Speaker Recognition in the Text-Independent Domain Using Keyword Hidden Markov Models

Kofi A. Boakye
kaboakye@icsi.berkeley.edu

May 11, 2005

Contents

1 Introduction

In its most general sense, the goal of speaker recognition is to identify a person by his or her voice. The ability to accurately and effectively do this has become of increased importance in this modern age. The use of voice-driven and voice-related applications is on the rise. In addition, concerns regarding security of information (both personal and intelligence) have never been greater.

The difficulty of the speaker recognition task is often understated. Popular conception often reduces the problem to finding a "voiceprint" for an individual. Some even go as far as to consider speaker recognition a solved problem. Sadly, this is not the case. Many challenges still exist in the area of speaker recognition, and these challenges are receiving attention in the research community. A variety of solutions have been proposed to address these challenges and this research project looks to contribute by proposing and investigating one such solution.

1.1 Text-Dependent versus Text-Independent Domains

Because speaker recognition is used in a variety of applications, there are different domains of interest for the technology. One major division is between the text-dependent and text-independent domains. In text-dependent (sometimes referred to as "text-constrained") speaker recognition, the lexical content of the speaker's utterance is presumed to be precisely known beforehand. Examples of this include entry-control and user authentication systems using a fixed or a prompted phrase. On the other hand, for text-independent domains, the speech of a speaker is largely unconstrained—and often cannot feasibly be constrained—and the lexical content of utterances is highly variable. Some example applications are speaker indexing of audio archives, background verification during commercial interactions, and forensic and security applications involving found speech. Because speaker cooperation is not necessary, systems designed for to the text-independent domain are often considered more flexible.

It has been widely observed that a gap in performance exists between systems in the text-dependent and text-independent domains. More specifically, for a given speaker, text-dependent systems achieve much higher accuracy than their text-independent counterparts. A primary reason for this is that text-dependent systems can explicitly model phonetic content, so the remaining sources of acoustic variability are more likely due to speaker differences. As a result, more detailed modeling of the speaker is possible. In

addressing this performance gap, an interesting question arises: Is it possible to capitalize on the advantages of text-dependent systems while allowing for the flexibility associated with systems used in the text-independent domain?

1.2 Bridging the Gap: The Use of Keywords

One possible solution is modeling select keywords expected to appear in the speech stream. These keywords can then be identified, extracted, and used to perform speaker recognition. It is true that this method ignores a large percentage of the available speech data when doing the speaker recognition (although all of the data must still be processed to do identification and extraction), but the expectation is that the ability to finely model these select words will produce high performance and make the trade-off worthwhile. Indeed, the use of a small amount of the total data is advantageous with regard to processing requirements.

In order to help ensure that using the keywords leads to good performance, it is important to select these words with great care. One criterion that the words should satisfy is that they occur with high frequency. This is to ensure that they appear with high probability in the speech stream and in a large enough amount that the speaker models can be adequately trained. Another equally important criterion is that the words have good inherent speaker-discriminative qualities. In general, determining a comprehensive list of such words beforehand is difficult, but there do exist certain collections of words believed to possess such characteristics. The discourse markers, backchannels, and filled pauses of conversational speech, for example, are hypothesized to have strong speaker-distinctive attributes because they are produced in a habitual and spontaneous manner [1].

For current text-independent speaker recognition systems, the standard practice is to generate speaker models using Gaussian Mixture Models (GMMs) that represent the distribution of a speaker's speech feature vectors as a mixture of many Gaussians, and by doing so pool the frames of speech into a single "generic-speech" model. For such a system, a Universal Background Model (UBM) is used to model generic non-target speech and this model is adapted to create a target model specific to a given speaker. Sturim *et al.* in [2] apply the text-dependent approach using GMMs and this UBM/Target paradigm to the domain of conversational telephone speech. In their experiments, they look at the performance of different word lists for the same prescribed recognition task. The two lists when compared to the baseline GMM system (i.e, the system using all of the speech) both performed competitively with this baseline system and both contained a significant number

of words from the discourse marker, backchannel, and filled pause categories.

Though the technique used in [2] yielded good results, there are other possibilities within the framework of text-dependent speaker recognition in a text-independent domain. One potential sub-optimality of the above system is the use of the GMM approach to speaker modeling. This "bag of frames" approach assumes the speech frames to be essentially independent. Such a method simply models a "generic" speech frame, and as a result, fails to take advantage of sequential information in the speech stream and, with it, more focused modeling which could aid in the speaker recognition. A natural alternative that captures sequential information and that produces more tightly focused speech states is to use Hidden Markov Models (HMMs) for speaker modeling. Indeed, HMMs have been employed in the context of text-independent speaker recognition systems before [3] [4] [5] [6], but these systems are generally based on simple monophone models or on broad phonetic classes. This is done to ensure *full* coverage of the large-vocabulary, text-independent domains for sufficient speaker modeling. The proposed system is then novel in that it limits coverage to a small set of frequently occurring, habitualized forms that can be very tightly modeled, to the extent that, in terms of performance, this reduced coverage is offset.

1.3 Project Scope and Overview

This project sought to look at the use of keyword Hidden Markov Models to perform speaker recognition in the text-independent domain of conversational telephone speech. The keywords chosen were selected from the categories of discourse markers, backchannels, and filled pauses, with the expectation that these words generally occur with high frequency and are speaker-distinctive. The structure for evaluation takes as its basis the NIST Extended Data Task, a text-independent single-speaker detection task using the Switchboard-1 and Switchboard-2 corpora.

This approach to speaker recognition is analyzed through the design and implementation of a keyword HMM system. The system is presented through a series of experiments detailing the stages of its development and, in the process, indicating the value of the different enhancements made. In addition, the system performance is analyzed in conjunction with other speaker recognition systems through score combination: a baseline GMM system; a Language Model system; a monophone HMM system; and a Sequential Non-Parametric (SNP) system (all to be described in 3.4). This is done to see, among other things, the degree of orthogonality of the information provided by this system relative to the others. Finally, a contrastive system using

monophone HMMs, similar to [5] and [6], is presented and analyzed.

1.4 Outline of Chapters

This report presents the project as follows: Chapter 2 describes the steps taken in building the system and how its performance was evaluated. Specifically, the extraction of the speech features from the waveforms, the extraction of the words from the feature stream, the model training, and the recognition trial scoring are detailed. Chapter 3 explains the specific tasks along with their corresponding corpora, the experiments performed using the system, and the results obtained. Chapter 4 provides some final conclusions from the study and offers possible extensions and future work.

2 Method

2.1 Speaker Recognition as Binary Detection

The topic of speaker recognition is divided into many tasks, and two of the primary tasks are speaker identification and speaker verification. In speaker identification the goal is to identify a speech utterance or segment as having been produced by one of N speakers, each of whom has been previously enrolled through one or more training sessions. In the distinct, though related, task of speaker verification, the objective is to determine whether an utterance has been generated by a putative target speaker. A system can, then, either accept the assertion that the speech belongs to the target speaker or reject it, declaring the utterance was generated by an impostor. It is this latter task of verification that was investigated in this project. Indeed, it is possible to perform speaker identification using techniques derived from speaker verification, making verification more appealing in terms of its ability to be generalized.

Since speaker verification essentially yields YES/NO decisions upon the evaluation of a received signal, it falls into the category of binary detection. Indeed, speaker verification, in many cases, is referred to as speaker detection. The typical approach to binary detection problems employs the use of a Log-Likelihood Ratio (LLR) in conjunction with thresholding to make the ACCEPT/REJECT decision. For a given speech segment X composed of speech feature vectors $\{\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_{N-1}\}$ the Log-Likelihood Ratio, $\mathrm{LLR}(X)$, is given by:

$$\mathrm{LLR}(X) = \log\, p(X|S) - \log\, p(X|\bar{S}) \tag{1}$$

where S represents the model under the hypothesis that the segment was produced by the speaker and $\bar{S}$ represents the model under the opposing hypothesis that it was not produced by the speaker. Should this ratio score exceed some threshold Θ, the speaker is accepted; otherwise the speaker is rejected.

As mentioned previously, for current text-independent verification systems the standard practice is to generate speaker models using speaker-adapted Gaussian Mixture Models (GMMs), a technique popularized by Reynolds *et al.* in [15] and [16]. The GMM models the class-conditional probability density function of a speech frame as a mixture of K class-conditional Gaussian distributions:

$$p(\vec{x}|c) = \sum_{k=0}^{K-1} \pi_{k,c}\, \mathcal{N}(\vec{x}; \vec{\mu}_{k,c}, \Sigma_{k,c}) \tag{2}$$

The class c is taken to be S or $\bar{S}$ as in equation (1) above. To facilitate computation of the probability of the segment, $p(X|c)$, an independence assumption is made regarding the speech frames and individual frame probabilities are multiplied. As any random permutation of the sequence of features yields the same probability, the term "bag-of-frames" is often applied to the approach, as mentioned in 1.2.

To model $\bar{S}$, a Universal Background Model (UBM) GMM is trained by using speech data from a large collection of impostor speakers. For a given system, the UBM is typically fixed and care is taken to ensure that target speakers are excluded from its training data. To model S, the model parameters (generally the Gaussian means) are adapted to the target speaker's training data using Maximum A Posteriori (MAP) adaptation. The benefits of this approach are that it better permits the distinctive differences in the speaker model to be emphasized as those parameters will tend to change the most, and it allows for fuller speaker models as there is typically much more background data than speaker training data. Additionally, the approach deals well with the case of unobserved data; if there is no speaker adaptation data for a particular event, the UBM is simply copied as the speaker model and any log-likelihood scores produced perfectly cancel, removing the influence of the event.

