A Study into Automatic Speaker Verification with Aspects of Deep Learning

by

Keith A. Jellyman

08/05/2018

A thesis submitted to the
UNIVERSITY OF BIRMINGHAM
in candidature for the degree of
MSc by Research

Supervisor: Prof. M. Russell

Department of Electronic, Electrical and Systems Engineering
School of Engineering
UNIVERSITY OF BIRMINGHAM

Abstract

Advancements in automatic speaker verification (ASV) can be considered to be primarily limited to improvements in modelling and classification techniques, capable of capturing ever larger amounts of speech data.

This thesis begins by presenting a fairly extensive review of developments in ASV, up to the current state-of-the-art with i-vectors and PLDA. A series of practical tuning experiments then follows. It is found somewhat surprisingly, that even the training of the total variability matrix required for i-vector extraction, is potentially susceptible to unwanted variabilities.

The thesis then explores the use of deep learning in ASV. A literature review is first made, with two training methodologies appearing evident: indirectly using a deep neural network trained for automatic speech recognition, and directly with speaker related output classes. The review finds that interest in direct training appears to be increasing, underpinned with the intent to discover new robust 'speaker embedding' representations.

Last a preliminary experiment is presented, investigating the use of a deep convolutional neural network for speaker identification. The small set of results show that the network successfully identifies two test speakers, out of 84 possible speakers enrolled. It is hoped that subsequent research might lead to new robust speaker representations or features.

Abbreviations

ASR	Automatic speech recognition
ASV	Automatic speaker verification
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DNN	Deep neural network
EM	Expectation maximisation
EER	Equal error rate
GMM	Gaussian mixture model
G-PLDA	Gaussian-probabilistic linear discriminant analysis
HT-PLDA	Heavy tailed-probabilistic linear discriminant analysis
JFA	Joint factor analysis
LDA	Linear discriminant analysis
MAP	Maximum a-posteriori
MFCC	Mel-frequency cepstral coefficients
minDCF	Minimum discrete cost function
T-Norm	Test-Normalisation
VAD	Voice activity detector
WCCN	Within class covariance normalisation
UBM	Universal background model

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Developing algorithms to automatically recognise individuals from their speech is a challenging task. Speech recordings often contain large amounts of variability, which if not accounted for can severely degrade performance. The sources of these variabilities can include speaker specific types, for example a person's emotional state, their health, and increase in age over time; transmission channel variabilities, such as coder-decoders (CODECs), and background noise; as well as other recording variabilities, such as phonetic content and duration.

A considerable amount of effort in addressing variability issues though has been made within the specific field of automatic speaker verification (ASV) [1–3]. ASV is about determining whether or not a speech recording was spoken by an enrolled speaker, compared with identification, which is about determining who specifically produced a speech signal out of a set of enrolled speakers.

Figure 1.1 illustrates how the percentage equal error rates (EER) on English telephony speech, over the period of 2004 to 2012, have decreased exponentially. The percentage EER corresponds to when the decision threshold value set, results in an equal percentage of false alarms and miss errors, and is often used as benchmark comparison score. The results presented are taken from the NIST speaker recognition evaluations (NIST-SRE) [4], which have become an almost official standard for comparing leading performances within the ASV research community.

In 2004, Figure 1.1 shows how the equal error rate (EER) performance on NIST-SRE English telephony trials corresponded to around 7.73% [5], dropping exponentially to just under 1% in 2012 [6]. The focus of the latest 2016 evaluation [7] appears to have now moved on to addressing non-English speech.

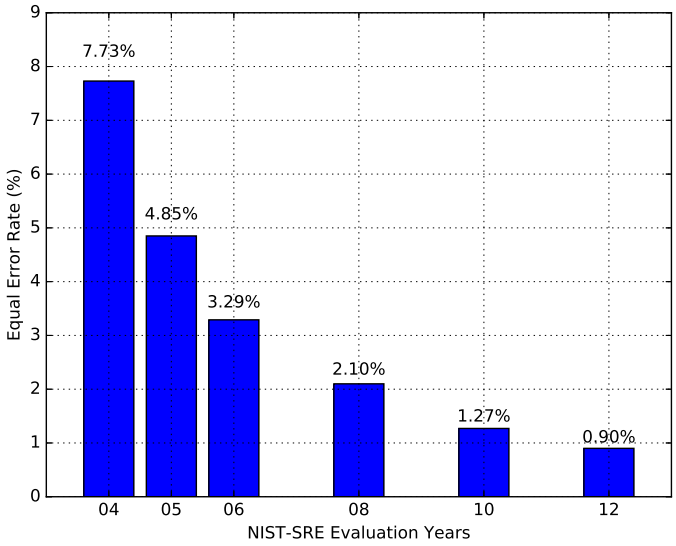


Figure 1.1: Exponentially decreasing percentage equal error rates (%EER) across the NIST-SRE trials from 2004 to 2012, with the 2004-05 %EER results taken from [5], and the remaining five years taken respectively from [6, 8–10].

Figure 1.1 shows how the performance on English telephony can be considered in many respects to be at, or if not very close, to the upper achievable limit. However it can be argued that much of the research focus from 2000 to achieve this performance, has been primarily about developing speaker models, that are capable of capturing variabilities across vast labelled corpora. There has been significantly less importance placed instead, with improving the understanding of speaker specific structures, and in particular the pre-stage feature extraction process. The majority of recognition systems still use cepstral features, which was originally proposed by Furui [11] in 1981.

This observation has as well been made recently in two eminent works. Garcia-Romero [12] for example makes this observation at the beginning of his thesis from 2012, writing that, "Recent advances in speaker recognition are not necessarily due to new or better understanding of speaker characteristics that are informative or interpretable by humans; rather, they are the result of improvements in machine learning techniques that leverage large amounts of data."

His statement has been followed by Todisco *et al* [13] at Speaker Odyssey 2016, where they wrote that, "There is more to be gained from the study of features rather than classifiers." They found that by adopting this approach with their detection of spoofing attacks work, they were able to achieve a 72% relative improvement with their newly proposed perceptually weighted cepstral coefficients. Their paper subsequently won one of the best paper awards.

To help give some impression of the exponential increase in speech training data used, the 2004 NIST evaluation contained approximately 500k gross trials. However the 2012 evaluation has since exceeded 114 million trials, equating to a 228 times increase in size. The extensive NIST speech corpora are usually then expanded even further, by taking advantage of the large amounts of available Switchboard corpora recordings [14].

