Voice-based Speaker Recognition Combining Acoustic and Stylistic Features

Sachin S. Kajarekar[1] , Luciana Ferrer[3], Andreas Stolcke[1,2], Elizabeth Shriberg[1,2]

[1] SRI International, Menlo Park, CA, USA,
[2] International Computer Science Institute, Berkeley, CA, USA
[3] Stanford University, Electrical Engineering Department, CA, USA

Abstract. We present a survey of the state of the art in voice-based speaker identification research. We describe the general framework of a text-independent speaker verification system, and, as an example, SRI's voice-based speaker recognition system. This system was ranked among the best-performing systems in NIST text-independent speaker recognition evaluations in the years 2004 and 2005. It consists of six subsystems and a neural network combiner. The subsystems are categorized into two groups: acoustics-based, or low level, and stylistic, or high level. Acoustic subsystems extract short-term spectral features that implicitly capture the anatomy of the vocal apparatus, such as the shape of the vocal tract and its variations. These features are known to be sensitive to microphone and channel variations, and various techniques are used to compensate for these variations. High-level subsystems, on the other hand, capture the stylistic aspects of a person's voice, such as the speaking rate for particular words, rhythmic and intonation patterns, and idiosyncratic word usage. These features represent behavioral aspects of the person's identity and are shown to be complementary to spectral acoustic features. By combining all information sources we achieve equal error rate performance of around 3% on the NIST speaker recognition evaluation for 2 minutes of enrollment and 2 minutes of test data.

1 Background

Automatic voiced-based speaker recognition is the task of identifying a speaker based on his or her voice. This task can be performed in a variety of ways. For example, the task might be to choose the speaker that generated the test sample, within a certain set of speakers. This closed-set recognition task is referred to as *speaker identification*. The task might be an open-set problem of deciding whether a given test sample was spoken by a certain speaker. In this case, the test can belong to an infinite set of speakers, one of them being the target speaker and all the others being impostors. This is referred to as a *speaker verification* task. Speaker recognition can also be defined as a mix of speaker identification and verification, where the task is broken into two parts – 1) does the claimed speaker belong to a given set?, and 2) if it does, which speaker from the set is it?

Speaker recognition systems may also be classified as text-dependent and text-independent. Text-dependent systems require a user to say a certain utterance, usually containing text that was present in the training data. This usually implies that text-dependent systems involve a limited vocabulary. There is no such constraint in text-independent systems, where the classification is done without prior knowledge of what the speaker is saying. An example of a text-dependent task is bank account verification where the user says a string of digits. Text-independent speaker verification is typical in surveillance and forensics. In a text-dependent system, knowledge of the words can be exploited to improve performance. Thus, text-dependent speaker recognition usually gives better performance than text-independent recognition for small amounts of training and testing data (on the order of 10 seconds). With more training and testing data, the performance of a text-independent system improves dramatically and has reached as low as 2-3% equal error rate (EER)[1].

This chapter will focus on the text-independent speaker verification task. It describes the evaluation framework used by speaker recognition evaluations (SREs) conducted by the United States National Institute of Standards and Technology (NIST). As an example, SRI's submission to the 2005 SRE is described, as a system that combines multiple voice-based features, both low- and high-level, and may be considered representative of the state of the art. For low-level acoustic modeling, the system uses a cepstral Gaussian mixture model (GMM) system that uses feature normalization to compensate for handset variability, as well as two new approaches to short-term cepstral modeling. It also uses several novel high-level stylistic features and successfully integrates them with low-level feature-based systems. After describing the submitted system in detail, we present further analyses of the relative importance of the low- and high-level subsystems.

[1] EER corresponds to the operating point at which false acceptance and false rejection errors are equally frequent. This is described in detail in Section 3.1.

2 Speaker Verification System Architecture

A general and widely used framework for speaker recognition is shown in Figure 1. Two types of information are provided to the system: test speech and claimed identity. The speech is converted to a set of features. These are the same that are used to train the statistical model for the claimed speaker. Next, a similarity measure of the test features with respect to the claimed speaker model is estimated. The similarity measure is commonly referred to as a score. This score is compared to a precomputed threshold (λ). If the score is greater than the threshold, the claimed speaker identity is accepted, and otherwise it is rejected. This section explains each of the blocks in Figure 1 in detail.

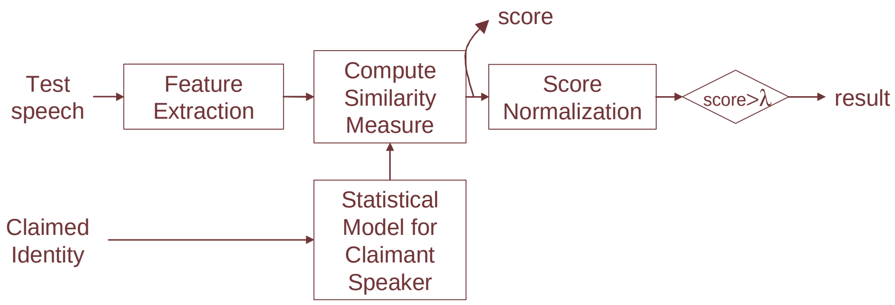

Figure 1 General framework of a speaker verification system

2.1 Feature Extraction

In the feature extraction step, the raw speech is converted into a set of measurements that aim to represent certain characteristics of the speech sample. A general feature extraction algorithm can be thought of as involving three steps: 1) computation of basic features, 2) transformation, and 3) conditioning on external events.