The same general method can be applied using keyword HMMs. Each keyword HMM is trained from frame sequences representing that keyword and so models $p(X|W)$ where W is a word or word sequence (keyword phrase). Since the frame sequences are produced by different speakers, one could obtain $p(X|W,S)$—a speaker-specific model—and $p(X|W,\bar{S})$—a generic background model—as in equation 1.It is possible,then, to compute LLR(X) from the accumulated log-probabilities output by word- or phrase-level HMM speech recognizers. The keyword-specific UBM is trained by using the instances of each keyword found in the background speakers' data and the speaker-specific model is generated using MAP adaptation of the background keyword HMMs.

2.2 Feature Extraction

In both speech and speaker recognition it is desirable to use a parameterization of the speech waveform that is robust and that captures as much of the information necessary to perform recognition while discarding the remainder, such as noise. Though the objectives of the two forms of recognition are quite different—speaker recognition is ultimately concerned with speech "quality" (as it relates to its producer) rather than content, and speech

recognition the opposite—the signal parameterization is typically the same. For the system in this project, speech feature vectors composed of Mel frequency cepstral coefficients (MFCCs) were used. The cepstra were obtained by processing a Hamming windowed version of the waveform with duration 25 ms and which was advanced by 10 ms steps. In the initial experiments, the features consisted of c_0 through c_{12} (with c_0 serving as an energy parameter) and their first differences (deltas). Later experiments extended the vector to include c_{13} to c_{19} along with their corresponding delta coefficients.

Though the standard cepstra are intended to be robust, they demonstrate a susceptibility to the effects of the speech channel. One effective technique to address this problem (and which was applied as an enhancement to the keyword HMM system) is cepstral mean subtraction (CMS). CMS seeks to remove any long-term average from the signal, such as that which would be contributed by a channel response. For CMS to be most effective the average should be computed over speech segments only (silence adversely biases the average) and this was the approach used in the system. The feature extraction and cepstral mean subtraction were performed using the HMM Toolkit, HTK [7].

2.3 Word Extraction

2.3.1 Word selection

The words used for recognition in the original baseline system were selected from among the common discourse markers, backchannels, and filled pauses and are shown in table 1. Again, the motivation for using these words is their two desirable qualities of i) occurring with high frequency in conversational speech, which ensures a sufficient number of examples for model training and testing; and ii) potentially possessing strong speaker-distinctive attributes as a result of their habitual, spontaneous nature.

Discourse Markers	**Backchannels**	**Filled Pauses**
actually, anyway, like, see, well, now	yeah, yep, okay, uhhuh, right	um, uh

Table 1: *Original word list decomposed by category.*

In later experiments the keyword list was extended to include some bigrams from the discourse marker and backchannel categories. Table 2 gives these words. It should be noted that, for the corpora used in the experiments, the original 13 words account for approximately 6% of the total number of

tokens. With the additional 6 word bigrams, the coverage is increased to about 10%.

Discourse Markers	Backchannels
you_know, you_see, i_think, i_mean	i_see, i_know

Table 2: *Additional word list decomposed by category.*

2.3.2 Forced Alignment Word Identification

In order to train models for the keyword HMMs, it was necessary to first locate the keywords within the speech stream. One approach taken to do this was using timing information obtained from the forced alignment to human transcripts by an Automatic Speech Recognition (ASR) system. The alignment attempts to find the most likely assignment of the frames in the acoustic signal to the words in the transcript according to probabilistic models generated from speech recognizer's training data.

The ASR system used was a stripped-down version of the SRI Hub-5 recognizer described in [8] with improvements using the 2001 and 2002 NIST evaluations. The front-end consisted of 13 MFCCs with first and second delta features and employed vocal-tract length normalization (VTLN) along with speaker-level cepstral normalization. Acoustic models were trained on the Switchboard-1 corpus, and the dictionary was derived from the CMU 0.4 dictionary augmented by the addition of multi-words. The language model used was a bigram language model based on a 34k-word vocabulary trained using a mixture of the Switchboard-1, CallHome English, and Broadcast News corpora.

2.3.3 ASR Word Identification

In practice, full human transcriptions are generally not available for the data of interest. Therefore, an alternative approach using ASR word hypotheses and their corresponding start and end times was taken as well. For recognition, the same ASR system was used. For Switchboard-1 the recognizer achieved a word error rate (WER) of about 30% and for Switchboard-2 about 38% [9]. The recognizer was intentionally made to be much simpler than the state-of-the-art in order to minimize the effects of having used speaker recognition training and test data (for the case of Switchboard-1) in its own training.

It should be noted that in both word identification scenarios, no attempt was made to filter words according to their syntactic/semantic roles. Many of the keywords, particularly the discourse markers such as *well*, *like*, and *see*, occur in other roles (e.g., *She did well, I'd like him to see that*) in which their acoustic qualities are probably different. Ultimately, this was an expedient design decision to simplify processing, but there was also a hope that the modeling and the scoring procedures would be robust enough to accommodate this mixed population.

2.4 Model Training

2.4.1 Background Model Training

In keeping with the dominant paradigm in speaker detection, the keyword system used universal background models to model $\bar{S}$ in equation (1). For a given keyword, a UBM was obtained by training an HMM using all of the instances of the word found in the background speaker data. Note that these HMMs were whole-word models, not models built up from shared phonetic components, as is sometimes alternatively done. The training of the HMM occurred in two stages.

In the first stage, the HMM was initialized through an iterative technique involving Viterbi segmentation and parameter updates. With this technique, state means and variances are computed by averaging all the vectors associated with each state. The state transition matrix is estimated by time counts of state occupation. For the Gaussian mixtures of a given state, each feature vector of the state is associated with its highest likelihood Gaussian and the mixture weights are computed according to the ratio of vectors per Gaussian. To start the process, a uniform segmentation of the word to the HMM states is presumed and parameters are initially estimated. For initial estimation of the Gaussians, a modified K-means clustering algorithm is used. In the second stage, Baum-Welch re-estimation of the HMM parameters is performed using the same training data. HTK was utilized for all model training.

Prior to training it was necessary to determine a prototype structure for each keyword HMM. The general HMM topology for all keywords was that of left-to-right sequences with self-loops and no skips. The distribution of each HMM state was modeled as a mixture of four Gaussians with diagonal covariance matrices. It was expected that, with four Gaussians, the models would be small enough to have good focus while large enough to account for sufficient acoustic variation; for example, as could be attributed to different

word usage, as mentioned in 2.3.3 above. The number of states for each HMM was determined heuristically: it was defined to be the smaller of i) the number of phones in the standard pronunciation of the word times three; and ii) the median duration of the word, as expressed in frames, divided by four.

2.4.2 Target Model Training

Each keyword HMM was then adapted to a given target speaker by means of MAP adaptation of the model means. The resulting mean for a state j and mixture component m is given by:

$$\hat{\mu}_{jm} = \frac{N_{jm}}{N_{jm} + \tau}\bar{\mu}_{jm} + \frac{\tau}{N_{jm} + \tau}\mu_{jm} \tag{3}$$

where τ is a weighting of the a priori knowledge to the adaptation speech, N is the occupation likelihood of the adaptation data, μ_{jm} is the speaker independent mean and $\bar{\mu}_{jm}$ is the mean of the observed adaptation data.

In the event that the training data for a target speaker provided no instances of the keyword for adaptation, the unadapted UBM was used for the speaker model as well. This effectively causes the two log-likelihoods (speaker-specific and UBM) to cancel and removes the influence of the word from the overall test score.

2.5 Scoring Test Trials

2.5.1 Recognition Scores

For a given test trial (i.e., a conversation side) , the recognition system must output a single log-likelihood ratio score. For the keyword HMM system this was done in the following manner. The keywords in the test segment were first located using either forced alignment or ASR transcripts as described in 2.3.2 and 2.3.3, respectively. Viterbi alignment and scoring of the target speaker HMM was then performed on the relevant feature sequences (i.e., those corresponding to that particular keyword). The log-probability obtained from the scoring was taken to be a target score. A corresponding UBM score was obtained by similar scoring on the UBM HMM. A token-level log-likelihood ratio score was then computed by subtracting the UBM score from the target score. These scores were then combined over all tokens and all keywords to produce a composite score. The basic system is indicated in figure 1.

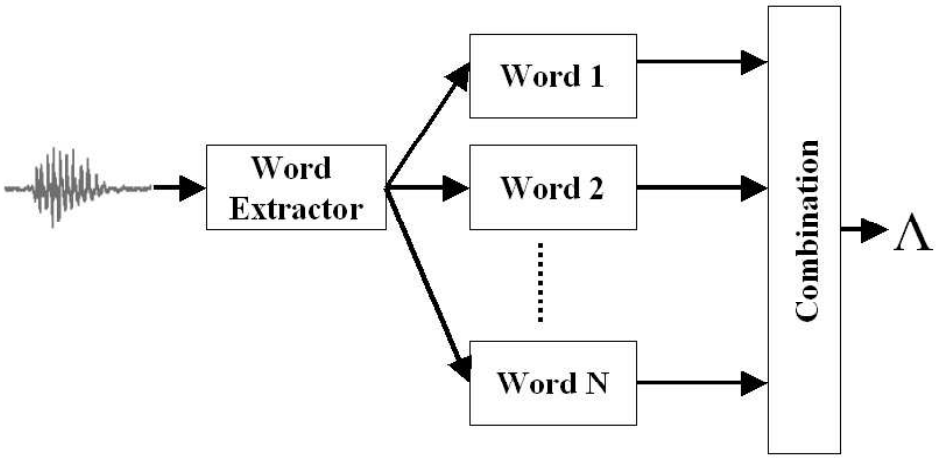

Figure 1: *System Architecture.*

2.5.2 Duration Normalization

As word durations influence log-probability scores, it was necessary to perform some kind of normalization based on duration. Different methods of normalization were examined and as a result different scoring methods were obtained:

Frame normalization
For a frame-normalized score, the composite UBM score (i.e., the sum of all of the individual UBM scores), was subtracted from the composite target speaker score and the result was divided by the total number of frames in all of the keyword instances.