Therefore despite the exceptional performance achieved with specifically English telephony, automatic speaker verification algorithms still remain somewhat susceptible to sources of variability that have not been included in the model training corpora [15], and especially when used in challenging degraded environments.

Referring again to Garcia-Romero's thesis [12], he shows how in less than benign conditions, performance degrades rapidly. Adding babble noise at 6dB signal-to-noise ratio (SNR), he found that the percentage equal error rate increased ten-fold relative to his original telephony conditions, at 1.43% to 10.7%. If automatic speaker verification algorithms therefore are to become truly robust to unwanted variabilities, then more robust features beyond cepstra are very likely needed. This theme forms the underlying motivation of the work presented.

1.1 Thesis Structure

This thesis begins by first formerly presenting the automatic speaker verification (ASV) task. Following this, a relatively in-depth review of the developments up to, and including the current

state-of-the-art is presented. ASV has seen an extensive amount of research published, the majority of which can be argued has been around developing speaker models and classification techniques, and not in the development of features. However, it is important to review the research published, in order to better understand the limits of the current techniques. The review starts with the pioneering work by Reynolds [16], on Gaussian mixture models (GMMs) with Bayesian adaptation during the mid-'90s, charting the advancements that have eventually led to the development of i-vectors (identity) by Dehak *et al* [3], in around 2010.

Following the review a series of practical tuning experiments are presented. It is found that published works can sometimes lack in specific implementation details. For example, a lot of publications will usually only write that they have taken whole evaluation data sets, such as NIST-SRE 2004 to 2006 [3], with little or no information on how they pre-screened or prepared the data, and their specific feature calculation details. The experiments presented, attempt to address some of these issues by repeating some of the leading published works [3, 12, 17], as well as hopefully providing a much deeper practical understanding of the limits of the various approaches.

Throughout the experiments presented a fixed reference test set is used. This is done to enable performances differences to be compared between earlier and later recognition approaches. These comparable set of experimental findings are hopefully of value, and contribution in particular, to ASV research community.

With the underlying motivation to discover new robust features beyond cepstra and speaker specific structures, this then motivates the initial investigation into the use of deep learning. Pioneered by LeClun *et al* [18], deep learning has led to major advancements in image object recognition [19], automatic speech recognition (ASR) [20], and machine translation [21].

Deep learning prescribes a data-driven automated methodology to discovering new features or representations from the raw data [18]. Multiple levels of representation are learnt automatically,

often using large multi-layer neural networks. The higher levels conceptually correspond to higher levels of abstraction, that are hopefully representative of the decision class, and are defined by composition of the lower level representations.

The success of deep learning within automatic speaker recognition research can be viewed as being rather limited to date, when compared to the other communities mentioned. Specifically, research investigating the direct training of deep neural networks (DNNs), where the output classes to be discriminated between are speaker related, has appeared somewhat limited until of very recent. This is most likely attributed to the often limited amounts of enrolment data available per speaker [22]. However, the direct training effectively tasks DNNs to discover new discriminative speaker features or representations.

One of the first published work that has investigated direct training, is by Variani *et al* [23] in 2014. However their work is limited to using recordings of speakers uttering the fixed phrase "Okay Google". Much of the related publications appear otherwise to either involve using a pre-trained deep neural network for automatic speech recognition [24, 25]. It is only very recently, that interest appears to be noticeably increasing, with in particular the work on "speaker embeddings" by Snyder *et al* [26].

With this in mind, a review of work investigating deep learning in the context of ASV is presented. This is followed by some experimental work, investigating the direct training of a deep convolutional neural network for speaker identification. The work presented is very much a first attempt and much further work is needed, but it is hoped that this might lead in the not too distant future to new robust speaker features, and insight. The thesis then closes with some final conclusions and recommendations.

Chapter 2

Automatic Speaker Verification

This chapter outlines briefly speaker verification from an automated research perspective.

Speaker verification is defined as the task of determining whether or not an unknown speech segment was spoken by some hypothesised speaker, and can be considered also as a speaker detection problem. Figure 2.1 shows the typical likelihood ratio implementation adopted for automated systems [1].

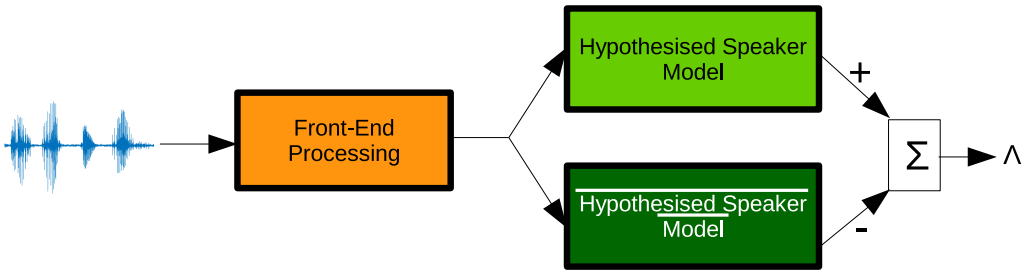


Figure 2.1: Likelihood ratio-based automatic speaker verification (ASV).

A raw speech segment Y is first passed through a front-end process to extract hopefully robust speaker dependent features. These features are then processed as an hypothesis test between:

H_0 : Y is from hypothesised speaker S,

H_1 : Y is not from hypothesised speaker S.

In order then to verify whether or not speech segment Y was uttered by speaker S, the ratio of

the two output hypothesis scores can be then taken:

$$\wedge = \frac{p(Y|H_0)}{p(Y|H_1)} \begin{cases} >= \theta & acceptH_0 \\ < \theta & rejectH_0 \end{cases} \tag{2.1}$$

where $p(Y|H_0)$ is the probability density function for the true hypothesis H_0, evaluated for observed speech segment Y. Conversely, H_1 represents the false hypothesis, that speech segment Y was not produced by the hypothesised speaker S.

If the likelihood ratio score exceeds or is equal to θ, then the true hypothesis is chosen, otherwise the false hypothesis is selected.

Often the log of the likelihood ratio is then taken [1], giving the final summation score form:

$$\wedge = \log p(X|\lambda_{hyp}) - \log p(X|\lambda_{\overline{hyp}}) \tag{2.2}$$

where $p(Y|H_0)$ in practice is represented by $p(X|\lambda_{hyp})$, with X a sequence of feature vectors $X = \{x_1, ..., x_T\}$ indexed at time $t \in [1, 2...T]$; H_0 is represented by model λ_{hyp}, and the alternate hypothesis H_1 by model $\lambda_{\overline{hyp}}$.