As an example, we can consider estimation of the Mel frequency cepstral coefficients (MFCCs), which are the most widely used features in speaker recognition. The process can be divided into the three steps above. First, the basic features are computed as follows. The speech segment is divided into overlapping segments. The frequency response is computed in each of these segments after applying a Hamming window. The output is then processed with a bank of filters centered uniformly on a Mel scale. The cepstral coefficients correspond to discrete cosine transform of the output of the filters. In the second step, a variety of transforms are applied to the MFCCs to reduce their sensitivity to handset variations or extend their expressive power. Typical transformations include:

computation of delta coefficients, cepstral mean subtraction (CMS), coefficient stream filtering (Hermansky and Morgan 1984) and histogram-based normalization (Pelecanos and Sridharan 1996). Finally, the transformed MFCCs are sometimes conditioned on word or phone identity obtained from an automatic speech recognizer (ASR) and modeled separately.

Although MFCCs are the most popular features, they have two main shortcomings. First, they are estimated from a short window of 10-50 ms. Thus they do not capture longer-term characteristics in the signal. Second, they are obtained directly from spectral parameters (filter-bank energies), making them sensitive to handset variations. Much work has been done in the search for features that can capture longer-range stylistic characteristics of a person's speaking behavior, such as lexical, rhythmic, and intonational patterns with an assumption that the new features will be robust to handset variations and will convey new information not reflected in the cepstral features (Sonmez et al. 1998; Reynolds et al. 2003). Recently, it has been shown that systems based on longer-range stylistic features provide significant speaker information that is complementary to the conventional system (Adami et al. 2003; Ferrer et al. 2003; Shriberg et al. 2005). Some examples of higher-level features will be given in Section 4.

2.2 Statistical Model and Similarity Measure

Until recently, the most commonly used statistical model in speaker recognition was, the Gaussian mixture model (GMM) (Reynolds et al. 2000). A GMM models the distribution of features as a mixture of Gaussian densities, each with a different weight. A typical GMM recognition setup includes two models – the universal background model (or speaker independent model, SI) and speaker dependent model (SD). The SI model is trained using data from a large set of speakers that are usually different from the test population. The SD model is usually obtained by adapting the SI model to the speaker's training (or enrollment) data with maximum a-posteriori adaptation. During testing, the logarithm of the ratio between the likelihood of the SI and the SD models given the data is used as a similarity measure.

Note that the method described above is a generative classification technique. It finds a statistical model of the two classes and generates a score that measures the difference in likelihood of those two models given the data. Generative classification methods do not focus on finding a model that is optimal for the classification task. A significant amount of work has been done on using discriminative modeling techniques for speaker verification. A very successful approach, popularized in the last few years, is based on support vector machines (SVMs) (Campbell 2002; Kajarekar 2005). SVMs are typically trained in binary mode to discriminate between the speaker's data and impostor data. The impostor data consists of several speakers and can coincide with the data used to train the SI GMM. The resulting SVM is a hyperplane separating the two classes in the predefined kernel space. During testing, the same kernel is used to compute a signed distance between the test sample and the hyperplane. This distance is used as a similarity measure or score, with positive values indicating that the sample is on the target speaker side of the hyperplane (but note that the decision threshold may be set to a nonzero value to bias the outcome in accordance with a given decision cost model).

Another shortcoming of the GMM-based approach is that it models the features as a bag of frames ignoring sequence information. Researchers have explored other modeling techniques, such as hidden Markov models (HMMs), to model sequence information (Newman et al. 1996). HMM-based approaches have been shown to outperform the GMM-based approach given enough training data. Another approach has been to model blocks of features, preserving the temporal information (Gillick et al. 1995).

2.3 Score Normalization

It has been observed that the score generated during testing is sensitive to factors such as the length of test data and the mismatch between train and test conditions, among others. To compensate for variation in these factors, the score is usually normalized using precomputed statistics. It is assumed that the influence of a factor leads to a Gaussian distribution of the scores. Therefore, it is compensated by subtracting the mean and dividing by the standard deviation obtained from a suitable sample (see the example below). Note that the statistics can be conditioned on different factors simultaneously. For example, the normalization statistics for the length of the test data can be obtained separately for each gender. During normalization the statistics are chosen based on the gender of the speaker model.

The normalization can be performed either with respect to a test data or with respect to the speaker model. In the former case, the same test is performed using different impostor models and the normalization statistics – mean and standard deviation – are obtained. During testing, the output score for the test data with any given model is normalized by subtracting the mean and dividing by the standard deviation. This type of normalization compensates for variations in test characteristics. The most commonly used normalization of this type is T-NORM(Auckenthaler et al. 2000). In the latter case, each model is tested with different impostor tests to compute the normalization statistics. During testing, the score for the given model with any other test data is normalized by these statistics to compensate for variations in the characteristics of the training data. The most commonly used normalization of this type is H-NORM (Reynolds et al. 2000).

