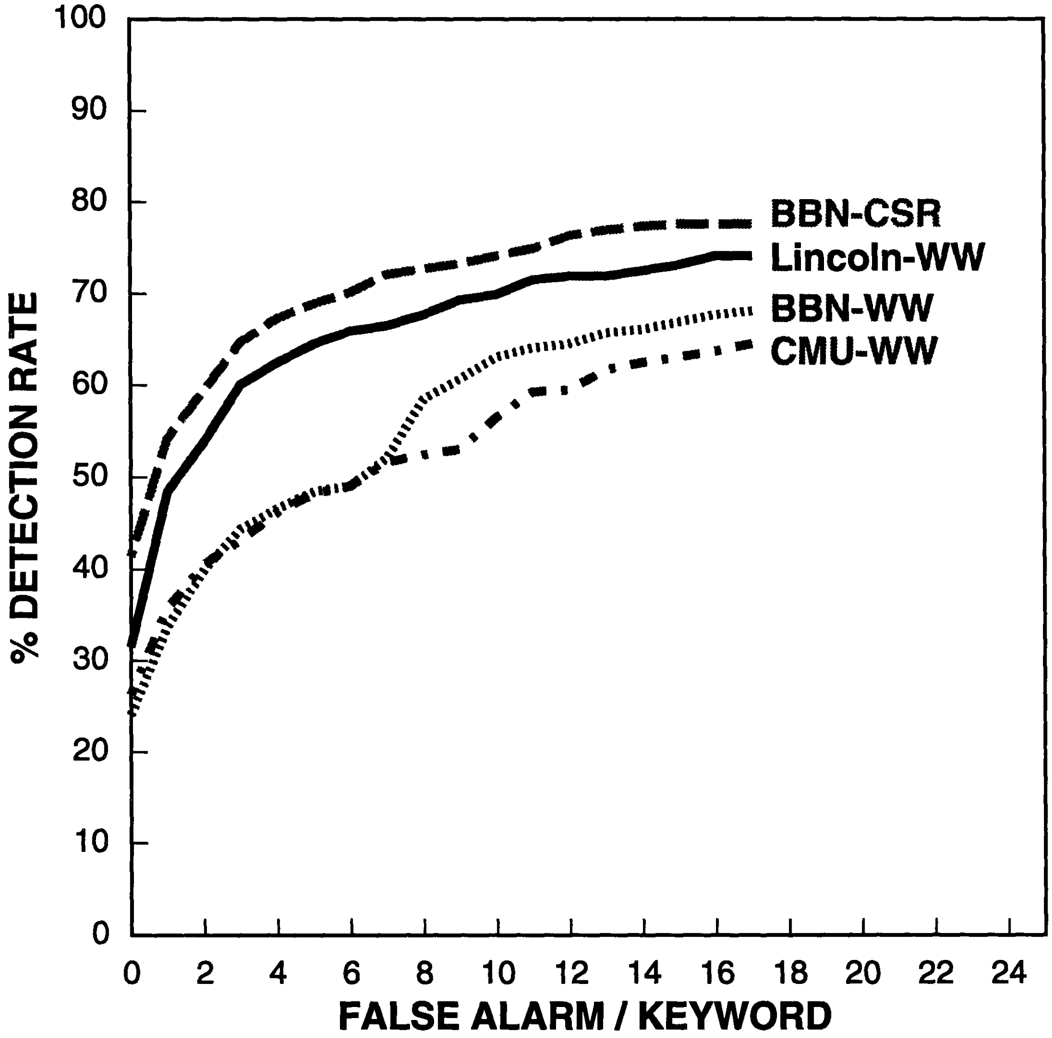


Figure 8-2: Receiver operating characteristic (ROC) curves of the BBN-CSR, Lincoln Whole-Word, BBN Whole-Word, and CMU Whole-Word wordspotters.