N-best frame normalization
N-best scoring consisted of frame normalization on the N tokens with the highest target log-likelihood ratio scores. The motivation was that the closest matches to the speaker may be the most important in making the detection decision.

Word normalization
For a word-normalized score, frame normalization was performed at the token level and the individual token scores were then averaged.

3 Results

3.1 Task Descriptions

As previously mentioned, the general speaker detection task involves determining whether a given utterance has been generated by a putative target speaker. The specific task used to evaluate the keyword HMM system was based on the Extended Data Tasks of the 2001 and 2003 NIST Speaker Recognition Evaluation (SRE)[10][11], each being a text-independent, single-speaker detection task. For the tasks, speaker models are trained using up to 16 (1, 2, 4, 8, and 16 for 2001; 4, 8, and 16 for 2003) telephone conversation sides containing approximately 2.5 minutes of speech. These models are then used in testing against conversation sides for determining target matches. This procedure marks a departure from earlier NIST tasks in which only 2 minutes of speech were used for model training and test segments averaged 30 seconds in duration. This was done to enable the investigation of techniques that examine phenomena occurring on longer timescales (e.g., prosody, idiolect [12], etc.[13]) and those involving longer-term statistics, and which, as a result, rely on more training data. Generating keywords in quantities large enough for robust modeling is an example of such a phenomenon; it is only within such a framework that one can perform speaker recognition using a constrained word set while not constraining the speech.

For training and testing an N-way cross-validation procedure is used in which the data is partitioned into N sections (or "splits") of approximately equal size and testing proceeds on each partition independently. That is, when a given partition is being tested, data from the other partitions can be used for background model training and any desired normalizations (e.g., T-norm [14], H-norm [16] [15], etc.). To analyze performance, the Detection Error Tradeoff (DET) curve [17], which plots false alarm probability versus missed detection probability for a range of thresholds, Θ, is used. In addition, two summary statistics are reported and examined: the Equal Error Rate (EER), which represents the point at which false alarm and missed detection probabilities are equal; and the minimum of the Detection Cost Function (DCF). The DCF is given by

$$\begin{aligned} C_{DET} &= C_{Miss} \times P_{Miss|Target} \times P_{Target} \\ &\quad + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1 - P_{Target}) \end{aligned} \tag{4}$$

NIST specifies the costs as $C_{Miss} = 10$ and $C_{FalseAlarm} = 1$, and the probability of target as $P_{Target} = 0.01$. It should be noted that for the experiments presented here, the speaker models were trained only using the 8-conversation side specification. This was to provide the best balance be-

tween availability of speaker training data (i.e., number of conversations) and the size of the speaker population, the issue here being that the larger 16-conversation condition involves significantly fewer speakers and consequently higher statistical variance.

3.1.1 The Switchboard-1 Corpus

The 2001 Extended Data Task utilized as its data set the Switchboard-1 Corpus for conversational telephone speech. This corpus consists of about 2400 two-sided telephone conversations among 543 speakers (302 male, 241 female) from all areas of the United States. The data is divided into 6 splits. The system development experiments presented here report results for testing on split 1 using splits 4, 5, and 6 for background model training data. Results over all six splits are reported for the final system using the cross-validation procedure described in 3.1 with the additional specification that testing on splits 1, 2, and 3 involved background models using 4, 5, and 6, and vice versa.

3.1.2 The Switchboard-2 Corpus

The 2003 Extended Data Task utilized phases II and III of the Switchboard-2 Corpus. Phase II consists of about 4500 5-minute telephone conversations involving 679 speakers recruited from Midwestern college campuses. The collection for phase III focused on the American South and involved 640 participants (292 male, 348 female). The data set consists of approximately 2600 telephone conversations. The combined data set is divided into 10 splits. For cross-validation, a similar approach is taken; for testing on splits 1-5, splits 6-10 are used for background model data and the situation is then reversed. This task is considered to be harder both because the demographics of the speakers are narrower as well as because participants were encouraged to use a greater diversity of telephone handsets. Results for this task are presented primarily to provide a contrastive data set to that on which the system was developed.

3.2 System Development Experiments

3.2.1 Experiment 1: Duration Normalizations

The first experiment sought to analyze the performance of the different duration normalizations described in 2.5.2. The best performing method would

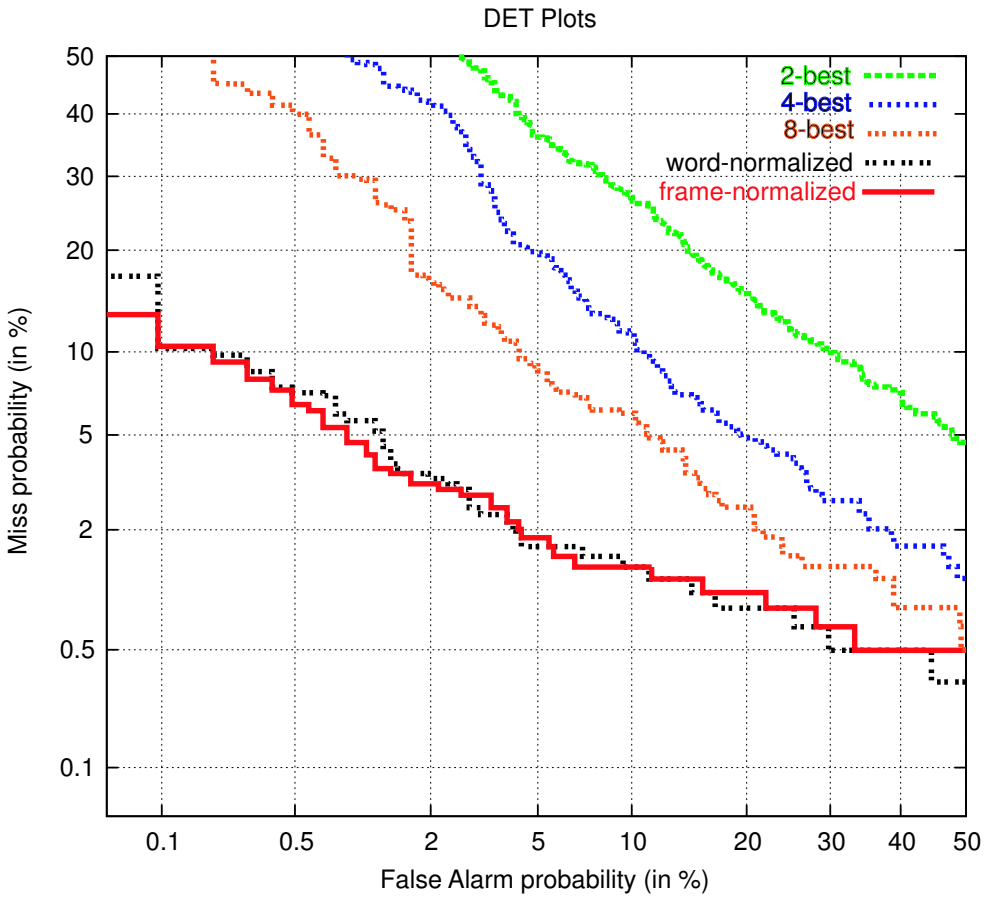

Figure 2: *DET Curves for different duration normalizations*

be the one used for subsequent experiments and would establish a baseline to be compared with enhanced versions of the system. Figure 2 shows the DET curves for frame-normalized, word-normalized, and N-best normalized (for $N = 2$, 4, and 8) scores. The corresponding equal error rates and minimum detection cost function values are given in table 3. From the figure one observes that the word- and frame-normalized scores give nearly equivalent performance. The summary statistics in the table confirm this further as the EERs and minDCFs are the same (2.87% for EER and 0.011 for minDCF). The N-best normalization lags significantly behind these two. N-best normalization does, interestingly, reveal the importance of having a sufficient amount of data not only for model training, but for test trial scoring. Simply increasing the number of tokens from two to four gives a nearly 40% relative improvement in EER and a 23% relative improvement in minDCF. A second doubling in the number of tokens gives a 33% relative improve-

ment in EER and a 40% relative improvement in minDCF, indicating that the scoring component of the system is very much under-supplied with data. Ultimately, the frame-normalization scoring method was chosen and those results used as the baseline. That being said, this baseline yielded surprisingly good performance given the relative simplicity of the system and the small percentage of the total data that was utilized.