3 Evaluation Procedure

NIST conducts annual speaker recognition evaluations to allow for meaningful comparisons of different approaches and to assess their performance relative to state-of-the-art systems. The evaluation proceeds as follows. NIST sends an evaluation package to the participants containing training and test specifications with the actual data. The amount of training and test data is varied between 10 seconds and 20 minutes and it defines different evaluation conditions. We typically submit results for two conditions: 1-side training with 1-side testing, and 8-side training with 1-side testing. A “side” here refers to one channel in a two-party telephone conversation, containing about 2.5 minutes of speech on average. The results are provided as normalized scores (see Section 2.3) for each trial, along with hard decisions – true speaker trial or impostor trial. The decision is usually based on a threshold computed from the same type of data from last year’s evaluation. NIST computes a decision cost function (DCF, explained below) from these results and ranks submissions in ascending order of cost.

The 2005 NIST SRE dataset (referred to as SRE05) is part of the conversational speech data recorded in a later phase of the Mixer project (Martin et al. 2004). The data contains mostly English speech and was recorded over telephone (landline and cellular) channels. The common evaluation condition is defined as the subset of trials for any of the main conditions for which all train and test conversations were spoken in English using handheld phones (including cordless phones). We submitted results for the 1-side train, 1-side test and the 8-side train, 1-side test conditions. The common condition subset consisted of 20,907 and 15,947 trials for these two conditions, respectively.

3.1 Performance Metrics

A speaker recognition system can make two types of errors: false acceptance (FA) and false rejection (FR). NIST uses these errors to generate a detection error trade-off (DET) curve. This is similar to a receiver operating characteristic (ROC) curve except that the axes are warped so that the curves map to almost straight lines. The performance of a system can be described by any point of interest on a DET curve, or by the complete curve itself. Two points on the DET curve are most commonly used. EER corresponds to the point where both types of errors are equally frequent (implying a cost model where both kinds of errors are equally important). NIST compares the performance of different systems using the detection cost function (DCF), a cost function where FA errors are 10 times more costly than false rejections, and impostors are 10 times more frequent in testing than target speakers. As a result, the point corresponding to a minimum DCF lies in the low-FA, high-FR area of the DET plot. In this chapter, results are presented in terms of the minimum value of the DCF measure over all possible score thresholds and the EER for trials corresponding to the common condition.

4 The SRI Speaker Verification System

In the previous sections, we described a general speaker recognition system. We also described the NIST evaluation framework for measuring performance and for comparing different systems. In this section, we describe SRI's submission to the 2005 NIST SRE as an example. This is a fairly complex system that includes two types of subsystems – some using short-term spectral features (referred to as *acoustic systems*) and some using longer-term stylistic features (referred to as *stylistic systems*). Figure 2 shows the different systems and their relationship with ASR and the combiner. In the figure, the first three systems are acoustic and the last three are stylistic.

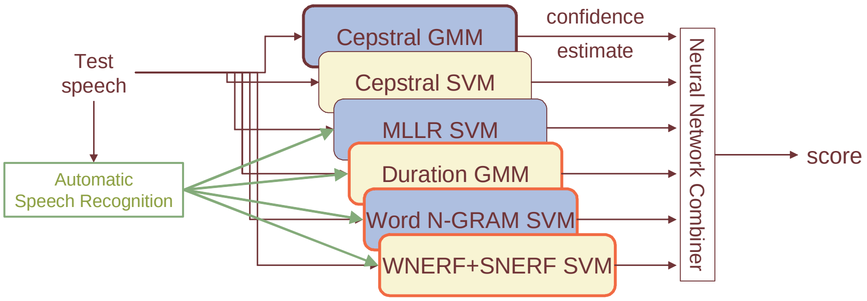

Figure 2 Overall architecture of the SRI submission to the 2005 NIST SRE. Note that Duration GMM refers to two sub-systems, state-duration and word-duration

4.1 Background Data and T-NORM Speakers

Background data for the GMM systems and impostor data for the SVM systems was obtained from the Fisher corpus, NIST 2002 cellular data, NIST 2003 extended data speaker recognition evaluation, and the Switchboard 2 Phase 5 (Cellular part 2) corpus; all corpora are available from the Linguistic Data Consortium (LDC). The Fisher data was divided into three sets: a background set containing speakers with only one recording, and two evaluation sets containing speakers with more than one recording. The

background set contains 1128 speakers with 5 minutes of data per speaker. For NIST 2002 cellular data, the SRE used data from the Switchboard Cellular part 2 corpus. Sixty male and female speakers were used with approximately 2 minutes of data per conversation side. The NIST 2003 extended data SRE used data from Switchboard 2 phases 2 and 3. This data was distributed in 10 nonoverlapping evaluation subsets. We used speakers from subsets 4, 5, 6, and 8 as background data. These contained 425 unique speakers and 4989 conversation sides. Each conversation side has approximately 2.5 minutes of data per conversation side. The Switchboard 2 Phase 5 corpus was used to train background models for some of the systems. Only one conversation side per speaker was used.
Table 1 shows the background/impostor data used by each system. The differences are due to memory constraints, the observed benefit of various data subsets during development, and a desire to avoid data that was also used in training the ASR system (so as to avoid bias).