Table 8.1: The number of trained parameters used by the whole-word wordspotter and LVCSR wordspotter

Type of Parameters	Whole-Word Wordspotter	LVCSR Wordspotter
States	234	6,000
Mixture Weights	920	1,536,000
Mixtures	23,000	6,400
State Weights	234	Not Applicable
Bigram Probability	484	4,000,000
Total	24,872	5,548,400

The LVCSR wordspotter requires fast processors with rapid access to large memories and long complex training procedures. For example, comparing the whole-word wordspotter to a LVCSR wordspotter also developed at Lincoln Laboratory, the whole-word wordspotter uses two orders of magnitudes fewer trained parameters than the LVCSR system. Table 8.1 shows how these parameters are apportioned for a tied-mixture LVCSR systems with 2,000 triphones, 128 tied mixtures, and 2,000 words. The input features are 12 cepstra and 13 delta cepstra. Thus for each mixture, a total of 50 floating point numbers are stored. Similarly, the amount of computation required for the two wordspotters is drastically different. According to previous experience with the whole-word wordspotter and the LVCSR system, the amount of computation time required for the LVCSR system is 10 to 20 times more than the time required for the whole-word system when extensive pruning is used with the LVCSR system.

The amount of training data required is also different for the two systems. The whole-word wordspotter can not be used on new keywords without retraining. However, for a new set of keywords, as long as a few samples of the keyword from a variety of speakers are available, a whole-word wordspotter can be quickly trained. The LVCSR system can be used on novel keywords without retraining. However, without having specific samples of the new keywords, the FOM of the general system can be as much as 20% lower than that of the system trained using samples of the new keyword. For example, Table 8.2 presents results from Carlson and Seward in which different LVCSR systems were trained using four different databases [7]:

1. Sentences recorded at Lincoln Laboratory which contain the 20 keywords in the credit card task,

2. Sentences recorded at Lincoln Laboratory which do not contain the 20 keywords in the credit card task,

3. Sentences from the NTIMIT database [32],

4. Sentences from the Switchboard credit card database.

It is clear that the training database affects the performance of the wordspotter tremendously. The inclusion of the appropriate keywords in the database resulted in a difference of 17.4% in the FOM of wordspotters trained with Lincoln databases. Also, the phonetically rich NTIMIT database provided much worse performance than the Switchboard credit card database. Thus, LVCSR system's advantage in being able to spot any given word must be put in context: it is clear that having a task dependent database is still the best way to obtain superior results.

Figure 8-3 compares the performance of the whole-word based system versus a LVCSR system in terms of model complexity, execution speed, and overall accuracy. The model complexity and the execution speed were derived from measuring the LVCSR system developed by Beth Carlson at Lincoln Laboratory. The overall accuracy shows the best performance that has been obtained on the test set of the research split. The LVCSR system performance is that of the LVCSR system developed at Lincoln Laboratory [7] and the whole-word system performance is that of the system presented in this thesis. The whole-word based system uses orders of magnitude less storage space and executes an order of magnitude faster than the LVCSR system while sacrificing four percentage points in FOM. For applications requiring low power and memory consumption such as personal digital assistants or cellular phones, the whole-word based system presented in this thesis can offer high performance and low memory and power consumption. Recently, new searching techniques have been developed that can significantly reduce the execution time of a LVCSR system [85], however, the memory requirements of such systems are still much larger than that of a whole-word based system.

8.3 Lessons Learned

In this thesis, Chapter 4 describes a high performing wordspotting system that is competitive with other existing whole-word based wordspotters. Two algorithms are introduced in

Table 8.2: FOM of a LVCSR system trained on different sets of data (From Carlson94).

Training Corpus	% Combined FOM
Lincoln Credit Card	46.2%
Lincoln non-Credit Card	28.8%
NTIMIT	25.1%
Switchboard Credit Card	64.0