The notional aim of automatic speaker verification research is to then develop both, robust models to represent the two hypotheses, and feature extractors, such that decision error rate is minimised.

Chapter 3

Gaussian Mixture Models (GMMs)

Automatic speaker verification (ASV) has advanced significantly since the mid-'90s, with major advancements appearing to emerge roughly every four years. The technology applied to telephony speech is now at a point where banks, and other large global corporations are beginning to use it on a daily basis [27, 28]. It was argued at the start of this thesis, that the focus of ASV research to achieve this level of performance, has been primarily around developing models that are capable of capturing variabilities across vast labelled corpora. Probably the key early instigating work in data-driven speaker models is by Reynolds *et al* [16], with their work on adapted Gaussian mixture models (GMM) from the mid-'90s.

Reynolds *et al* [16] proposed the training of a single large GMM based universal background model (UBM), to both capture unwanted variabilities, and to help counter for the often limited amount of hypothesised speaker training data by maximum a-posteriori (MAP) adaptation. The use of a single large UBM also importantly provides a probabilistic reference, allowing speaker models to be compared. Such was the success of this formulation, it essentially still underpins the current state-of-the-art with i-vectors [2, 3]. The purpose of this chapter is therefore to review the theory and implementation of GMM-UBMs proposed by Reynolds *et al* [1, 16].

Figure 3.1 shows an expanded GMM-UBM verification system, with the non-hypothesised (a), and hypothesised (b) pre-training model stages included. The diagram illustrates how Reynolds *et al* [16] represents the non-hypothesised speaker model by a single large universal background model. The main alternative approach according to Reynolds *et al* [16] at the time, was otherwise to train a collection of individual non-hypothesised speakers, specific for each speaker.

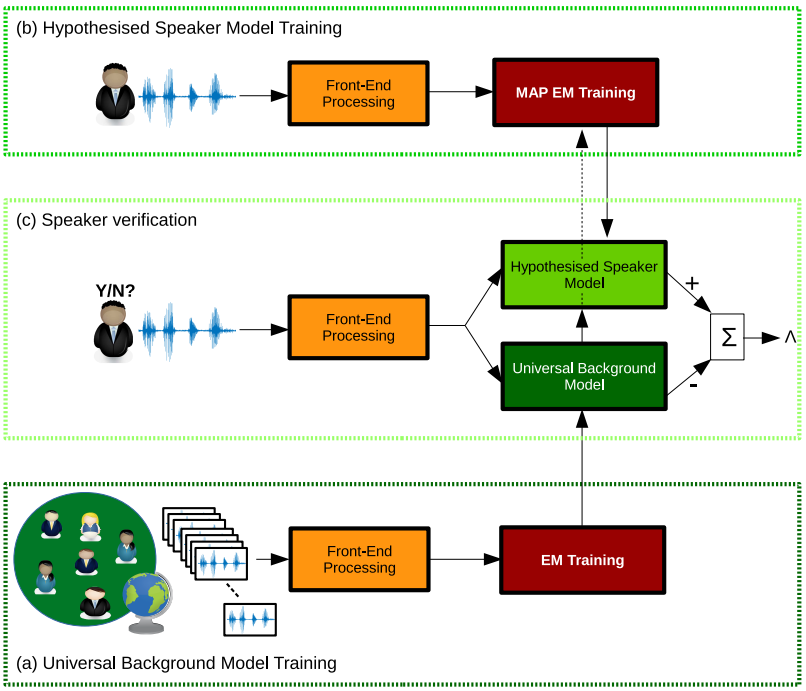


Figure 3.1: Likelihood-ratio based speaker verification system using a single reference universal background model (UBM). The first stage to training a GMM-UBM ASV, is to (a) train a UBM on a extremely last cohort of background speakers, enabling then, (b) hypothsised speaker models to be then trained using MAP adapted EM training, relative to the central reference UBM. With both the non-hypothesised UBM and hypothesised speaker models trained, likelihood-ratio based speaker verification (c) can then be applied to unknown speech recordings, with yes/no decisions made using a pre-defined score decision threshold.

However for applications requiring a large number of hypothesised speakers, preparing individual background sets for each speaker is far from practicable [29].

Figure 3.1 illustrates how the first stage to training a GMM-UBM system, is to (a) train the UBM against a large cohort of background speakers. This is implemented usually with the expectation maximisation (EM) algorithm, with the amount of background speech material used often vast. For telephony conditions, this often involves multiple NIST-SRE evaluation and Switchboard corpora [3], equating to thousands of hours of speech data.

Once a UBM has been successfully trained, it is then used in the maximum a-posteriori (MAP) adapted EM training of hypothesised speaker models (b). The final non-hypothesised UBM and the hypothesised speaker models, can then be used to perform likelihood ratio-based speaker verification on unknown speech recordings (c), as defined previous by Equation 2.1.

Interestingly, the use of a single large background model was also proposed by Carey *et al* [30] at a similar time, which they referred to as a 'General Model'. The difference with their work is that they proposed instead a text dependent hidden Markov model (HMM) speech recogniser, fed into a form of recurrent neural network to make decisions. According to Auckenthaler *et al* [31] however, subsequent work by Parris *et al* [32] investigating discrimination of phonemes for text independent verification applications, was found not to perform as well as GMM-UBM based equivalents. Auckenthaler *et al* concluded in [31] that this was most likely due to the training data not being able to be shared between the Gaussian mixture components, in comparison to GMM-UBM systems. With the HMM, the data is aligned to the phonetic classes, and consequently no data is then able to be shared between the classes, leading to poorly trained distribution model parameters.

The purpose of this chapter, is to primarily review the early pioneering works in data-driven GMM-UBMs by Reynolds *et al* [16] during the mid-'90s. This chapter is structured, presenting each of the main processing stages in order of processing, beginning with the 'Front-End Processing' on feature extraction. This is then followed with the UBM, how it is effectively a large GMM-trained using the EM algorithm, and then the MAP adapted EM training of the hypothesised speaker models. The chapter then concludes with briefly reviewing normalisation techniques commonly applied to the output log-likelihood ratio scores. Research has found that setting the verification decision threshold can be quite troublesome [29], with a number of score normalisation proposed.

3.1 Front End Processing (Feature Extraction)

Feature extraction is a critical first stage in an ASV system, transforming the raw speech signal into vectors that elicit speaker specific characteristics, as well as removing redundancy. Nearly all ASV systems proposed compute cepstral features [15, 16, 29], first proposed by Davis and Mermelstein [33] for ASR, and then applied to speaker recognition by Furui [11].