Table 1: Background data used in each system

Feature, System	Fisher	Switchboard 2, phase 5	NIST 2002 cellular	NIST 2003 extended	
				All data	Only one conversation per speaker
Cepstral, GMM	x		x	x	
Cepstral, SVM	x		x		x
MLLR, SVM	x				x
Duration, GMM	x				x
Word N-gram, SVM	x	x			x
Word+Syllable NERF, SVM	x				x

The score from each system is normalized using T-NORM. Normalization statistics were obtained by using 248 speakers from one of the Fisher evaluation sets. These conversation sides were cut to contain around 2.5 minutes of data to match the evaluation conditions. The same set of T-NORM speakers is used to normalize scores in both 1-side and 8-side training conditions.

4.2 ASR System

The long-term/higher-level features used in the development and evaluation data are based on decoding the speech data with SRI's 3-times-real-time conversational telephone speech recognition system, using models developed for the Fall 2003 NIST Rich Transcription evaluation. The system is trained on Switchboard 1, some Switchboard 2, and LDC CallHome English data, as well as Broadcast News and Web data for the language model; no Fisher data was used in training the ASR models. The ASR system performs two decoding passes, first to generate word lattices, then to apply speaker-dependent acoustic models adapted to the output of the first pass. The speaker adaptation transforms, word-level 1-best recognition output, and word-, phone-, and state-level time alignments were then used to extract features for the various speaker models described in the next section.

4.3 Individual Systems

4.3.1 Cepstral GMM (Baseline) System

The cepstral GMM system uses a 300-3300 Hz bandwidth front end consisting of 19 Mel filters to compute 13 cepstral coefficients (C1-C13) with cepstral mean subtraction, and their delta, double delta, and triple-delta coefficients, producing a 52-dimensional feature vector.

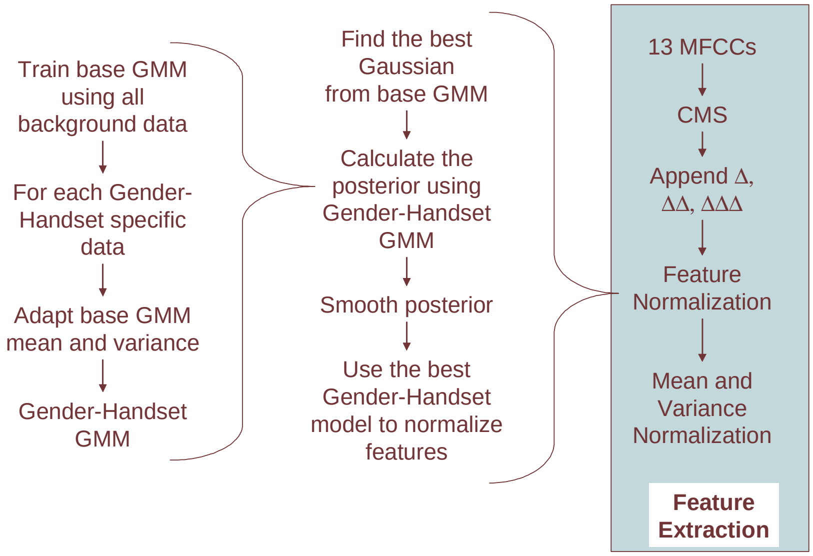
Train base GMM using all background data
For each Gender-Handset specific data
Adapt base GMM mean and variance
Gender-Handset GMM
Find the best Gaussian from base GMM
Calculate the posterior using Gender-Handset GMM
Smooth posterior
Use the best Gender-Handset model to normalize features
13 MFCCs
CMS
Append Δ, ΔΔ, ΔΔΔ
Feature Normalization
Mean and Variance Normalization
Feature Extraction

Figure 3: Flowchart of the feature extraction process for GMM

A gender- and handset-dependent transformation is trained for these features as follows. First, as shown in Figure 3, the distributions of these feature vectors are modeled by a 2048-component GMM (referred to as *base GMM*) using data from the Fisher collection and the NIST 2003 extended data speaker recognition evaluation. The data is also labeled with two genders (male and female) and three handset types (electret, carbon-button, cellular). Using the data for each gender and handset type, a new model is created by adapting the base GMM. Both means and variances are adapted. The base GMM and six gender-handset GMMs are used for feature transformation (Reynolds 2003) EXPLAIN. The main difference from (Reynolds 2003) is that the need for external labels is eliminated by selecting the most likely gender-handset model for each frame and using it for feature normalization. The likelihood is converted into a posterior probability and accumulated over a certain time span. Features after this transformation (described above) are Z-normalized using the mean and variance computed over the utterance.

The base GMM is trained from scratch using the normalized features and is referred to as the *speaker-independent* GMM. *Speaker-dependent* GMMs are adapted from the background GMM using maximum a posteriori (MAP) adaptation of only the means of the Gaussian components. Verification is performed using the 5-best Gaussian components per frame selected with respect to the background model scores. The resulting scores are processed with T-NORM.

4.3.2 Cepstral SVM System

The cepstral SVM system is an equally weighted combination of four SVM subsystems based on the cepstral sequence kernel proposed by (Campbell 2002). All of them use basic features that are similar to the cepstral GMM system. The only difference is that MFCC features are concatenated with only delta and double-delta features, resulting in a 39-dimensional feature vector. This vector undergoes feature-transformation and mean-variance normalization using the same procedure as explained earlier. Each normalized feature vector (39 dim) is concatenated with its second (39x39) and third (39x39x39) order polynomial coefficients. The mean and standard deviation of this vector is computed over the utterance to get a mean polynomial vector (MPV) and a standard deviation polynomial vector (SDPV) for each utterance.