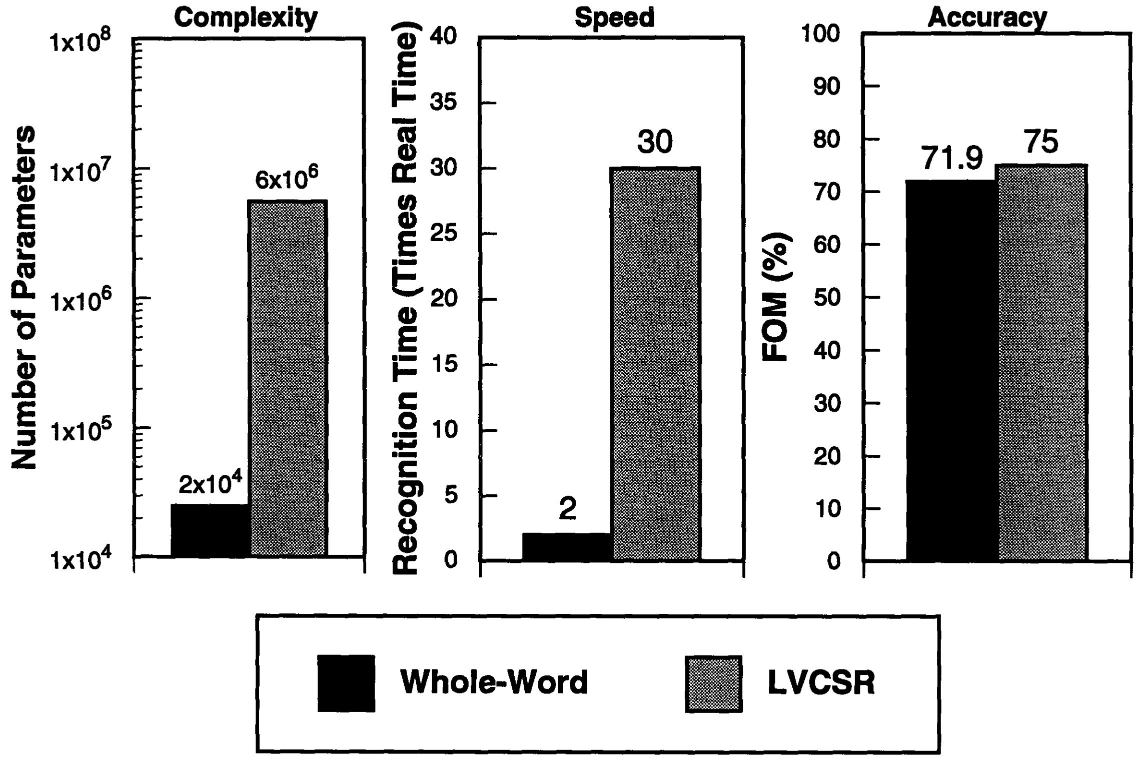


Figure 8-3: Comparison between a whole-word based system and a LVCSR system on complexity, speed, and accuracy.