Figure 3.2 shows the typical cepstral feature extraction process.

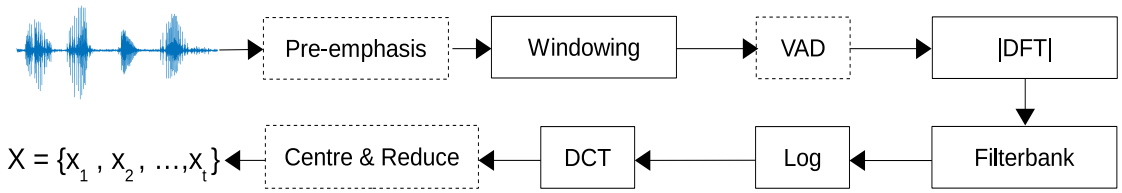


Figure 3.2: Cepstral feature extraction process.

Eight processing stages can be seen, through from the underlying audio signal to the final set of feature vectors X.

(1) Pre-emphasis: The first stage is pre-emphasis, which is the appliance of a filter to enhance the higher frequencies of the spectrum, defined in [29] as:

$$x_p(t) = x(t) - a \cdot x(t-1) \tag{3.1}$$

where a is usually between [0.95, 0.98].

Pre-emphasis is not always applied (illustrated here by the dashed line), the choice being empirically defined on a case by case basis [29]. In this work, pre-emphasis has not been explored.

(2) Windowing: Speech signals are rapidly varying time signals, and as such in the second stage, they are split into short-time frame windows [29]. A hamming or hanning window is then

applied to each window, tapering the speech signal to reduce side effects. The window size used here throughout is 32ms, intentionally chosen to be a power of two, with 256 samples at 8kHz sampling frequency for the next Fourier transform stage. The increment is 16ms giving 50% overlap.

(3) VAD: Voice activity detection is required to remove non-speech and silence time frames, that would otherwise lead to degraded speaker verification decisions. In noisy conditions, the robustness of the VAD is likely to be critical to achieving good performance, with it acting as a filter quite early into the process.

(4) |DFT|: The Fourier transform is next applied and its modulus taken, giving an estimate of the power spectrum.

(5) Filterbank: The power spectrum is then multiplied through a filterbank, which comprises a series of bandpass filters often triangular shaped, and spaced either linearly or perceptually according to the Bark/Mel scale [29].

The Mel scale is given by the equation below, and is said to reflect human perception of pitch:

$$f_{Mel} = 1000 \cdot \frac{log(1 + f_{LIN}/1000)}{log2} \tag{3.2}$$

The filterbank used here is Mel spaced with nominally 26 triangular filters, but a more appropriate number for future research on telephony band-limited speech is possibly around 20 [34].

(6) Log: The logarithm of the filterbank outputs is then taken, to better reflect human perception of loudness. It is likely also applied to compensate for the natural downwards tilt of the magnitude spectra with frequency.

(7) DCT: To then transform into the cepstral domain, the discrete cosine transform (DCT) is taken. This effectively is a decorrelation on top of the Fourier transform, and a dimension

reduction. Typically a higher order of 19 coefficients is retained [3, 12].

(7) Centre & Reduce: The cepstral coefficients are then centralised to the mean on a recording basis, which is found from a practical experience here to be critically important for achieving good performance. Early work by Reynolds in [35] showed that cepstral mean subtraction gave a 25% percentage increase in performance, removing potential convolutional noise. He also compared Rasta filtering as alternate process, finding that whilst this helped, it was no better if not worse than cepstral mean removal. The cepstra is also sometimes then "reduced" or variance normalised [29]. Cepstral mean subraction with variance normalisation (CMV) is applied here.

3.2 Dynamic Feature Extraction

In order to try and incorporate dynamic time varying information into the models, delta and double delta cepstra are normally computed. They are usually calculated as a polynomial approximation, spanning a finite duration either side to the current feature vector.

The defining equations taken from [29] are as follows

$$\begin{aligned} \frac{\delta c_f(t)}{\delta t} \approx \Delta c_f &= \frac{\sum_{k=-l}^{l} k \cdot c_{f+k}}{\sum_{k=-l}^{l} |k|} \\ \frac{\delta^2 c_f(t)}{\delta^2 t} \approx \Delta\Delta c_f &= \frac{\sum_{k=-l}^{l} k^2 \cdot c_{f+k}}{\sum_{k=-l}^{l} k^2} \end{aligned} \tag{3.3}$$

where k defines the time frame and f the cepstral feature order. Both Reynolds [16] and Garcia-Romero [12] set l as two frames either side of the current feature vector [16].

The open source Voicebox toolkit created by Brookes [36] used here to extract the cepstral features for all experiments, uses four time frames either side to compute the first order, and

one frame either side for the second order differential.

3.3 The Universal Background Model

The universal background model (UBM) is a large Gaussian mixture model, typically consisting of between 512 to 2048 mixture components, and is used to compute the required non-hypothesised speaker likelihood $p(X|\lambda_{\overline{hyp}})$ defined in Equation 2.2.

Following Reynolds *et al* [16], a GMM probability density used to estimate a likelihood function, can be defined as the weighted linear combination over M unimodal Gaussian densities

$$p(X|\lambda) = \sum_{i=1}^{M} w_i p_i(X) \tag{3.4}$$

where the M Gaussian densities $p_i(X)$ are defined as

$$p_i(X) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} exp\left\{-\frac{1}{2}(X-\mu_i)'(\Sigma_i)^{-1}(X-\mu_i)\right\} \tag{3.5}$$

with w_i representing the weights (which must satisfy the constraint $\sum_{i=1}^{M} w_i = 1$, and are strictly non-negative); μ_i, the mean, being a vector D x 1; and Σ_i, the D x D covariance matrix. The parameters of a Gaussian mixture model thus are denoted by Reynolds [16] as $\lambda = \{w_i, \mu_i, \Sigma_i\}$.

The likelihood model training criterion can then be defined as the cumulative product across all feature vector instances, $X = \{x_1, ..., x_T\}$:

$$p(X|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda) \tag{3.6}$$

where again often the logarithm is taken, giving a summation form:

$$logp(X|\lambda) = \sum_{t=1}^{T} log\, p(x_t|\lambda) \tag{3.7}$$

The UBM is trained in practice using the expectation-maximisation (EM) algorithm, which iteratively refines the GMM parameters to locally maximise the likelihood $p(X|\lambda)$.