Normalization method	**EER(%)**	**Min. DCF**
frame	2.87%	0.011
word	2.87%	0.011
2-best	17.23%	0.075
4-best	10.36%	0.058
8-best	6.96%	0.034

Table 3: *EER and Min. DCF performance for different duration normalizations.*

System	**EER (%)**	**Min. DCF**
baseline	2.87	0.011
baseline + additional words	2.53	0.0071
baseline + higher cepstra	1.88	0.0064
baseline + CMS	1.35	0.0089
combined (true transcription)	1.01	0.0045
combined (ASR output)	1.01	0.0038
final (true transcription)	1.06	0.0054
final (ASR output)	1.25	0.006

Table 4: *System performance for experiments 2 through 7. The first six entries give results for split 1 of Switchboard-1 alone and the last two entries are for all 6 splits.*

3.2.2 Experiment 2: Additional Words

In the initial baseline experiment, the word list consisted of only individual keywords (see table 1 of 2.3.1), 13 in total, from the backchannel, filled pause, and discourse marker categories. There do, however, exist backchannels and discourse markers that consist of multiple words, so the list was expanded to include 6 keyword bigrams (see table 2 of 2.3.1) as well. Each of these word pairs was treated as a single entity and was modeled using a single HMM "word" model. With these additions, the EER gets about a 12% relative reduction and the minDCF a 35% relative reduction. Looking at the DET plots (figure 3) the difference in the EERs of the two curves is minor, but

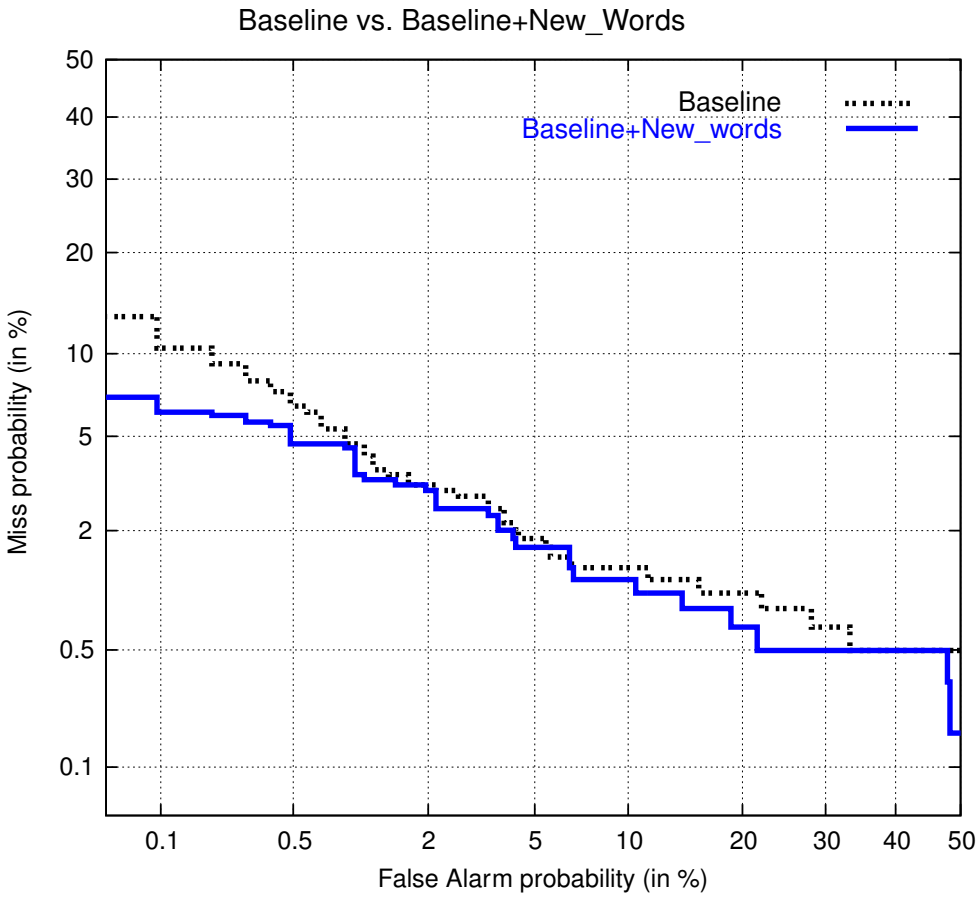

Figure 3: *Baseline versus additional words.*

there is a noticeable drop in low false alarms (the region where the minDCF lies) with the additional words. Table 4 gives the summary statistics for this and all subsequent enhancement experiments.

A complementary way of looking at the results for this system is to display the EER for each word (or phrase) when tested in isolation, along with its frequency of occurrence, as in figure 4. This gives a rough idea of the discriminative capability of each keyword. It is important to look at both EER and frequency, as the individual EERs alone convolve speaker-characterizing ability with the word frequency. For the majority of the words, the EERs obtained lie within a small performance range around 7%, even though the word frequencies vary significantly. The exceptions are the last two entries of both the single-word and word-pair groupings. This is most likely because of the very small number of data observations for these keywords, as indicated in the figure; in other words, the data sufficiency requirement is not

being satisfied. It is particularly interesting that the word yielding the best performance, *yeah*, gives an EER of 4.63% on its own, as compared to the EER of 2.53% for the entire set.

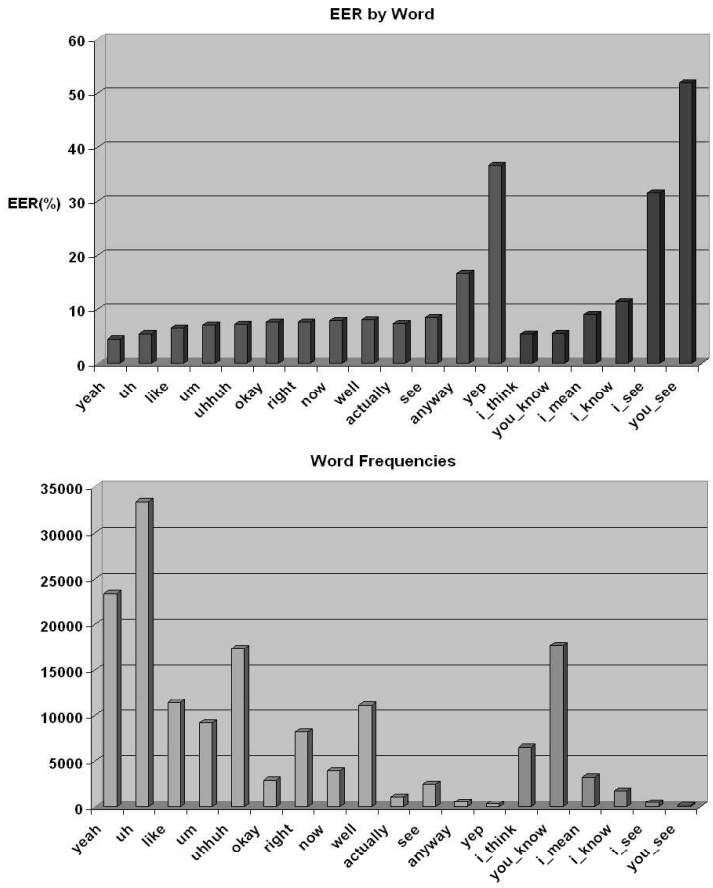

Figure 4: *Individual word/phrase EERs and frequencies.*

3.2.3 Experiment 3: Higher-order Cepstra

Including higher-order cepstral coefficients in the speech feature vector has been shown to give improved performance for numerous speaker recognition systems. It is possible that these coefficients carry more speaker-sensitive information (information regarding pitch, for example), and so it was of interest to apply this enhancement to the baseline system. The input cepstral features were extended to include cepstra up to c_{19}, rather than up to c_{12},

along with their first differences. The result is an impressive absolute reduction in EER of about 1% (34% relative) and in minDCF of 0.0046 (42% relative). The DET curves are displayed in figure 5.

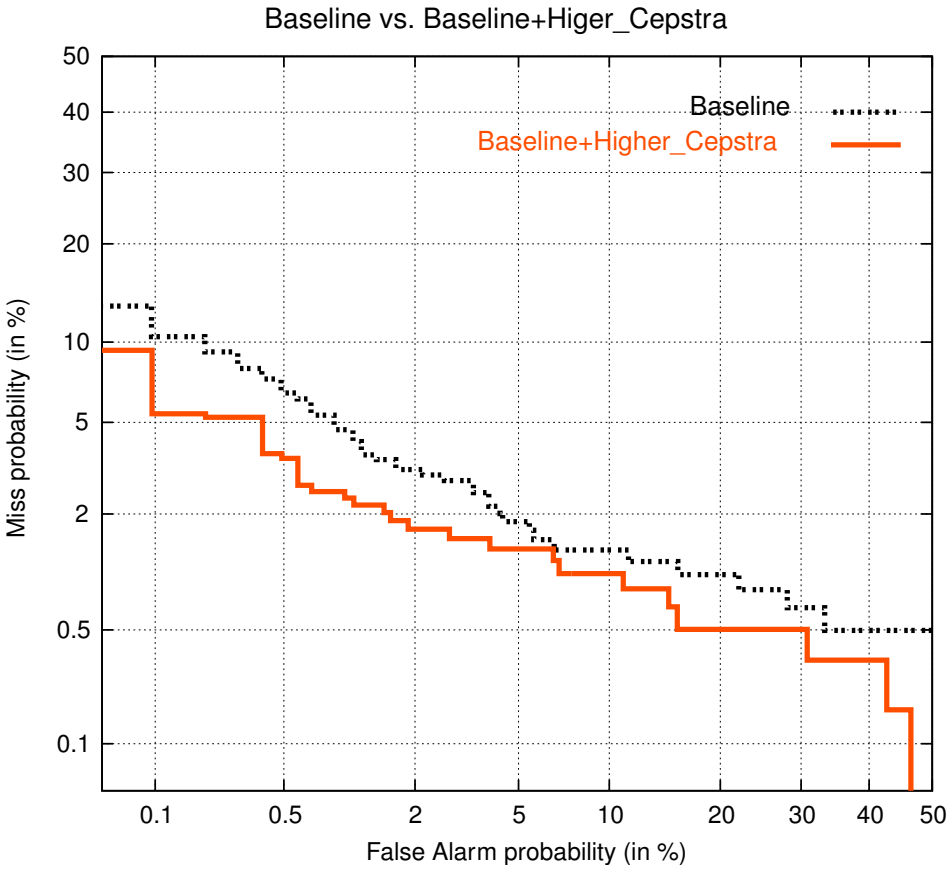

Figure 5: *Baseline versus higher order cepstra.*

3.2.4 Experiment 4: Cepstral Mean Subtraction

As discussed in 2.2, the standard Mel-frequency cepstral coefficients are susceptible to channel effects and CMS is commonly performed to compensate for this. For the task of speaker recognition on conversational telephone speech, channel effects are potentially of great concern because undesirable variability may be introduced by speakers using different handsets. Processing the features using CMS, then, was considered a critical enhancement to the baseline and the performance results for this are shown in the DET curves of figure 6 and in table 4. The minDCF reduction is 19% relative. The

EER reduction is 53% relative—quite large. However, the curves in figure 6 show that performance degrades in the very low false alarm region. It was hypothesized that this was due in part to the very small number of test trials represented in this region of the curve and this hypothesis was supported by later experiments involving all 6 splits of Switchboard-1. Another possible contributing factor is that CMS may actually be removing useful channel information for the cases of speakers who consistently use the same handset.