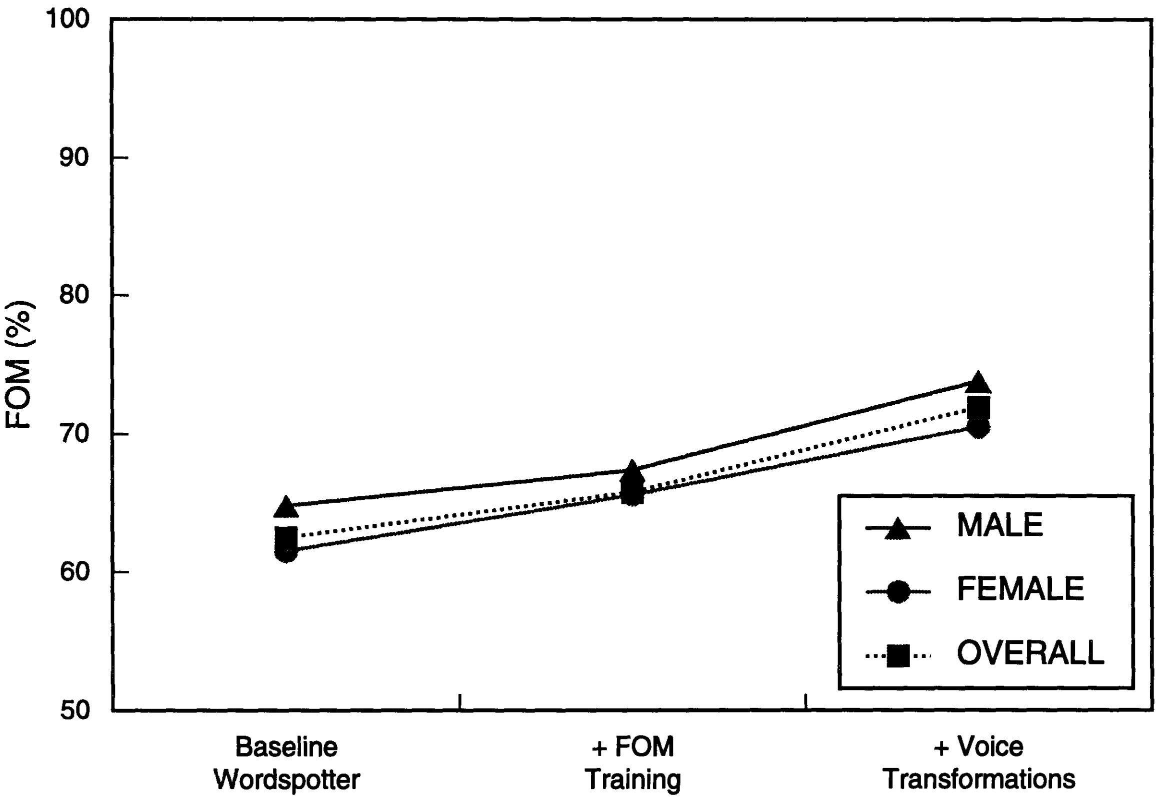


Figure 8-4: Improvements in FOM after Figure of Merit training was applied on the baseline wordspotter, and after voice transformation was jointly applied with FOM training.

Chapters 5 and 6 to improve the performance of the baseline wordspotter:

- Figure of Merit Training
- Voice Transformation

It was first shown that a discriminative training technique that directly maximizes the evaluation criterion of interest can outperform a maximum likelihood approach. Such improvements are possible because the amount of data is insufficient to represent the variability of the data. Also, the correct probability models in terms of the number of Gaussian mixtures and the number of states required to represent the temporal variability of speech are unknown. By using the Figure of Merit training approach, the model parameters are adjusted to maximize the Figure of Merit. The FOM on the combined testing set increased by 3.3 percentage points after five iterations of FOM training.

The discriminative approach suffers from the possibility of overtraining on the training set. A complementary approach to Figure of Merit training called *Voice Transformation* was developed. Using voice transformation to increase the variability of the database improved the robustness of the wordspotting system, resulting in an overall increase of 9.6 percentage points in FOM when Figure of Merit training and voice transformation were combined. Figure 8-4 summarizes the improvements in FOM on the testing set when FOM training was applied on the baseline wordspotter and when voice transformations and Figure of Merit training were jointly applied.

Studies were also conducted on unsupervised speaker adaptation approaches to enhance the accuracy of the wordspotter. While experiments demonstrated that changing the spectral transformation ratio affects the accuracy of the wordspotter, deriving the correct transformation ratio for a given speaker requires extra computation during wordspotting and long speech segments for reliable estimation. Thus, speaker adaptation approaches are not suitable for tasks which require rapid recognition without the luxury of obtaining additional data to adapt to the speaker.

To investigate the possibility of training a wordspotter system without any real speech, speech obtained from an artificial synthesizer was used to train a wordspotting system. Wordspotting results are much inferior to wordspotters trained with natural speech. Informal auditioning of the machine generated speech also revealed many artifacts. Unless speech synthesis technology is drastically improved, in the foreseeable future better speech recognition systems will still require the collection of human speech. However, the voice transformation algorithm can be used to introduce variabilities from known sources and make the collected training database more useful.