3.4 UBM-MAP Speaker Model Training

The speaker specific models are trained via a Bayesian maximum a-posteriori (MAP) adaptation procedure, relative to the UBM. This procedure was proposed by Reynolds [16], following the original MAP procedure published by Gauvain and Lee [37], and is similar to the EM algorithm, but differs in the maximisation stage.

By defining the speaker models in reference to the UBM, helps to alleviate the small amount of training data usually only available to train the individual models. UBM's typically are trained with more than 1000 hours of audio [3,12], meaning that they are well defined. Crucially as well, the UBM in effect provides a probabilistic reference, allowing speaker models to be compared. The MAP training also helps to provide a tight coupling between the UBM and the speaker models [29], improving performance.

Figure 3.3 re-drawn from Reynolds [16], helps to illustrate this process, with a fictional two dimensional feature space. The well trained UBM model mixture components can be seen to be adapted dependent on whether there is a high count of speaker specific training data. Components *C1* and *C2* for example do have a high amount and are adapted accordingly, but *C3* does not, and is not changed.

It is useful to derive the MAP training equations with respect to the original Bayes' rule, because

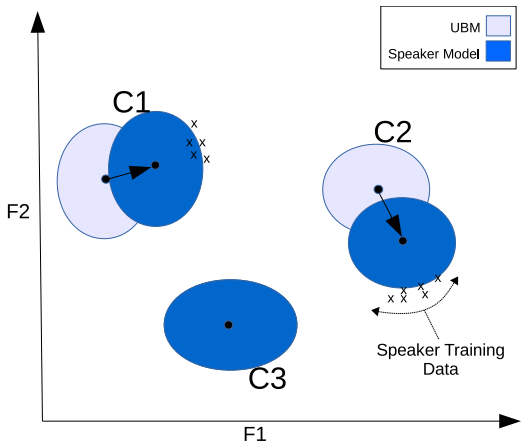


Figure 3.3: Illustration of how the speaker model Gaussian components are referenced relative to the universal background model (UBM) via maximum a-posteriori adaptation (MAP). Gaussian components $C1$ and $C2$ are opportunely adapted by the speaker enrolment features available, with $C3$ left defined by the original UBM.

it sets the ground for the current state-of-the-art i-vector [3] presented later. Taking Bayes' rule:

$$\begin{aligned} posterior &= \frac{prior \cdot likelihood}{evidence} \\ P(A|B) &= \frac{P(A) \cdot P(B|A)}{P(B)} \end{aligned} \tag{3.8}$$

To train a speaker model (λ), Reynolds [16] defines an EM style MAP modified training procedure. Taking that x_t instead now represents the training features from a hypothesised speaker, it is first aligned with the UBM mixture components:

$$p(i|x_t) = \frac{w_i p_i(x_t)}{\sum_j^M w_j p_j(x_t)} \tag{3.9}$$

where w_i represents the prior weight, for the i'th mixture component spanning 1 to M, and the likelihood probability is calculated by the Gaussian probability density function:

$$p_i(x_t) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} exp\left\{-\frac{1}{2}(x_t - \mu_i)'(\Sigma_i)^{-1}(x_t - \mu_i)\right\} \tag{3.10}$$

After the initial alignment of the observation vectors x_t with the UBM, the sufficient statistics of the the posterior distribution $p(i|x_t)$ are computed:

$$\begin{aligned} n_i &= \sum_{t=1}^{T} p(i|x_t) \\ E_i(x) &= \frac{1}{n_i} \sum_{t=1}^{T} p(i|x_t) x_t \\ E_i(x^2) &= \frac{1}{n_i} \sum_{t=1}^{T} p(i|x_t) x_t^2 \end{aligned} \tag{3.11}$$

This Reynolds notes is the same as the expectation step of the EM algorithm. However the next stage, where the model parameters are updated to maximise the likelihood is different.
Reynolds [16] adapts the old UBM model mixture component parameters with the new expectations, when computing the new model parameters:

$$\begin{aligned} \hat{w}_i &= [\alpha_i^w n_i/T + (1 - \alpha_i^w) w_i]\gamma \\ \hat{\mu}_i &= \alpha_i^m E_i(x) + (1 - \alpha_i^m)\mu_i \\ \hat{\sigma}_i^2 &= \alpha_i^v E_i(x^2) + (1 - \alpha_i^v)(\sigma_i^2 + \mu_i^2) - \hat{\mu}_i^2 \end{aligned} \tag{3.12}$$

A scale factor γ is used to ensure that all adapted mixture weights sum to unity.

The adaptation is controlled by the data-dependent mixing coefficient $\alpha_i^p, p \in \{w, m, v\}$, and is defined as:

$$\alpha_i^p = \frac{n_i}{n_i + r^p} \tag{3.13}$$

where r^p is a fixed relevance factor for parameter p. As illustrated earlier in Figure 3.3 with the conceptual two dimensional feature space, Equation 3.13 shows how mixture components with more speaker training data will have a larger α_i^p, and as such will be adapted more.

The relevant factor r^p controls how much the speaker training data adapts the mixture compo-

nents. The value used here is 19, following the Microsoft Research (MSR) Identity toolbox [38]. Reynolds *et al* [16] report that they find relevance factors in the range of 8 to 20 to be insensitive to ASV performance. They decided on a value of 16.

Further to this, according to Bimbot *et al* [29], it is found that making r model parameter dependent also brings little benefit. It is found as well that typically adapting only the means brings real benefit. This again was empirically shown by Reynolds *et al* in [16].

The MAP adaptation procedure proposed by Reynolds can be viewed intuitively, as the linear trade-off between the *prior* (initialised with the UBM), and the *likelihood* given the speaker specific training features X available:

$$p(\lambda|X) \propto \arg\max_{\lambda} \left(logp(X|\lambda) + logp(\lambda)\right) \tag{3.14}$$

The MAP concept above still in fact forms the foundation of current state-of-the-art i-vectors [2, 3], which are defined on the GMM-UBM ASV.

3.5 Score Normalisation

With the UBM and speaker models trained, GMM-UBM speaker verification can be applied. An unknown segment of speech audio can be processed against a hypothesised speaker model and the UBM, with the difference between the outputted model probability scores computed using Equation 2.2. If the score exceeds a set threshold, then the hypothesised speaker is accepted or otherwise rejected.

Unfortunately research has found setting this decision threshold to be quite troublesome [29]. The models are ultimately modelled on cepstral features, which are in themselves highly susceptible to unwanted variabilities. Some examples of these unwanted variabilities include

- speaker variabilities: mood, emotion, health, and fundamental frequency;
- transmission channel variabilities: CODECs, background noise, and handset;
- and other variabilities: duration, and phonetic content.