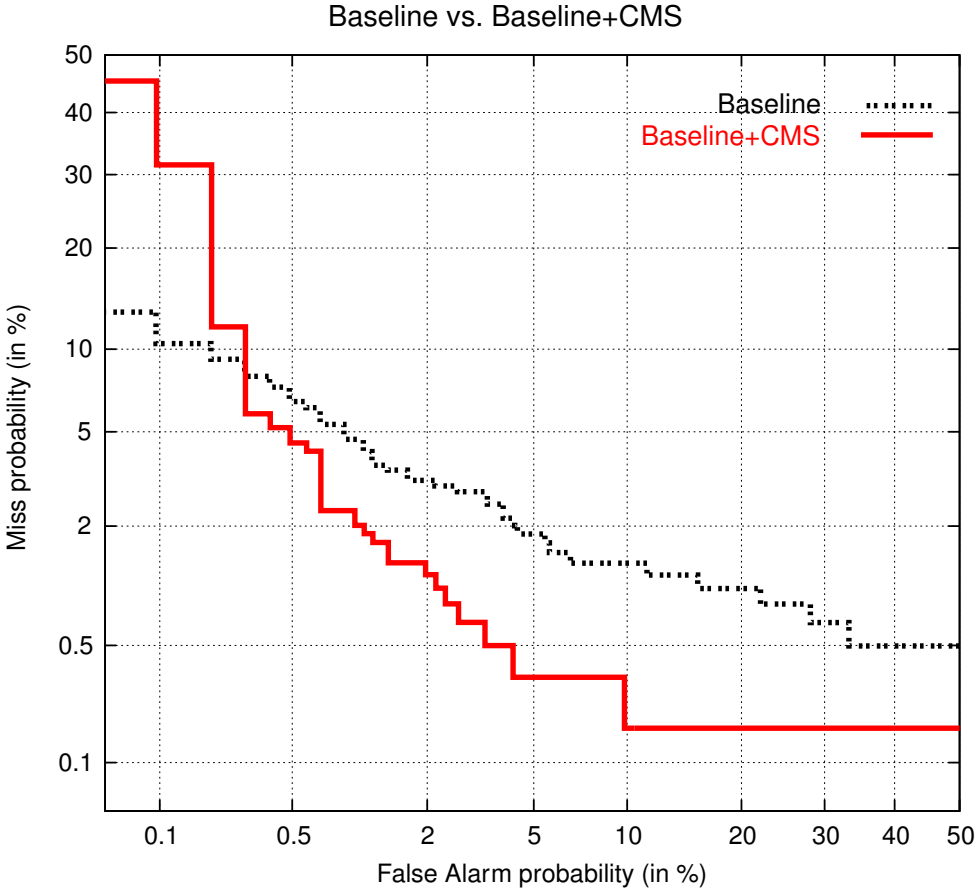

Figure 6: *Baseline versus CMS.*

3.2.5 Experiment 5: Combined System

Having seen the improved performance obtained from each enhancement, a natural next step was to incorporate all of them into a single system. The resulting EER is 1.01%, representing a 65% performance gain over the baseline system. The improvement in minDCF is equally significant: the value is re-

duced by 59%. These results indicate that the information obtained through the different enhancements is, to some extent, complementary. The composite DET curve along with the the contributing stages is displayed in figure 7. The combined system performance is particularly impressive given that certain common score normalizations (Z-norm [18], H-norm [16], or T-norm [14]) were not employed and such a small percentage of each conversation contributed to the system scoring.

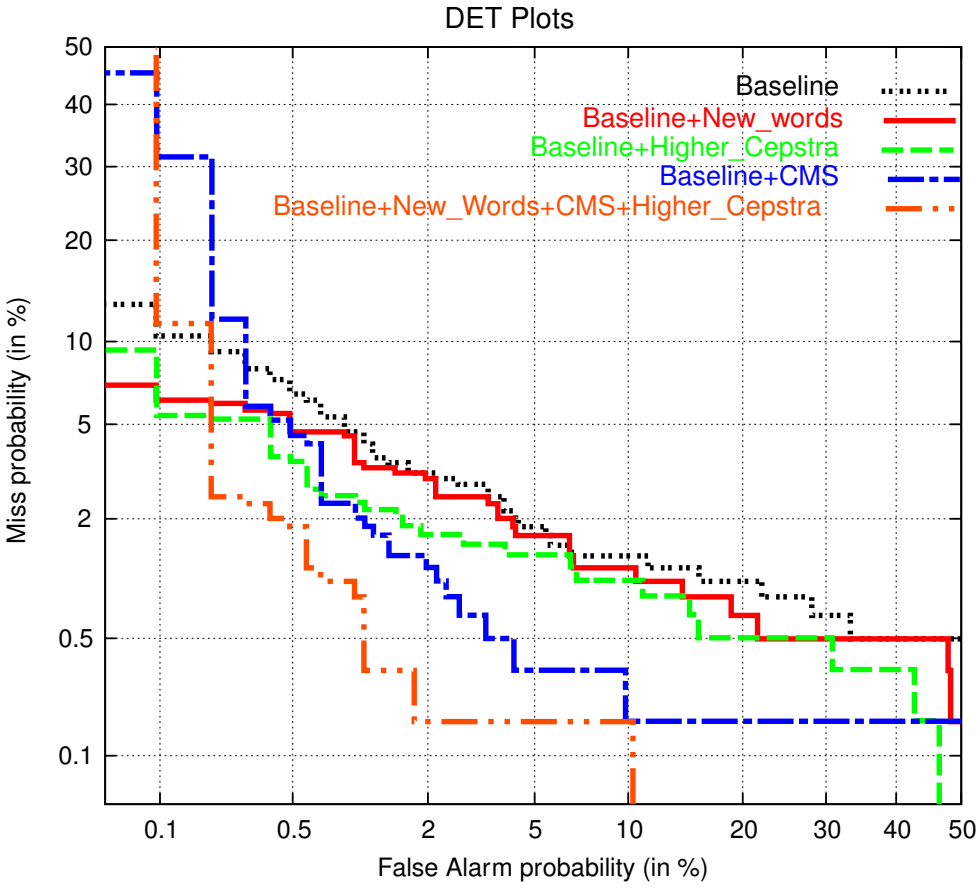

Figure 7: *DET curves for all enhancement experiments.*

3.2.6 Experiment 6: ASR Transcription

In all of the previous experiments, the word identification was based on true (i.e., human-generated) transcription and the word extraction was based on a forced-alignment of the speech stream to these transcripts as described in 2.3.2. While this procedure is useful for system development and validation

of the technical approach, any real-world implementation would necessarily rely on ASR output rather than expert human transcription. This experiment looks to compare the performance of the combined system using true transcription and ASR output. The ASR output was generated using the Switchboard-1 recognizer described in 2.3.2 and 2.3.3. As shown in table 4, the EER and minDCF for the ASR transcription system and the human transcription one are comparable. The corresponding curves are shown in figure 8. These results indicate good performance for fully automatic transcription and help to validate this approach for real-world systems.

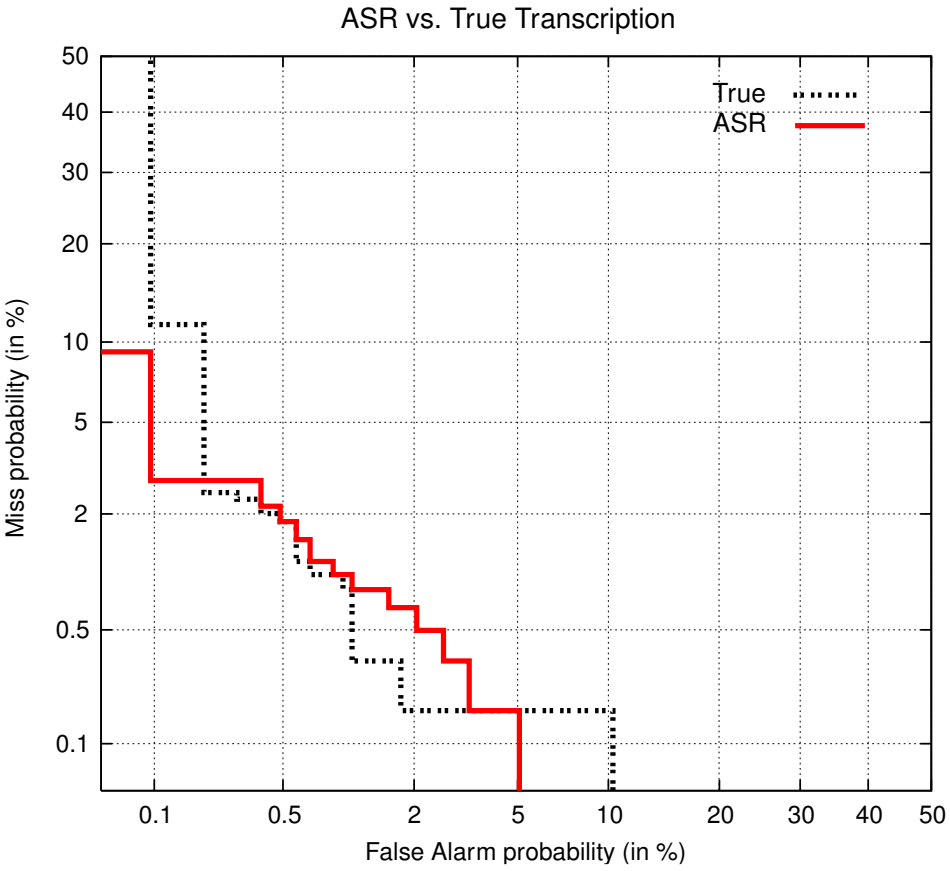

Figure 8: *ASR versus true transcription.*

3.2.7 Experiment 7: Final System Performance

In this experiment, evaluation of the final combined systems is extended to cover all six splits of Switchboard-1. This is contrasted with the analy-