Finally, two subjects participated in wordspotting experiments to determine the strengths and the weaknesses of the wordspotter. It was found that the human subjects could detect almost all of the false alarms generated by the wordspotter when the complete words were provided. When the complete sentences were played, the human subjects achieved almost perfect scores. The current wordspotting system is capable of detecting the occurrence of keywords by concentrating on the occurrence of vowel sounds. However, the current wordspotting system has difficulty using emphasis on the consonants in each keyword to discriminate between keywords and false alarms. Also, it was found that the current wordspotting system still does not have as good a detection rate as the human subjects due to worse ability in handling different variations of keywords such as extremely long durations. Overall, the performance of the wordspotter is about 20 percentage points below that of human subjects. Clearly, much work remains in improving wordspotting performance.

8.4 Future Work

Possible directions for extending the work in this thesis can be considered in two dimensions. For speech recognition tasks, promising results have been obtained by using artificial transformation techniques to increase the variability of the data. In this thesis the variability was added in the formant frequency domain. Other possible sources of variations can also be considered, for example, the rate of speech, the manner of speaking, pitch, and dialect can possibly be transformed. Recently, there has been work in creating a *voice font* that characterizes a person's voice [30]. Such tools can be used to generate more controlled variabilities in a speech database to increase the robustness of a wordspotting system.

However, the speech recognizer performance is still far from that of a human being. The experimental results from Chapter 7 have shown that human beings can discriminate between consonant pairs such as /g/ and /k/ much better than the wordspotting system can. Better features may need to be developed that more clearly capture the differences between consonants that do not last for a long time in speech waveforms.

Along the other dimension, this thesis demonstrates the effectiveness of combining discriminative training techniques that are directly targeted toward optimizing the evaluation criterion. The possibility of overtraining can be offset through adding variability according to *a priori* knowledge about the source of variability. Recently similar approaches have been used in other domains such as autonomous navigation [60] and financial prediction [1]. It would be interesting to apply the techniques presented in this thesis to another domain such as heart beat monitoring or defect detection on the manufacturing line, where the amount of variability is large and the amount of training data is thus relatively limited.

Bibliography

[1] Yaser S. Abu-Mostafa. A method for learning from hints. In Steven José Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 73–80, San Mateo, California, 1993. Morgan Kaufmann.

[2] James K. Baker. *Stochastic Modeling as a Means of Automatic Speech Recognition.* PhD thesis, Carnegie Mellon University, 1975.

[3] Mats Blomberg. Synthetic phoneme prototypes in a connected-word speech recognition system. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, pages 687–690, Glasgow, Scotland, May 1989.

[4] Mats Blomberg, Rolf Carlson, Kjell Elenius, Bjorn Granstrom, and Sheri Hunnicutt. Speech recognition based on a text-to-speech synthesis system. In *Proceedings European Conference on Speech Technology*, pages 687–690, Edinburgh, 1987.

[5] Hervé Bourlard and Nelson Morgan. *Connectionist Speech Recognition: A Hybrid Approach.* Kluwer Academic Publishers, Boston, 1994.

[6] Beth Carlson. Personal Communication, 1995.

[7] Beth Carlson and D. C. Seward IV. Diagnostic evaluation and analysis of insufficient and task-independent training data on speech recognition. In *Proceedings Speech Research Symposium XIV*, Johns Hopkins University, 1994.

[8] Eric I. Chang and Richard P. Lippmann. A boundary hunting radial basis function classifier which allocates centers constructively. In Stephen José Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 139–146, San Mateo, California, 1993. Morgan Kaufmann.

[9] Eric I. Chang and Richard P. Lippmann. Figure of merit training for detection and spotting. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 1019–1026, San Mateo, California, 1994. Morgan Kaufmann.

[10] Benjamin Chigier. Rejection and keyword spotting algorithms for a directory assistance city name recognition application. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 93–96, San Francisco, USA, March 1992.

[11] Nancy Ann Daly. *Acoustic-Phonetic and Linguistic Analyses of Spontaneous Speech: Implications for Speech Understanding.* PhD thesis, Massachusetts Institute of Technology, 1994.