According to Bimbot *et al* [29], score normalisation was thus introduced to try compensate for this unwanted variability, in an attempt to make the setting of the decision threshold speaker-independent. In their journal paper, they make reference to the study by Li and Porter [39], who they write observed large variances in the target speaker score (intra-speaker scores), and impostor scores (inter-speaker scores) during trialling.

From their study, they proposed the use of the impostor score distributions to normalise the outputted speaker model scores respectively:

$$\hat{\wedge}_\lambda(X) = \frac{\wedge_\lambda(X) - \mu_\lambda}{\sigma_\lambda} \tag{3.15}$$

where μ_λ and σ_λ are the mean and standard deviation normalisation parameters for speaker model λ, derived from an impostor score distribution.

This formulation led to an extensive amount of score normalisation research, including zero score normalisation (Z-Norm) above, handset normalisation (H-Norm) by Reynolds *et al* [16], test normalisation (T-Norm) by Auckenthaler *et al* [40], cellular normalisation by Reynolds for NIST 2002 [29], and combinations.

The issue with unwanted speaker and channel variabilities is still very much a contested topic. In more recent times this has led to techniques such as probabilistic linear discriminant analysis (PLDA) [41, 42] for scoring, which effectively is a factor analysis implementation. This is discussed in the following chapter, in association with i-vectors [3], which build upon the GMM-UBM work of Reynolds [16].

Chapter 4

I-Vectors

The success of adapted Gaussian mixture models (GMMs) by Reynolds *et al* [16] highlighted the potential of data-driven approaches. Their work and others related [31, 32], spawned in effect the next decade of research into data-driven modelling techniques, culminating with i-vectors in 2010 by Dehak *et al* [3]. I-vectors have since largely remained as the state-of-the-art, with only in recent years their successful fusion with deep learnt automatic speech recognition [22, 43]. This chapter reviews i-vectors by Dehak *et al* [3], and works by Kenny *et al* [2] prior to deep learning.

I-vectors conceptually build upon the maximum a-posteriori (MAP) adaptation of hypothesised speaker models proposed by Reynolds *et al* [16], with factor analysis. The use of MAP adaptation relative to the UBM helps to mitigate the often limited amounts of hypothesised speaker training material available, by providing a form of model regularisation. The UBM also provides a probabilistic reference allowing speaker models to be better compared.

Equation 4.1 below is taken from [22, 44], and succinctly defines the i-vector, as the maximum a-posteriori (MAP) point estimate of the latent vector w. The t-th observation vector x_t is assumed to be generated by the GMM defined:

$$x_t \sim \sum_c \gamma_{ct} N\left(\mu_c + T_c w, \Sigma_c\right) \tag{4.1}$$

where c represents the Gaussian mixture component index; T_c is a matrix representing a low-rank subspace known, known as the total variability subspace from which the means of each Gaussian are adapted to a speech recording; w is a normal-distributed latent vector that is recording

specific, and is referred to as an i-vector, μ_c and Σ_c are the mean and covariance matrix of the unadapted c-th Gaussian of the speaker population (note that Σ_c can be updated [2], but in practice this supposedly found to bring little benefit [22]); and γ_{ct} represents the alignment of the observation x_t to Gaussian component c at time t, or weight, and is given by $\gamma_{ct} = p(c|x_t)$. The weights γ_{ct}, covariance matrix Σ_c, and offset mean μ_c, are derived in practice from a universal backgound model (UBM) [12, 45].

Following Garcia-Romero's description [12], Equation 4.1 illustrates how the means of the GMM are assumed to be random vectors generated by a second stage of generative modelling. A supervector is constructed, by concatenating together all the mixture component means $M = [M_1^T, ..., M_C^T]^T$, which is assumed to obey the linear model:

$$M = \mu + Tw \tag{4.2}$$

where: M is of dimension CFx1, with C number of mixture components, and feature dimension order F; and that the prior distribution of the mean supervector M is Gaussian, with mean μ and covariance TT^*.

The total variability subspace captures both the desired speaker, and undesired variabilities. The matrix T (CFxK_T) defines the mapping from the high dimensional CF supervector space, with the eigenvectors of TT^* notionally defining the K_T principle eigenvectors of the supervector covariance matrix [2]. The dimension of the total variability factors w or i-vector is K_Tx1.

Prior to i-vectors, Kenny *et al* [46] proposed the Joint Factor Analysis (JFA) model, where the factor analysis model of the means of each mixture component is expanded to:

$$M = \mu + Ux + Vy + Dz \tag{4.3}$$

with: U (CFxK_U) and V (CFxK_V) matrices of low-rank, representing the speaker and channel

variability subspaces respectively; x (K_U x 1) and y (K_Vx1) are normal-distributed independent latent vectors representing the speaker and channel dependent factors; and D is a diagonal CFxCF matrix representing residual variability not captured by the two other components, with z (CFx1) representing the respective normal-distributed latent residual-factors.

JFA approximately halved the percentage equal error rate of GMM-UBM verification based systems, from notionally 8% down to 4% [17]. Dehak *et al* [3] states that the original motivation to investigate i-vectors, utilising a single total variability subspace, was because he found that the channel factors (x) of the JFA model, in fact contained speaker information. During JFA scoring, Dehak *et al* [3] describes how the likelihood of a test utterance feature vector is computed against a channel-compensated speaker model ($M-Ux$), implying thus a potential loss of speaker discriminative information.

It should also be notably mentioned, that performances similar to JFA were also being achieved using nuisance attribute projection (NAP) [47]. The key concept described in [48] behind NAP, is to remove dimensions from a support vector machine (SVM) expansion, which are irrelevant to the speaker recognition task. NAP was developed independently in parallel to JFA.

The intention of this chapter is to review the theory and implementation behind i-vectors. As such, the remainder of this chapter is structured, with first a review of the training process to estimate the total variability matrix (T-matrix), before then summarising the key T-matrix training steps and extraction of i-vectors from speech utterances. The chapter then concludes by reviewing verification procedures, and in particular probabilistic linear discriminant analysis (PLDA) [41] scoring, which has been key to achieving state-of-the-art performance.

4.1 Training the T-Matrix

In this sub-section, the total variability matrix (T-matrix) training procedure is presented.