Two SVM systems use transformations of the MPV as follows. The covariance matrix of the MPV is computed using background data, and corresponding eigenvectors are estimated. Since the number of background speakers (S) is less than the number of features (F), there are only S-1 eigenvectors with nonzero eigenvalues. The leading S-1 eigenvectors are normalized with the corresponding eigenvalues. Two SVMs are trained using (1) features obtained by projecting the MPVs on the leading S-1 normalized eigenvectors and (2) features obtained by projecting the MPVs on the remaining F-S+1 unnormalized eigenvectors. A similar procedure is performed on the vector obtained by dividing MPV by SDPV to obtain two additional SVM systems.

All SVM systems use a linear kernel function. During training, to compensate for the imbalance of positive and negative training samples, each false rejection is considered 500 times more costly than a false acceptance. The output scores from these four systems are normalized with T-NORM separately and then linearly combined with equal weights to obtain the final score. We used the SVMlite toolkit (Joachims 1998) to train SVMs and classify instances, with some modifications to allow for more efficient processing of large data sets. The same SVM training and scoring setup was used not only for the cepstral system, but also for the various other SVM-based systems described below.

4.3.3 MLLR SVM System

The MLLR SVM system uses speaker adaptation transforms generated as by-products of the ASR processing as features for speaker recognition (Stolcke et al. 2005; Stolcke et al. 2006). A total of 10 affine 39x40 transforms (two from the first decoding pass, and eight from the second pass) are used to map the Gaussian mean vectors from speaker-independent to speaker-dependent ASR models. The transforms are estimated using maximum-likelihood linear regression (MLLR) (Leggetter and Woodland 1995), and can be viewed as a encapsulation of the speaker's acoustic properties. The transform coefficients form a 15,600-dimensional feature space. To equate the dynamic ranges of the various feature dimensions, each is rank-normalized by replacing the original values with their ranks in the background data, and scaling the ranks to lie in the interval [0, 1]. The resulting normalized feature vectors are then modeled by SVMs using a linear kernel and T-normalized, as described before. One advantage over standard cepstral models is that the features are inherently text-independent. Another benefit is that the ASR features are subject to various normalization and compensation methods, and thus provide information that is complementary to the cepstral GMM and SVM models. For a detailed account of the MLLR-SVM speaker verification approach see (Stolcke et al. 2005), as well as (Stolcke et al. 2005; Stolcke et al. 2006) for recent improvements beyond the system described here.

4.3.4 Word N-gram SVM System

The word N-gram based SVM system aims to model speaker-specific word usage patterns (Doddington 2001), represented via word N-gram frequencies. While Doddington’s original approach modeled N-gram likelihoods, our approach is to treat the N-gram frequencies of each conversation side as a feature vector that is classified by a speaker-specific SVM.

Based on experimentation with the Fisher and Switchboard development data, all orders of N-grams from 1 to 3 were chosen as potential candidates for feature dimensions. Every unigram, bigram, and trigram occurring at least three times in the background set was included in the N-gram vocabulary of the system. This resulted in a vocabulary of 125,579 N-grams. The relative frequencies of each N-gram in the conversation side form the feature values. As in the MLLR system, the features are rank-normalized to the range [0,1]. SVM modeling and T-NORM are carried out as described before.

4.3.5 Duration GMM System

This system models a speaker's idiosyncratic temporal patterns in the pronunciation of individual words and phones (Ferrer et al. 2003), inspired by earlier work on similar features for conversational speech recognition (Gadde 2000).

The duration modeling framework can be outlined as follows. We assume that a set of tokens is given, which can correspond to phones, syllables, words, word classes, and so on. For each of these tokens we create feature vectors containing the duration of a set of smaller units that constitute these tokens. For example, if the tokens correspond to words, the smaller units can be the phones in the words. If the tokens are phones, the smaller units can be the states of the HMMs that represent the phones in the speech recognizer. Figure 4 shows a sketch of those two cases.

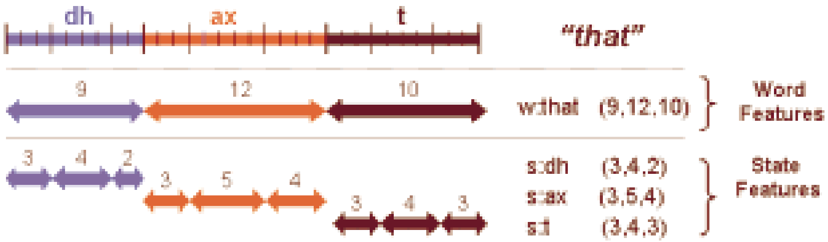

Figure 4: Examples of duration features

The goal is to obtain, for each target speaker, a model of the duration pattern for each token. We do this by first training a universal (speaker-independent) model for each token. For this, we collect duration vectors for all the occurrences of a certain token in the background data. Using these vectors we train a GMM model for the token, which corresponds to a *standard* or *universal* pattern of durations for that token. Finally, we obtain the speaker models for that token by doing MAP adaptation of the means and weights of the Gaussians in the universal token model, using the duration vectors for all occurrences of that token in the speaker's data. We do this for every token, obtaining a collection of universal token models and a collection of adapted token models for each speaker.