[12] S. B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

[13] Celinda de la Torre and Alejandro Acero. Discriminative training of garbage model for non-vocabulary utterance rejection. In *Proceedings International Conference on Spoken Language Processing*, pages 475–478, Yokohama, Japan, September 1994.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[15] Vassilios Digilakis and Leo Neumeyer. Speaker adaptation and phone recognition experiments on WSJ and Switchboard. In *Proceedings DoD Workshop: Frontiers in Speech Processing*, Rutgers University, 1994.

[16] Alvin W. Drake. *Fundamentals of Applied Probability Theory.* MacGraw-Hill Book Company, Boston, 1967.

[17] Harris Drucker, Robert Schapire, and Patrice Simard. Improving performance in neural networks using a boosting algorithm. In Steven José Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 42–49, San Mateo, California, 1993. Morgan Kaufmann.

[18] Paul Duchnowski. *A New Structure for Automatic Speech Recognition.* PhD thesis, Massachusetts Institute of Technology, 1993.

[19] W. J. Ebel and J. Picone. Human speech recognition performance on the 1994 CSR S10 corpus. In *Proceedings ARPA Spoken Language Systems Technology Workshop*, Austin, Texas, January 1995.

[20] Ellen Eide, Herbert Gish, J. Robin Rohlicek, and Angela Mielke. Error analysis on Switchboard. In *Proceedings Speech Research Symposium XIV*, Johns Hopkins University, 1994.

[21] Eloquence. Eloquence user's manual. Eloquence Incorporated, 1994.

[22] Ming-Whei Feng, Francis Kubala, Richard Schwartz, and John Makhoul. Improved speaker adaptation using text dependent spectral mappings. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, pages 131–134, New York City, USA, April 1988.

[23] Ming-Whei Feng, Francis Kubala, Richard Schwartz, and John Makhoul. Iterative normalization for speaker-adaptive training in continuous speech recognition. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, pages 612–615, Glasgow, Scotland, May 1989.

[24] Herb Gish, Robin Rohlicek, Kenney Ng, Philppe Jeanrenaud, and Manhung Siu. A comparison of two HMM word spotting systems. Presentation Slides for the DARPA ANNT Review, September 1992.

[25] Herbert Gish and Kenney Ng. A segmental speech model with applications to word spotting. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 447–450, Minneapolis, USA, April 1993.

[26] John J. Godfrey, Edward C. Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume I, pages 517–520, San Francisco, March 1992.

[27] Hynek Hermansky, Nelson Morgan, Aruna Bayya, and Phil Kohn. RASTA-PLP speech analysis technique. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume I, pages 121–124, San Francisco, March 1992.

[28] Edward Hofstetter and Richard Rose. Techniques for task independent word spotting in continuous speech messages. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 101–104, San Francisco, USA, March 1992.

[29] Xuedong Huang and Kai-Fu Lee. On speaker-independent, speaker-dependent, and speaker-adaptive speech recognition. *IEEE Transactions on Speech and Audio Processing*, 1(2):150–157, April 1993.

[30] Voxware Incorporated. Toolvox: The next generation of voice utilities. Product Brochure.

[31] Naoto Iwahashi and Yoshinori Sagisaka. Speech spectrum transformation by speaker interpolation. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume I, pages 461–464, Adelaide, Australia, April 1994.

[32] Charles Jankowski Jr., A. Kalyanswamy, S. Basson, and J. Spitz. NTIMIT: A phonetically balanced, continuous speech, telephone bandwidth speech database. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume I, pages 109–112, Albuquerque, New Mexico, April 1990.

[33] Charles R. Jankowski Jr., Hoang-Doan H. Vo, and Richard P. Lippmann. A comparison of signal processing front ends for automatic word recognition. *IEEE Transactions on Speech and Audio Processing*, July 1995.

[34] Phillipe Jeanrenaud, Kenny Ng, Robin Rohlicek, and Herbert Gish. Phonetic training and language modelling for word spotting. In *Proceedings Speech Research Symposium XIII*, pages 233–235, Baltimore, Maryland, 1993.

[35] Lars Knohl and Ansgar Rinscheid. Speaker normalization and adaptation based on feature-map projection. In *Proceedings Eurospeech*, pages 367–370, Berlin, Germany, July 1993.