The T-matrix is trained iteratively, using typically the expectation maximisation (EM) algorithm. Dehak *et al* [3] writes that the same procedure used to estimate the eigenvoice matrix V in [2], is used. Equation 4.4 is taken from [2], and defines the eigenvoice matrix V within the familiar mean supervector maximum a-posteriori (MAP) model, for speaker s; and the unadapted speaker population offset μ, usually defined by a universal background model (UBM).

$$M(s) = \mu + Vy(s) \tag{4.4}$$

The one significant difference with the eigenvoice matrix V Dehak *et al* [3] writes, is that all the training recordings from the same speaker are considered as belonging to the same speaker. However when training the T-matrix, they pretend that all the training recordings from the same speaker are derived from different speakers. The objective is to estimate the total variability subspace, independent of speakers and channels. For further reference, a useful practical implementation guide is also given by Lei in [45].

Thus following the eigenvoice matrix V training by Kenny *et al* [2], and substituting V for T; they proposed the likelihood objective function to maximise, similar to Reynolds [16], as

$$\prod_{s} P_{T,\Sigma}\left(X(s)\right) \tag{4.5}$$

where $P_{T,\Sigma}\left(X(s)\right)$ is the probability of the training observation feature vector $X(s)$, for speaker s, given the GMM model $\lambda(s)$ corresponding to the supervector containing $Tw(s)$ with unadapted covariance matrix Σ (note again that Σ can be updated [2], but in practice this is supposedly found to bring little benefit [22]). The probability is then extended across all speakers s by the

taking the product.

The training procedure proposed by Kenny *et al* [2], can be split into three stages. The first stage is a preparation stage, whilst the second and third correspond to the iterative expectation maximisation (EM) algorithm:

(1) Prepare the required zero'th, first, and second order statistics expanded into matrices.

(2) The expectation step of the EM algorithm: the posterior distribution of the latent i-vector $w(s)$ (equivalent to $y(s)$ with eigenvoices) is calculated, using the current alignment of speaker s's feature vector, the current estimate of T, and the prior $N(y|0, I)$.

(3) The maximisation step: update T by a linear regression, using the new posterior distribution estimates of $w(s)$.

Fundamentally the training procedure can be viewed as fitting feature observations to the Gaussian component densities, such that the conditional likelihood criterion previous (Eq. 4.5) is maximised. Thus following Kenny *et al*'s [2] observations, and taking the standard GMM model equation to be maximised

$$\sum_s \sum_c \left(N_c(s) log \frac{1}{(2\pi)^{F/2} |\Sigma_c|^{1/2}} - \frac{1}{2} \sum_t \left(x_t - M_c(s)\right)^* \Sigma_c^{-1} \left(x_t - M_c(s)\right) \right) \tag{4.6}$$

where s ranges over all speakers in the training set, c spans all mixture components, and t is summed over all time frames x_t aligned with c for speaker s; if the covariance matrix Σ is already estimated, then the problem can be seen to reduce to a least mean squares minimisation exercise of fitting Gaussian densities

$$\sum_t \left(X_t - M_c(s)\right)^* \left(X_t - M_c(s)\right) \tag{4.7}$$

where Σ_c^{-1} can be dropped.

Unfortunately fitting the Gaussian densities is complicated because the true Gaussian state structure for each speaker s is hidden, and the GMM model parameters are not fully sufficient to explain $\lambda(s)$ relative to the training observation vectors $X(s)$.

More practically explained, the Gaussian component densities will have regions of overlap, leading to regions of uncertainty as to how to best fit the training observations, and therefore giving rise to the presence of latent parameters. This is also further complicated by the use of MAP adaptation, requiring a maximum criterion trade-off between the original model prior parameters values $p(\lambda(s))$, and new training data likelihood updates $p(X(s)|\lambda(s))$ [16, 37].

To address these difficulties, Kenny *et al* [2] follow Reynolds *et al*'s [16] and Gauvain and Lee's [37] approach, by utilising the EM algorithm to train the T matrix. The EM algorithm is used to iteratively estimate T, such that the likelihood criterion $\prod_s P_{T,\Sigma}(X(s))$ accrued across all training speakers s is maximised.

Kenny *et al* [2] propose repeatedly estimating in turn the expectation of the latent parameters (i-vectors $w(s)$) whilst holding all the model parameters constant, and then subsequently feeding the new latent parameter estimates back in to update the model parameter estimates (the T matrix). The EM process is repeated until $\prod_s P_{T,\Sigma}(X(s))$ converges at a maximum value.

The EM training of the T-matrix is next presented in depth, following the eigenvoice V matrix training procedure in [2], whilst also remembering that all training recordings are treated as if they were all produced by different speakers instead.

4.1.1 Stage 1 - Prepare Required Statistics

For each training speaker recording, the zero'th, first and second order statistics are required:

$$\begin{aligned} N_c(s) &= \sum_t p(c|x_t) \\ F_c(s) &= \sum_t (x_t - \mu_c) \\ S_c(s) &= \sum_t (x_t - \mu_c)(x_t - \mu_c)^* \end{aligned} \tag{4.8}$$

where c is the mixture component, s is a speaker recording, t is the time frame, and μ is the speaker independent mean offset, typically defined by a pre-trained UBM.

Expanding into matrices:

$$NN(s) = \begin{pmatrix} N_1(s) * I & & \\ & \ddots & \\ & & N_C(s) * I \end{pmatrix} \quad FF(s) = \begin{pmatrix} F_1(s) \\ \vdots \\ F_C(s) \end{pmatrix} \quad SS(s) = \begin{pmatrix} S_1(s) & & \\ & \ddots & \\ & & S_C(s) \end{pmatrix}$$

where $NN(s)$ is the zero'th order diagonal matrix, of dimension CDxCD, with C mixture components and D feature dimension order; $FF(s)$ represents the first order matrix of length CD, centralised to the speaker independent mean offset (UBM), and I is the identity matrix; and $SS(s)$ represents the second order diagonal matrix of dimension CDxCD.

4.1.2 Stage 2 - Calculate Posterior Expectation of $w(s)$ (EM Algorithm)

The posterior distribution of the latent vector $w(s)$, is calculated using the current alignment of the speaker s's training feature data X, the current estimates of T, and the prior $N(y|0,1)$. On the first iteration T is randomly initialised.

Kenny *et al* [2] derive the required expectation of $w(s)$ from Bayes' posterior equation below (dropping the reference to s):

$$P_T(w|X) \propto P_T(X|w)N(w|0,I) \tag{4.9}$$

or in notation used so far:

$$p(w|X,T) \propto p(X|w,T)p(w) \tag{4.10}$$

where $p(w|X,T)$ represents the current estimate of posterior distribution of w to be calculated, and T the model parameters to be estimated.