Given a test sample, the score for a certain set of tokens is computed as the sum of the likelihoods of the feature vectors in the sample corresponding to those tokens given the corresponding token models. This number is then divided by the total number of component tokens, that is, the total number of constituent units that were scored. This is better than dividing by the number of top-level tokens because not all the models are formed by the same number of units in the case of the word features. The final score is obtained as the difference between the speaker-specific model scores and those from the corresponding background models. This step makes the scores less dependent on what was spoken. Furthermore, excluding from the final score those tokens for which the speaker model had very few samples was found to improve performance. The score is further normalized using T-NORM.

SRI's duration system consists of two separate sets of tokens, one set for the most frequent words in the background data (with phones as component units) and another for phones (with HMM states as component units). For each set, separate scores are computed, resulting in two separate systems that are later combined at the score level.

Some refinements of the above method include the use of context-dependent token models, where tokens are determined by both the word identity and a discretized version of the pause after the word, and a backoff strategy by which context-dependent token models are replaced by context-independent token models when the number of samples available for adaptation of the context-dependent model is too small.

4.3.6 Word and Syllable NERF (WNERF and SNERF) Subsystem

This high-level system is intended to model the speaker's prosodic characteristics. Non-uniform extraction regions (NERs) (Kajarekar et al. 2004; Shriberg et al. 2005) are defined as feature extraction regions that are computed based on events that can be automatically detected from the speech signal, such as pauses, peaks in the pitch track, or a specific set of words. For each of these NERs, various features (NERFs) are extracted using the word and phone alignments from the speech recognizer, pitch and energy tracks, and any other input of interest.

In our current work, NERs are defined by the syllables in the utterance, which are extracted automatically from the recognition output. Pitch, energy, and duration features are computed over each syllable and its surroundings, based on linguistic knowledge about which prosodic characteristics are likely to help differentiate speakers. These features are modeled using SVMs after applying a transformation that converts the syllable-level features into a single conversation-level vector containing a detailed description of the distribution of the features and their sequences.

For each feature, a set of bins is defined by choosing thresholds such that the resulting bins are approximately equally probable in the background data. Then, for each conversation side, the counts corresponding to each of these bins, normalized by the total number of syllables in the data, constitute the transformed vector. This vector is augmented by the normalized counts of the most frequent sequences of two and three bins. These sequences also include pauses, which are quantized into hand-chosen bins. The transformed vector thus includes a description of the characteristics of the features' development over time. In (Shriberg et al. 2005) we presented a detailed analysis of the contribution of the different types of features and different sequence lengths to the performance of the system.

Our current system includes a second set of features corresponding to word-constrained NERFs, which use the same prosodic features as above, but constrain extraction location to specific sets of words. The transformed versions of both sets of features are concatenated and rank-normalized and modeled using SVMs. As shown below, the resulting system is currently the best performing of our high-level systems and the one that gives the largest improvement when combined with the low-level systems.

4.4 System Combination

The scores of the individual systems are combined using a neural network (NN) with a single feed-forward layer that uses a sigmoid output node during training and a linear output for test set predictions. The combiner is trained to achieve minimum squared error with output labels 0 (impostor) and 1 (target). Target and impostor priors are set to 0.09 and 0.91 during training in order to optimize DCF. The combiner is trained using the scores obtained for SRE04 data, which we assumed to be a reasonably representative of the SRE05 data.

4.5 Individual System Results

Table 2 shows the performance of different component systems from the submission. Results show that the cepstral GMM system gives the best performance in the 1-side training condition. Among systems using high-level features, the SNERF system gives the best performance in the 1-side condition. For the 8-side training condition, the cepstral SVM system gives the best performance among cepstral systems, and the SNERF system gives the best performance among the systems that use high-level features.

Table 2: Performance of component systems for SRE05 (DCF refers to minimum DCF).

System	Short name	1-side training		8-side training	
		DCF$_{x100}$	%EER	DCF$_{x100}$	%EER
Cepstral GMM	CepGm	2.48	7.17	1.69	4.91
MLLR SVM	CepMl	2.52	10.34	1.20	5.50
Cepstral SVM	CepSv	2.68	7.26	1.03	3.05
SNERF	StySn	5.22	14.06	2.75	6.52
State Dur	StySd	6.03	15.36	3.19	8.02
Word Dur	StyWd	7.83	19.23	3.74	8.62
Word N-gram	StyWn	8.60	24.58	4.84	11.25

4.6 Combination Results

With all these systems available for combination and various different ways of combining them, several questions arise: Which systems are more important for the combination? Can we ignore some of them without losing accuracy? Does the importance of the systems depend on the amount of training data?

Table 4 shows combination results for some meaningful subsets of systems. The first line corresponds to the cepstral GMM system alone. This conventional speaker recognition system is commonly used as the baseline against which new systems are compared. The second line shows the combination results of that system with the two novel cepstral systems, the cepstral SVM and the MLLR SVM. The combined system achieves an improvement in the DCF of 33% for the 1-side condition and 53% for the 8-side condition. Similar improvements are obtained when combining the baseline with the four stylistic systems: word N-gram, SNERF, and both duration systems. Finally, when all systems are combined, the relative improvement over the baseline alone is 47% in the 1-side condition and 67% in the 8-side condition. Clearly, the benefit of the new systems, both cepstral and stylistic, increases as more data is available for training.