[36] J. Kupin. Wire: a wire-line simulator, manual page, April 1993.

[37] Kai-Fu Lee. *Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System.* PhD thesis, Carnegie Mellon University, 1988.

[38] Kai-Fu Lee and Sanjoy Mahajan. Corrective and reinforcement learning for speaker-independent continuous speech recognition. *Computer Speech and Language*, 4:231–245, 1990.

[39] Yuchun Lee. *Classifiers: Adaptive Modules in Pattern Recognition Systems.* Master's thesis, Massachusetts Institute of Technology, 1989.

[40] Richard Lippmann and Elliot Singer. Hybrid hmm/neural-network approaches to wordspotting. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume I, pages 565–568, Minneapolis, USA, April 1993.

[41] Richard P. Lippmann. Pattern classification using neural networks. *IEEE Communications Magazine*, 27(11):47–64, 1989.

[42] Richard P. Lippmann. Review of neural networks for speech recognition. *Neural Computation*, 1:1–38, 1989.

[43] Richard P. Lippmann, Eric I. Chang, and Charles Jankowski Jr. Wordspotter training using figure-of-merit back propagation. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume I, pages 389–392, Adelaide, Australia, April 1994.

[44] Richard P. Lippmann, Linda C. Kukolich, and Elliot Singer. Lnknet: Neural network, machine learning, and statistical software for pattern classification. *The Lincoln Laboratory Journal*, 6(2):249–268, 1993.

[45] Jeffrey N. Marcus. A novel algorithm for HMM word spotting, performance evaluation and error analysis. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 89–92, San Francisco, USA, March 1992.

[46] Helen Mei-Ling Meng. *The Use of Distinctive Features for Automatic Speech Recognition.* Master's thesis, Massachusetts Institute of Technology, 1991.

[47] Hideyuki Mizuno and Masanobu Abe. Voice conversion based on piecewise linear conversion rules of formant frequency and spectrum tilt. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume I, pages 469–472, Adelaide, Australia, April 1994.

[48] Pedro J. Moreno and Richard M. Stern. Sources of degradation of speech recognition in the telephone network. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume I, pages 109–112, Adelaide, Australia, April 1994.

[49] David P. Morgan and Christopher L. Scofield. *Neural Networks and Speech Processing.* Kluwer Academic Publishers, Boston, 1991.

[50] Kenney Ng. *A Comparative Study of the Practical Characteristics of Neural Network and Conventional Classifiers.* Master's thesis, Massachusetts Institute of Technology, 1990.

[51] Kenney Ng. Personal Communication, 1995.

[52] Kenney Ng, Herbert Gish, and J. Robin Rohlicek. Robust mapping of noisy speech parameters for HMM word spotting. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 109–112, San Francisco, USA, March 1992.

[53] National Institute of Standard and Technology. The road rally word-spotting corpora cd-rom, readme.doc, August 1991.

[54] National Institute of Standards and Technology. Switchboard Corpus, Credit Card Conversations, Wordspotting Training Set, Speech Disc 8-1.2, 1992.

[55] Yoshio Ono, Hisashi Wakita, and Yunxin Zhao. Speaker normalization using constrained spectra shifts in auditory filter domain. In *Proceedings Eurospeech*, pages 355–358, Berlin, Germany, July 1993.

[56] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1989.

[57] David S. Pallett, Jonathan G. Fiscus, William M. Fisher, John S. Garofolo, Bruce A. Lund, Alvin Martin, and Mark A. Przybocki. 1994 benchmark tests for the arpa spoken language program. In *Proceedings ARPA Spoken Language Systems Technology Workshop*, Austin, Texas, January 1995.

[58] Douglas Baker Paul. Speech recognition using hidden markov models. *Lincoln Laboratory Journal*, 3(1), 1990.

[59] G. E. Peterson and H. L. Barney. Control methods used in a study of vowels. *Journal of Acoustical Society of America*, 24:175–84, 1952.

[60] Dean A. Pomerleau. Efficient training of artificial neural netowrks for autonomous navigation. *Neural Computation*, 3(1), 1991.