sis of only split 1 for all previous experiments. Using the additional trials for the other splits serves to increase the confidence in the summary statistics that are computed (EER and minDCF) and improves the resolution of the DET curves, as can be seen in figure 9. This figure shows the DET curves for both the ASR and true transcriptions. The EERs are 1.25% and 1.06%, respectively (each compared with 1.01% for split 1 alone), indicating a somewhat greater performance degradation for the ASR transcription system. The minDCFs increase for both systems as well, though their resulting values are similar (0.006 for ASR and 0.0054 for true transcription). Based on these values and the DET curves for the two systems, it appears that their performance when compared to one another remains comparable. Also, looking at the DET curves reveals that the poor performance of the systems in the low false alarm region is no longer evident. This suggests that the phenomenon was at least partly related to the smaller number of trials for a single split.

Having applied all the enhancements to the system, it would be of interest to compare this system to the text-constrained GMM system introduced by Sturim . *et al* in [2] and referred to in 1.2. This latter system also uses a shortlist of keywords from which it extracts acoustic frames and uses only those frames in building and scoring more conventional GMM models. A direct comparison is difficult since they employed different word sets with different frequencies of occurrence, so results are somewhat conflated with coverage statistics. However, the performance seems generally comparable: in the 1% EER range for 2001 Extended Data Task. More careful comparison, having access to the GMM system's scores, using the same wordlists, signal processing, and normalizations, as well as an exploration of which types of words are most valuable to each system, would be illuminating.

3.2.8 Experiment 8: Switchboard-2 performance

In the previous experiment the goal was to see how the keyword HMM system's performance generalized to the rest of the Switchboard-1 data set. Also of interest, though, is how the system's performance would generalize to a corpus that was not used for system development. This experiment consists of a run of the system on the Switchboard-2 corpus described in 3.1.2. This corpus is regarded as more challenging for speaker recognition, in part because of the greater homogeneity of the speaker population. In light of this, a direct cross-corpus comparison of the system performances is not appropriate. A more suitable comparison is how the system's performance on each corpus compares to that of some additional reference system. Here the reference system was a standard cepstral GMM system, made available

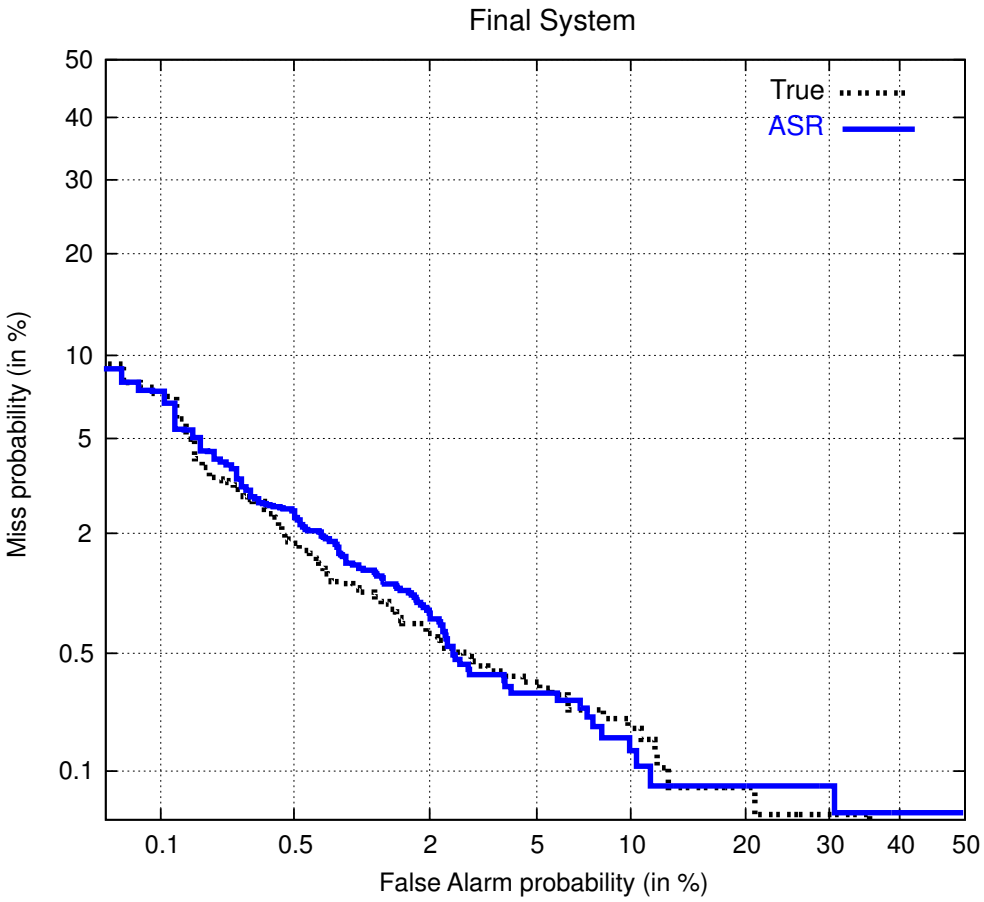

Figure 9: *Final system (ASR and true transcription).*

by SRI [19]. The Switchboard-1 GMM system used for H-Norm followed by T-Norm. The Switchboard-2 GMM system used feature mapping [20] (in lieu of H-Norm) and T-Norm. Both systems use a mixture of 2048 Gaussian components.

The results for the two corpora are shown in table 5. Results for Switchboard-1 are derived from testing on all 6 splits. Those from Switchboard-2 are derived from testing on all 10 splits of that corpus. The first thing to note is that the keyword system lags behind the GMM system for both data sets. An alternate method of comparing results is to look at the relative change of each system's performance across the two corpora. For the GMM system, the EER is approximately multiplied by 2.6 when moving from Switchboard-1 to Switchboard-2 and the minDCF is multiplied by 1.8. For the keyword system, the EER is scaled by 2.5, and the minDCF triples. The performance

degradation for the keyword system, then, is similar that of the GMM for the EER metric, but significantly greater for the minDCF one.

Corpus	System	EER (%)	Min. DCF
SWB-1	Keyword	1.25	0.006
	GMM	0.90	0.005
SWB-2	Keyword	3.11	0.018
	GMM	2.36	0.009

Table 5: *System performance for keyword HMM and GMM systems for Switchboard-1 and Switchboard-2.*

3.3 Monophone HMM System

As mentioned in 1.2, the use of HMMs for speaker recognition is in itself not an innovation for the keyword system. There are, for example, HMM systems that are based on monophone models and broad phonetic classes. These systems serve as an interesting contrast to the keyword HMM system as they represent a trade-off between token coverage—using phones or phonetic classes, full coverage can be achieved—and "sharpness" of modeling—word-conditioned modeling means much more of the acoustic variation is speaker-discriminative.

To analyze the trade-off a monophone HMM system was implemented and tested on split 1 of Switchboard-1. The implementation, in fact, represented only a few changes to the keywords system. Rather than choose 19 keywords to be identified and extracted for training and testing, 43 phones were used. The HMM topology also differed in that all of the phone models consisted of three states (rather than the variable number of states for keywords), with 128 Gaussians per state. A final difference was in the model training. The state models were trained by successive splitting and Baum-Welch re-estimation, starting with a single Gaussian per state. The parameters for this Gaussian were obtained from a flat start using global values computed over a small subset of the background training data.

The results are given in table 6 and figure 10. The EER obtained for the monophone system is 1.16% as compared to 1.01% for the keyword system. The minDCF is 0.0046 as compared to 0.0038. These results suggest that the performance of the two systems is rather similar (particularly when variance of the statistics is taken into consideration). The monophone system, however, uses about ten times the data of the keyword system. This illustrates well the benefits of sacrificing token coverage for improved modeling, as is done with the keyword approach.