Table 3: Performance for the cepstral GMM (baseline) and the combination of that system with the rest of the cepstral systems, the stylistic systems and all systems together

Systems being combined	1-side training	8-side training

	DCFx100	%EER	DCFx100	%EER
Baseline	2.48	7.17	1.69	4.91
Baseline + new cepstral	1.66	4.61	0.80	2.45
Baseline + stylistic	1.77	4.89	0.83	2.45
All systems combined	1.31	4.10	0.56	2.03

Table 4 and Table 5 show the best combination results when we allow a fixed number of systems to be used by the combiner for both training conditions. Each line in these tables shows which systems lead to the best performance when *N* systems are allowed. We start with the cepstral GMM as the 1-best system. For 1-side training condition (Table 4), the two best systems are two cepstral systems. However, for 3-best combination, the cepstral GMM is replaced by the cepstral SVM, and the SNERF system is chosen in addition. The 4-best combination again includes the cepstral GMM system. The best performance is obtained using 5 of the 7 systems, without the cepstral GMM and state duration systems. This indicates two things: first, the state duration system is probably redundant once the other systems are being used, and second, our combiners are not able to handle redundant features well, by overfitting the training data and not always generalizing to new test data. Ideally, we should be able to detect such cases and ignore subsystems that are not needed. To this end, further research on system selection and more robust combiners is needed.

Table 4: Best possible N-way combinations for the NN combiner for the 1-side training condition. System names refer to those defined in Table 2.

N	Cep Gm	Cep Ml	Sty Sn	Sty Wd	Cep Sv	Sty Wn	Sty Sd	DCF x10
1	🟧							2.47
2	■	■						1.98
3		■	■		■			1.67
4	■	■	■		■			1.58
5		■	■	■	■	■		1.49
6	■	■	■	■	■	■		1.60
7	■	■	■	■	■	■	■	1.47

Table 5: Same as Table 4 but for the 8-side training condition

N	Cep Sv	Sty Sn	Cep Ml	Sty Wn	Sty Wd	Cep Gm	Sty Sd	DCF x100

1	🟧							1.03
2	⬛	⬛						0.75
3	⬛	⬛	⬛					0.66
4	⬛	⬛	⬛	⬛				0.61
5	⬛	⬛	⬛	⬛	⬛			0.59
6	⬛	⬛	⬛	⬛	⬛	⬛		0.59
7	⬛	⬛	⬛	⬛	⬛	⬛	⬛	0.60

Similar observations can also be made for the 8-side training condition. Table 5 shows that even though the best *N* systems for each value of *N* are chosen independently so as to optimize the performance for that number of systems, the subset of systems chosen for a certain *N* includes the subset chosen for *N-1* systems for all cases except N=3 and N=5 for the 1-side condition. This is an important result. There is nothing forcing, say, the best 2-way combination to include the single best system, rather than two other systems that, when combined, give better performance than the best system alone. But given that the results turned out this way, we can very easily rank the importance of the seven systems by looking at the order in which they are added as we allow more systems in the combination.

From the tables we see that the order in which systems are chosen is highly dependent on the amount of training data. In Table 2, we can see that the performance of the subsystems is, without exception, closer to the cepstral GMM in the 8-side condition than in the 1-side condition. For example, the EER of the SNERF system is twice that of the cepstral GMM for the 1-side condition, while it is only 30% worse for the 8-side condition. This explains the bigger relative improvement obtained from combining the baseline with the other systems for the 8-side condition than the 1-side condition (Table 4) and it also explains the difference in the order in which the systems are added for those two conditions. In Table 4 and Table 5 both the SNERF and the Word-Ngram systems are added earlier in the 8-side condition than in the 1-side condition where they have worse performance relative to the baseline. Overall we see that both factors, the performance of the system with respect to the baseline and the amount of new information the system conveys about the speakers, affect which system is chosen next. This qualitative observation accounts for the alternating pattern by which stylistic and cepstral systems are added to the combination.

5 Summary and Future Directions

In this chapter, we have described a general speaker recognition system. The system consists of three blocks: feature extraction, similarity measure computation, and score normalization. We have also described these blocks in detail giving state-of-the-art approaches used in each block. As an example of an actual speaker recognition system, we presented SRI's submission to the 2005 NIST SRE evaluation. The system consists of three systems using acoustic features and an equal number of systems modeling stylistic aspects of speech. An analysis of the results showed the relative importance of both cepstral and stylistic systems being combined. It was found that improvements over the baseline cepstral system when combining all subsystems range from 47% to 67%, with larger improvements for the 8-side condition. The overall results justify and encourage the development of nonstandard systems utilizing prosodic or lexical features, or which model the spectral features in a manner different from GMMs.