[61] Thomas F. Quatieri and Robert J. McAulay. Shape invariant time-scale and pitch modification of speech. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 25(2):183–192, February 1992.

[62] Lawerence R. Rabiner and B. H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3:4–16, 1986.

[63] Lawerence R. Rabiner and Ronald W. Schafer. *Digital Processing of Speech Signals.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978.

[64] Steve Renals, Nelson Morgan, Michael Cohen, and Horacio Franco. Connectionist probability estimation in the DECIPHER speech recognition system. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume I, pages 601–604, San Francisco, March 1992.

[65] Douglas A. Reynolds. Xtalk, a silence and cross-talk detector. Lincoln Laboratory Internal Memo, 1993.

[66] Douglas A. Reynolds, Marc A. Zissman, Thomas F. Quatieri, Gerald C. O'Leary, and Beth A. Carlson. The effect of telephone transmission degradations on speaker recognition performance. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume 1, pages 329–332, Detroit, Michigan, May 1995.

[67] J. R. Rohlicek, P. Jeanrenaud, Kenney Ng, Herbert Gish, B. Musicus, and M. Siu. Phonetic training and language modeling for word spotting. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 459–462, Minneapolis, USA, April 1993.

[68] Richard Rose. Discriminant word spotting techniques for rejecting non-vocabulary utterances in unconstrained speech. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 105–108, San Francisco, USA, March 1992.

[69] Richard Schwartz, Yen-Lu Chow, and Francis Kubala. Rapid speaker adaptation using a probabilistic spectral mapping. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, pages 633–636, Dallas, April 1987.

[70] B. G. Secrest and G. R. Doddington. An integrated pitch tracking algorithm for speech systems. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1352–1355, 1983.

[71] Stephanie Seneff. A computational model for the peripheral auditory system: Application to speech recognition research. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1983–1986, Tokyo, Japan, 1986.

[72] Kenneth Stevens. Personal Communication, 1995.

[73] Gilbert Strang. *Linear Algebra and Its Applications.* Harcourt Brace Jovanovich, Publishers, San Diego, 1988.

[74] Rafid A. Sukkar and Jay G. Wilpon. A two pass classifier for utterance rejection in keyword spotting. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 451–454, Minneapolis, USA, April 1993.

[75] Luis Villarrubia and Alejandro Acero. Rejection techniques for digit recognition in telecommunication applications. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 455–458, Minneapolis, USA, April 1993.

[76] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-37(3):328–339, March 1989.

[77] Hisashi Wakita. Normalization of vowels by vocal-tract length and its application to vowel identification. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-25(2):183–192, April 1977.

[78] Lynn D. Wilcox and Marcia A. Bush. Training and search algorithms for an interactive wordspotting system. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 97–100, San Francisco, USA, March 1992.

[79] Jay G. Wilpon, Lawerence R. Rabiner, Chin-Hui Lee, and E. R. Goldman. Automatic recognition of keywords in unconstrained speech using hidden markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(11):1870–1878, November 1990.

[80] Steven J. Young. *HTK: Hidden Markov Model Toolkit.* Cambridge University Engineering Department, Cambridge, 1.4 edition, 1992.

[81] Torsten Zeppenfeld, Rick Houghton, and Alex Waibel. Improving the MS-TDNN for word spotting. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 475–478, Minneapolis, USA, April 1993.

[82] Torsten Zeppenfeld, Alex Waibel, and Rick Houghton. Word spotting on Switchboard. Presentation Slides for the DARPA ANNT Review, September 1992.

[83] Torsten Zeppenfeld and Alex H. Waibel. A hybrid neural network, dynamic programming word spotter. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 77–80, San Francisco, USA, March 1992.

[84] Yunxin Zhao. A new speaker adaptation technique using very short calibration speech. In *Proceedings IEEE International Conference on Acoustics Speech and Signal Processing*, volume II, pages 562–565, Minneapolis, USA, April 1993.

[85] Victor W. Zue. Personal Communication, 1995.