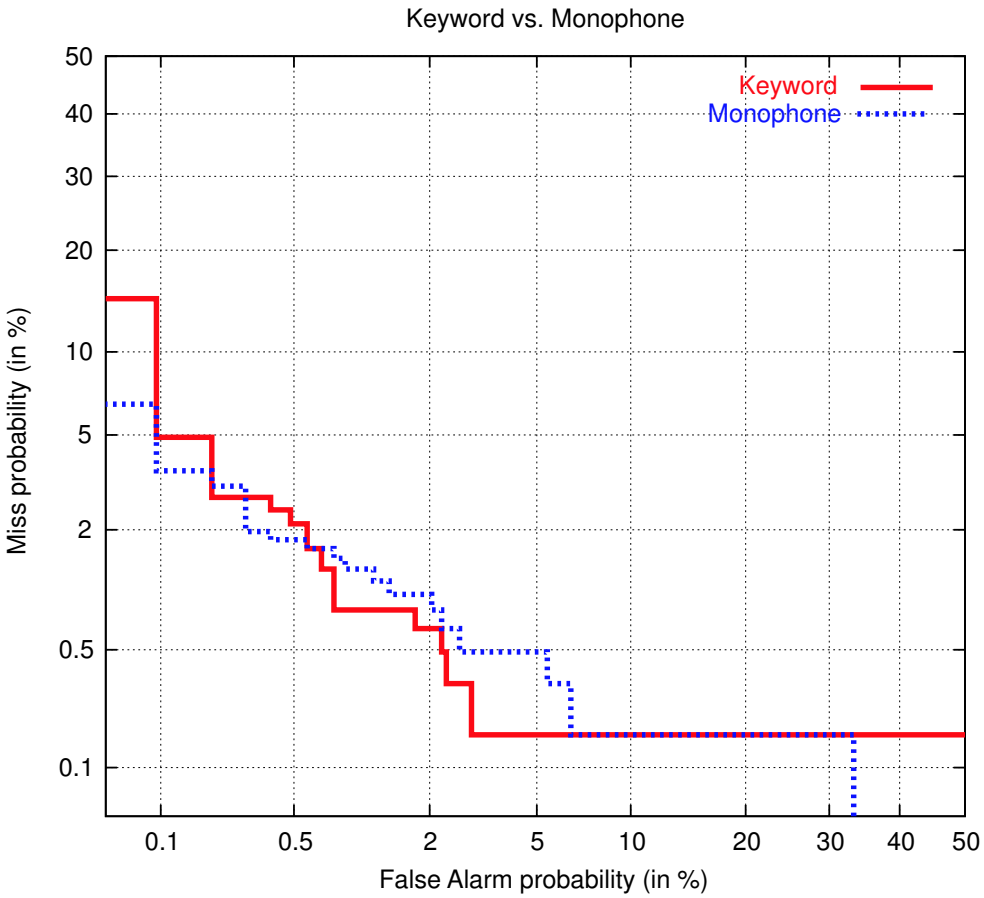

Figure 10: *Keyword HMM versus Monophone HMM system.*

3.4 System Combination Experiments

Through the results of the previous experiments, the keyword HMM system has demonstrated good performance in isolation. Many speaker recognition systems, however, consist of a combination of individual subsystems to take advantage of the different information provided by each of these different sources. The combination often takes place at the score level and a variety of combination techniques is used (multi-layer perceptron, support vector machine, maximum entropy, etc.). It is quite natural, then, to look at how the keyword system combines with other known systems.

For this experiment, each of four systems was combined with the keyword HMM system. The combination was performed on each trial at the score level using LNKnet software from MIT Lincoln Laboratory [21]. A Multi-

System	EER (%)	Min. DCF
Keyword	1.01	0.0038
Monophone	1.16	0.0046

Table 6: *System performance for Keyword and Monophone HMM systems. Results are for split 1 of Switchboard-1.*

Layer Perceptron (MLP) was used to find combination weights through an optimization procedure that minimized the minDCF. The MLP consisted of only an input and output layer (i.e., no hidden layers) and the experiment was performed on splits 1 to 3 of Switchboard-1. The use of additional splits was in anticipation of the high performance (and as a result small number of errors) of the combined systems; by increasing the number of trials—and with it, the number of errors of both types (false alarm and missed detection)—the noise in the statistics is reduced. Since the test data (i.e., the detection trial scores) was used in the training of the weights, these results approximate an upper bound on score combination performance for the systems. The results for the various combinations are given in table 7.

The GMM system is the Switchboard-1 GMM described in 3.2.8 and provided by SRI. The Language Model (LM) system listed is the bigram modeling developed by Doddington in his idiolect work [12]. The system detects speakers based on the distributions (both target speaker and background) of high-frequency bigrams of the training data set, where the bigrams are obtained from ASR transcripts. The monophone HMM system is that described above in 3.3. The system labeled "SNP" is a Sequential Non-Parametric system described by Gillick *et al.* in [22]. This system produces speaker hypotheses based on the Euclidean distance of frame sequences. The system uses ASR output to compare phone unigrams in the test and target conversations, using dynamic time warping to align frame sequences of different lengths.

From the results, it is clear that all systems benefit from the score fusion. This indicates that the information provided by the keyword system in each case complements that of the other systems, even though three of the four systems are also based on acoustic features. The least improved system appears to be the monophone HMM system, which is not surprising owing to its similarity in design to the keyword HMM system. Next are the LM and SNP systems which show similar combined performances, though the systems are extremely different; the SNP system is a purely cepstral approach with no probabilistic modeling while the LM system is a text-based approach modeling bigram occurrences. It is particularly of interest that the LM system, whose stand-alone performance is clearly the worst, can in

combination yield a comparable performance. This illustrates the degree of orthogonality to the keyword HMM system provided by this system. Most improved is the GMM system, indicating the potential benefit of incorporating the keyword HMM system into state-of-the-art systems, whose basis is typically a GMM.

System	Stand-alone Performance		Combined Performance	
	EER(%)	**minDCF**	**EER(%)**	**minDCF**
Keyword HMM	1.08	0.005	-	-
GMM	0.97	0.005	0.43	0.002
LM	9.81	0.056	0.65	0.003
Monophone HMM	1.56	0.007	0.86	0.004
SNP	1.67	0.009	0.70	0.003

Table 7: *Score fusion performance. The second and third columns give system EER percentage and minDCF, respectively, in isolation and the fourth and fifth give the corresponding values when fused with the keyword system. Results are reported for splits 1 to 3 of Switchboard-1.*

3.5 Mixture Variation Experiment

In creating the keyword HMM system, certain design choices, particularly those relating to the HMM structure, were made either heuristically or in an ad-hoc way. This was done because of a lack of a more principled approach to addressing such design issues. Having made those choices and developed an initial system, though, it was then possible to analyze the effect of some of these choices by varying them. This experiment looked at the effect of changing one aspect of the HMM structure—the number of Gaussian mixtures per state—on the system performance. The system was run on all splits of Switchboard-1 with models consisting of 1, 2, 4, 8, and 16 Gaussians per state. The 4-Gaussian topology of the baseline system was determined by trying to balance the level of focus of the modeling—which would call for fewer Gaussians—with the robustness to accommodate significant acoustic variation—which would call for more Gaussians.

As with other experiments, a table of the summary statistics and DET plots are provided (table 8 and figure 11, respectively). It should be noted that, for these results, the keyword *actually* is not included, as the background model for this keyword failed to train in the 16-Gaussian case owing to insufficient data.This lack of data is not surprising given the low frequency of occurrence shown in figure 4.

From both the table and the DET plots, a trend is quite clear: The

system performance improves significantly for each increase in the number of Gaussians per state. Of particular note is that, with 16 Gaussians, the performance in terms of both EER and minDCF essentially matches that of the GMM system given in table 5 of 3.2.8. . A natural question is whether the keyword system can surpass the GMM system in performance with yet another increase in the number of Gaussians. A revision of the word list involving the exclusion of some low frequency words would be necessary before this could be investigated. Regardless, it is clear that four Gaussians per state does not handle all of the acoustic variation. Part of this may be because of the variation that the different semantic/syntactic roles contributes, as mentioned in 2.3.3.

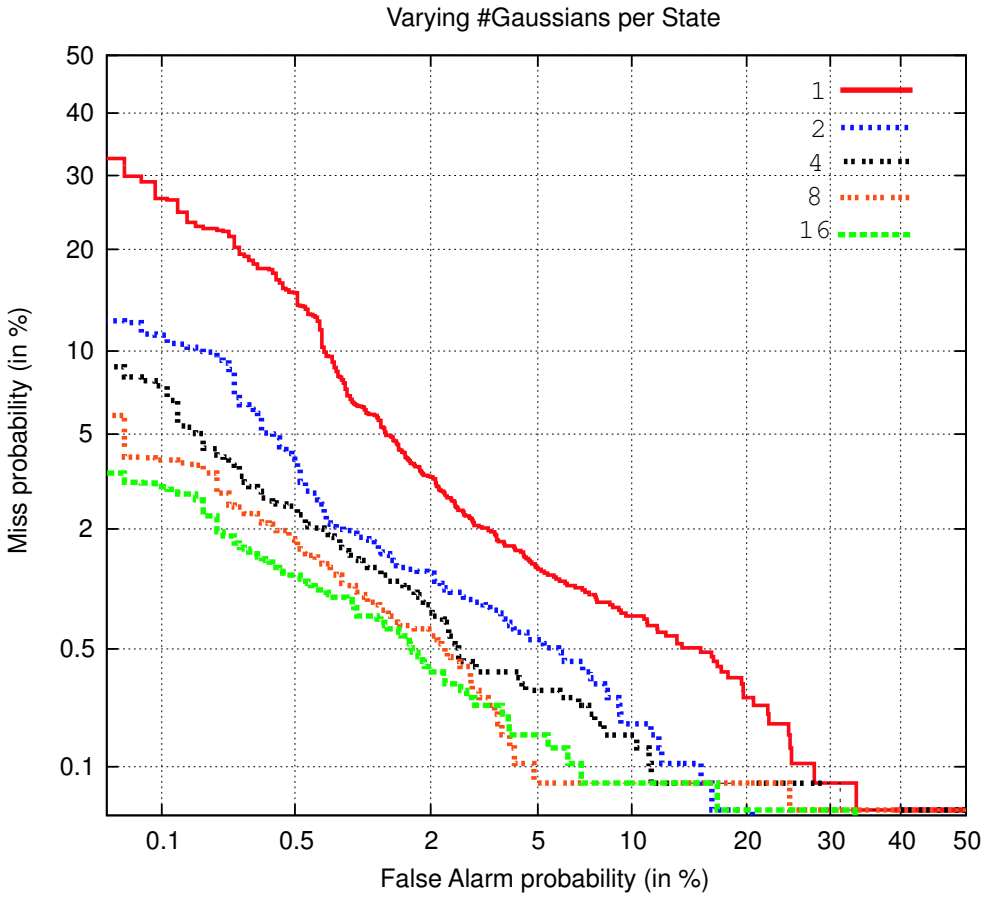

Figure 11: *System performance for varying number of Gaussians per HMM state. The results are given for all 6 splits of Switchboard-1.*