There are several exciting research directions currently being explored in the area of speaker recognition. One of the most important is related to the problem of intersession variability (Kenny et al. 2005). As mentioned before, handset variability was considered the most important source of degradation for the speaker recognition system. This source is now subsumed under session variability which can arise from changes in channel, choice of words, or even choice of language (for multilingual speakers) . Recently proposed techniques aiming to address the issue have shown significant improvements in for both GMM- and SVM-based systems (Vogt et al. 2005; Hatch et al. 2006; Kenny et al. 2006). Another important area of research is the exploration of robust combination strategies. In addition to NNs, SVMs with different kernels are being explored as possible alternatives. The combiner is enhanced by incorporating external variables for conditioning the data (Ferrer et al. 2005; Solewicz and Koppel 2005). Finally, research is continuing to explore new high-level stylistic features, as well as feature selection methods for effective modeling of this very large feature space.

6 Acknowledgements

This work was funded by a DoD KDD award via NSF IRI-9619921 and IIS-0544682 and NASA Award NCC 2-1256. We thank our SRI colleagues - Kemal Sonmez, Anand Venkataraman, Harry bratt and Ramana Rao Gadde for their help. We also thank our colleagues from International Computer Science Institute for helpful discussions. We acknowledge help from the community (including NIST) as well for helping us and for making voice-based speaker recognition such an exciting research area. Note that the views herein are those of the authors and do not reflect the views of the funding agencies.

References

Adami, A., et al. (2003). Modeling Prosodic Dynamics for Speaker Recognition. ICASSP.

Auckenthaler, R., et al. (2000). Improving a GMM Speaker Verification System by Phonetic Weighting. Proc. of ICASSP, Phoenix.

Campbell, W. M. (2002). Generalized Linear Discriminant Sequence Kernels for Speaker Recognition. ICASSP, Orlando.

Doddington, G. (2001). Speaker recognition based on idiolectal differences between speakers. Eurospeech, Aalborg, Denmark.

Ferrer, L., et al. (2003). Modeling Duration Patterns for Speaker Recognition. Eurospeech, Geneva.

Ferrer, L., et al. (2005). Class-based Score Combination for Speaker Recognition. Eurospeech, Lisbon.

Gadde, V. R. R. (2000). Modeling Word Durations. International Conference on Spoken Language Processing, Beijing.

Gillick, D., et al. (1995). Speaker Detection without Models. ICASSP, Philadelphia, PA.

Hatch, A., et al. (2006). Within-class Covariance Normalization for SVM-based Speaker Recognition. ICSLP, Pittsburgh, PA.

Hermansky, H. and N. Morgan (1984). "RASTA Processing of Speech." IEEE Transactions on Speech and Audio **2**: 578--589 author H. Hermansky.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. European Conference on Machine Learning.

Kajarekar, S. (2005). Four Weightings and a Fusion: A Cepstral-SVM System for Speaker Recognition. ASRU, San Juan, IEEE.

Kajarekar, S., et al. (2004). Modeling NERFs for Speaker Recognition. Odyssey 04 Speaker and Language Recognition Workshop, Toledo, Spain.

Kenny, P., et al. (2005). Factor Analysis Simplified. ICASSP, Philadelphia, PA, IEEE.

Kenny, P., et al. (2006). Improvements in Factor Analysis Based Speaker Verification. ICASSP, Toulouse, France, IEEE.

Leggetter, C. and P. Woodland (1995). "Maximum Likelihood Linear Regression for Speaker Adaptation of HMMs." Computer Speech and Language **9**: 171-186.

Martin, A., et al. (2004). Conversational Telephone Speech Corpus Collection for the NIST Speaker Recognition Evaluation 2004. IAD.

Newman, M., et al. (1996). "Speaker Verification through Large Vocabulary Continuous Speech Recognition." ICSLP.

Pelecanos, J. and S. Sridharan (1996). Feature Warping for Robust Speaker Verification. 2001: A Speaker Odyssey: The Speaker Recognition Workshop, Crete, Greece, IEEE.

Reynolds, D. (2003). Channel Robust Speaker Verification via Feature Mapping. ICASSP, Hong Kong, China, IEEE.

Reynolds, D., et al. (2003). SuperSID: Exploiting High-level Information for High-performance Speaker Recognition□, http://www.clsp.jhu.edu/ws2002/groups/supersid/supersid-final.pdf. ICASSP, Hong Kong, IEEE.

Reynolds, D., et al. (2000). "Speaker Verification Using Adapted Mixture Models." Digital Signal Processing **10**: 181-202.

Shriberg, E., et al. (2005). "Modeling Prosodic Feature Sequences for Speaker Recognition." Speech Communication **46**(3-4): 455-472.

Solewicz, Y. A. and M. Koppel (2005). Considering Speech Quality in Speaker Verification Fusion. INTERSPEECH, Lisbon, Portugal.

Sonmez, K., et al. (1998). A lognormal model of pitch for prosody-based speaker recognition. Eurospeech, Rhodes, Greece.

Stolcke, A., et al. (2006). Improvements in MLLR-Transform-based Speaker Recognition. IEEE Odyssey 2006 Speaker and Language Recognition Workshop, San Juan, Puerto Rico.

Stolcke, A., et al. (2005). MLLR transforms as features in speaker recognition. Eurospeech, Lisbon, Portugal.

Vogt, R., et al. (2005). Modeling Session Variability in Text-independent Speaker Verification. Eurospeech, Lisbon, Portugal, ISCA.