# Gaussians	EER (%)	Min. DCF
1	2.53	0.016
2	1.44	0.009
4	1.25	0.006
8	0.98	0.005
16	0.92	0.004

Table 8: *Keyword system performance for varying number of Gaussians per HMM state. Results are given for all six splits of Switchboard-1. Note that the keyword "actually" was not included due to insufficient data for training.*

4 Conclusions and Future Work

In motivating this project, the question was asked: Is it possible to capitalize on the advantages of text-dependent systems while allowing for the flexibility associated with systems used in the text-independent domain? Based on the results presented here, the answer is unquestionably yes. By modeling select keywords chosen from the backchannel, filled pause, and discourse marker categories using Hidden Markov Models, a well-performing speaker recognition system for the text-independent task of conversational telephone speech was implemented. In its most preliminary form the system achieved an equal error rate performance of 2.87% for split 1 of Switchboard-1 for the NIST Extended Data Task. The project also demonstrated the relative importance of the addition of the following to this system: additional word bigrams, higher order cepstra, and cepstral mean subtraction. CMS proved to be the best addition with respect to EER. Its negative affect on performance in the low false alarm region of the DET curve influenced the minDCF, but it was shown that this was partly due to the number of trials; when extended to all six splits the phenomenon was no longer evident. Degradation when switching from human-generated to automatic transcription was shown to be small, and the final EER for the system was 1.25% on the six splits of Switchboard-1, and under 1% with an increased number of Gaussians.

In addition, a cross-corpus comparison suggested that the good performance observed on the development data set could generalize to other data sets. Combination experiments showed that the system could fuse effectively with other systems, providing complementary information. The contrastive monophone HMM system experiment validated this project's approach by showing that data coverage can be sacrificed for more focused modeling to yield benefits. Finally, variation of the number of Guassians per HMM state indicated that modeling could be improved and that certain ad-hoc and

expedient design choices have resulted in sub-optimal performance.

Given some of these conclusions—in particular the last one—there are a number of possibilities regarding future work for this system. Some are as follows:

1. Continuing the exploration of the different choices for HMM topology, both mixture model makeup as well as the number of states
2. Refining the keyword list in a number of different ways, such as using more words from these classes, highest-frequency words in the domain regardless of role, and/or words and phrases that are particularly characteristic for each individual target speaker
3. Filtering the keyword occurrences to use only the intended functions (discourse marker, filled pause, backchannel) or building separate word models for the separate functions (as in the *like* example of 2.3.3);
4. Performing additional fusion experiments with different systems such as the increasingly popular support vector machine systems described in [23], [24], and [25].

These variations may potentially interact. For example, it may be that by filtering the words by usage, building separate models for each, the models can be more tightly focused and will then require fewer Gaussians in the HMM states.

This project has demonstrated the potential of word-conditional acoustic modeling for speaker recognition in text-independent domains. It is hoped that the use of approaches such as this will gain favor in the speaker recognition research community and lead to well-performing systems that complement those in existence today. In addition, by applying an approach traditional to the text-dependent domain in a text-independent setting, the project marks a step towards narrowing the gap between these two domains. With the availability of large amounts of data such as the Switchboard corpora and with the establishment of the Extended Data Task, the potential to explore this does seem great and with the results obtained in the investigation described here, the future does seem promising.

5 Acknowledgments

There are a number of individuals who were instrumental to the successful completion of this project and to them I would like to extend my deepest gratitude. Thanks to Nelson Morgan, who served as my official advisor and who has given me the opportunity to pursue this interesting and exciting research within his group. Special thanks to the Speaker ID group at SRI, in particular Sachin Kajarekar and Andreas Stolcke, who provided many of the technical components (e.g., ASR output and GMM scores) as well as technical expertise needed for the system implementation and experimentation. In the speech group at the International Computer Science Institute I would like to thank Yang Liu and Chuck Wooters, who provided much assistance during this work's humble beginnings as a course project, along with entire ICSI Speaker ID group—Andy Hatch, Dan Gillick, Steve Stafford, Shawn Cheng, Nikki Mirghafori, George Doddington, and Barbara Peskin—who followed my progress, provided input, and made the work generally more enjoyable. A very special thanks goes to the P.I., Barbara Peskin, with whom I consulted closely and extensively throughout the duration of the project. Finally, I would like to thank my sponsors at the AT&T Labs Fellowship Program and the National Science Foundation (award IIS-0329258) for funding support.

6 References

[1] L.P. Heck, "Integrating High-Level Information for Robust Speaker Recognition," presentation at WS'02, Johns Hopkins University, July 2002.

[2] D.E. Sturim, D.A. Reynolds, R.B. Dunn, and T.F. Quatieri, "Speaker Verification using Text-Constrained Gaussian Mixture Models," *Proc. ICASSP-02*, vol. 1, pp. 677-680, 2002.

[3] M.J. Carey, E.S. Parris, H. Lloyd-Thomas, and S. Bennett, "Robust Prosodic Features for Speaker Identification," *Proc. ICSLP-96*, vol. 3, pp. 1800-1803, 1996.

[4] J.L. Gauvain, L.F. Lamel, and B. Prouts, "Experiments with Speaker Verification over the Telephone," *Proc. Eurospeech'95*, vol. 1, pp. 651-654, 1995.

[5] M. Newman, L. Gillick, Y. Ito, D. McAllaster, and B. Peskin, "Speaker Verification through Large Vocabulary Continuous Speech Recognition," *Proc. ICSLP-96*, vol. 4, pp. 2419-2422, 1996.

[6] F. Weber, B. Peskin, M. Newman, A. Corrada-Emmanuel, and L. Gillick, "Speaker Recognition on Single- and Multispeaker Data," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 75-92, 2000.

[7] HMM Toolkit (HTK): http://htk.eng.cam.ac.uk/

[8] A. Stolcke, *et al.*, "The SRI March 2000 Hub-5 Conversational Speech Transcription System," *Proc. NIST Speech Transcription Workshop*, College Park MD, 2000.

[9] S. Kajarekar, L. Ferrer, A. Venkataraman, K. Sonmez, E. Shriberg, A. Stolcke, and R.R. Gadde, "Speaker Recognition using Prosodic and Lexical Features," *Proc. IEEE Speech Recognition and Understanding Workshop*, pp. 19-24, St. Thomas, U.S. Virgin Islands, 2003.

[10] NIST 2001 Speaker Recognition website:
http://www.nist.gov/speech/tests/spk/2001.

[11] NIST 2003 Speaker Recognition website:
http://www.nist.gov/speech/tests/spk/2003.

[12] G. Doddington, "Speaker Recognition based on Idiolectal Differences between Speakers," *Proc. Eurospeech'01*, vol. 4, pp. 2521-2524, 2001.

[13] D.A. Reynolds, *et al.*, "The SuperSID Project: Exploiting High-level Information for High-Accuracy Speaker Recognition," *Proc. ICASSP-03*, vol. IV, pp. 784-787, 2003.

[14] R. Auckenthaler, M. Carey, H. Lloyd-Thomas, "Score Normalization for Text-Independent Speaker Verification Systems" *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42-54, 2000.

[15] D.A. Reynolds, "Comparison of Background Normalization Methods for Text-Independent Speaker Verification" *Proc. Eurospeech'97*, vol. 2, pp. 963-966, 1997.

[16] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker Verification using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19-41, 2000.

[17] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET Curve in Assessment of Detection Task Performance," *Proc. Eurospeech'97*, vol. 4, pp. 1895-1898, 1997.

[18] D.A. Reynolds, "The Effects of Handset Variability on Speaker Recognition Performance: Experiments on the Switchboard corpus." *Proc. ICASSP-96*, vol. I, pp. 113-116, 1996.

[19] S. Kajarekar, *et al.*, "Speaker Recognition Using Prosodic and Lexical Features" *Proc. IEEE Workshop on Speech Recognition and Understanding*, pp. 19-24, 2003.

[20] D.A. Reynolds, "Channel Robust Speaker Verification via Channel Mapping" *Proc. ICASSP-03*, vol. 2, pp.53-56, 2003.

[21] LNKnet: http://www.ll.mit.edu/IST/lnknet

[22] D. Gillick, S. Stafford, B. Peskin, "Speaker Detection without Models" em Proc. ICASSP-05, vol. 1, pp. 757-760, 2005.

[23] W.M. Campbell, "A Sequence Kernel and its Application to Speaker Recognition" in *Advances in Neural Information Processing Systems*, 2001.

[24] P. Moreno and P. Ho, "A New SVM Approach to Speaker Identification and Verification Using Probabilistic Distance Kernels" *Proc. Eurospeech'03*, pp. 2965-2968, 2003.

[25] A.O. Hatch, B. Peskin, and A. Stolcke, "Improved Phonetic Speaker Recognition Using Lattice Decoding" *Proc. ICASSP-05*, vol. I, pp. 169-172, 2005.