Kenny *et al* gives a full derivation in Appendix-*Proof of Proposition 1* of [2], but essentially solves Eq. 4.9, by presuming the distribution form:

$$p(w|X(s),T) \propto exp\left(-\frac{1}{2}(w-a(s))^* l(s)(w-a(s))\right) \tag{4.11}$$

where $a(s)$ represents the required posterior mean expectation of the latent vector $w(s)$, which they define as:

$$a(s) = l^{-1}(s)T^*\Sigma^{-1}FF_x(s) \tag{4.12}$$

and $l(s)$ is the covariance matrix of the posterior of $w(s)$, defined as:

$$l(s) = I + T^*\Sigma^{-1}NN(s)T \tag{4.13}$$

Kenny *et al* derive $a(s)$ by essentially beginning from Equation 4.9, solving for:

$$P_T(w|X) \propto P_T(X|w)N(w|0, I) \tag{4.14}$$

where the reference to s was dropped for ease of notation.

They then define $P_T(X|w)$ in Appendix-*Lemma 1* as:

$$log\, P_{\boldsymbol{T},\Sigma}\left(X(s)|w(s)\right) = G_\Sigma(s) + H_{\boldsymbol{T},\Sigma}\left(s, \boldsymbol{w(s)}\right) \tag{4.15}$$

with $G_{\Sigma(s)}$ representing a Gaussian log likelihood function given by expression:

$$log\, p(x|s) = G_\Sigma(s) = \sum_{c=1}^{C}\left(\frac{N_c(s)}{TF} log \frac{1}{(2\pi)^{D/2}|\Sigma_c|^{1/2}} - \frac{1}{2} tr\left(\Sigma_c^{-1} S_c(s)\right)\right) \tag{4.16}$$

where TF represents the total time frames, and $H_{\boldsymbol{T},\Sigma}(s, \boldsymbol{w})$ containing the required hidden w terms is defined as:

$$H_{\boldsymbol{T},\Sigma}(s, \boldsymbol{w}) = \boldsymbol{w^* T^* \Sigma^{-1} F F}(s) - \frac{1}{2}\boldsymbol{w^* T^* N N(s) \Sigma^{-1} T w} \tag{4.17}$$

The proof given to derive $H_{\boldsymbol{T},\Sigma}(s, \boldsymbol{w})$ involves multiplying out the exponent terms from log likelihood distribution $p(x|w(s))$. First they define some useful terms, and again drop the reference to s. They let $\boldsymbol{O} = \boldsymbol{T}\boldsymbol{w}$, where O_c denotes the c'th block of O, with each block a Dx1 vector in size (D is the feature dimension order), and define the second order statistic centred additionally by O_c:

$$S_c(O_c) = \sum_t (x_t - \mu_c - O_c)(x_t - \mu_c - O_c)^* \tag{4.18}$$

The log likelihood distribution $p(x|w)$ can then be defined using the two newly defined terms as:

$$p(x|w) = \sum_c \left(N_c \frac{1}{(2\pi)^{D/2}|\Sigma_c|^{1/2}} - \frac{1}{2} tr\left(\Sigma_c^{-1} S_c(O_c)\right) \right) \tag{4.19}$$

To derive $H_{\boldsymbol{T},\Sigma}(s, \boldsymbol{w})$, Kenny *et al* expand out the exponent term as follows (remembering the second order statistic $S_c(s)$ from Equations 4.8 defined earlier, and N_c accounting for missing required summation over time for $(O_c O_c^*)$:

$$S_c(O_c) = S_c - F_c O_c^* - O_c F_c^* + N_c O_c O_c^* \tag{4.20}$$

re-including the trace

$$tr\left(\Sigma_c^{-1} S_c(O_c)\right) = tr\left(\Sigma_c^{-1} S_c\right) - 2F_c \Sigma_c^{-1} O_c + O_c^* \Sigma_c^{-1} N_c O_c \tag{4.21}$$

where the trace of $\Sigma_c^{-1} S_c$ is only required because the matrices in practice are diagonal; the two $F_c O_c^*$ and $O_c F_c^*$ terms can be grouped because they are equal scalars when multiplied out; and $O_c^* \Sigma_c^{-1} N_c O_c$ comes from the transpose trace property $tr(ABB^*) = tr(B^* AB)$.

Summing across all Gaussian mixture components leads to the final proof of $H_{\boldsymbol{T},\Sigma}(s,\boldsymbol{w})$, with $\sum_c tr\left(\Sigma_c^{-1}S_c\right)$ corresponding to $G_\Sigma(s)$:

$$\sum_c tr\left(\Sigma_c^1 S_c(O_c)\right) = \sum_c tr\left(\Sigma_c^{-1}S_c\right) - \boldsymbol{2O^*\Sigma^{-1}FF} + \boldsymbol{O^*NN\Sigma^{-1}O} \tag{4.22}$$

Having derived $H_{\boldsymbol{T},\Sigma}(s,\boldsymbol{w})$ containing the pertinent latent vector w terms, Kenny *et al* substitutes back into their required Bayesian posterior distribution definition for w (dropping the reference to s):

$$\begin{aligned}
P_T(w|X) &\propto P_T(X|w)N(w|0,I) \\
&\propto exp\left(\boldsymbol{w^*T^*\Sigma^{-1}FF} - \frac{1}{2}\boldsymbol{w^*T^*NN\Sigma^{-1}Tw} - \frac{1}{2}w^*w\right) \\
&\propto exp\left(w^*T^*\Sigma^{-1}F - \left(\frac{1}{2}w^*\left[T^*N\Sigma^{-1}T + I\right]w\right)\right) \\
&= exp\left(w^*T^*\Sigma^{-1}F - \frac{1}{2}w^*lw\right) \\
&\propto exp\left(-\frac{1}{2}(w-a)^*l(w-a)\right)
\end{aligned}$$

By assuming that the posterior distribution $p_T(w|X)$ must be of a Gaussian form, they solve to find the required posterior mean value a, given covariance matrix l^{-1}:

$$a(s) = l^{-1}(s)T^*\Sigma^{-1}FF(s) \tag{4.23}$$

The original mean MAP adapted i-vector equation (Eq. 4.2), can thus be expanded to its full form:

$$\begin{aligned}
M(s) &= Tw(s) + m \\
\hat{M(s)} &= Tl^{-1}(s)T^*\Sigma^{-1}FF_X(s) + m
\end{aligned} \tag{4.24}$$

4.1.3 Stage 3 - Update T via Maximisation (EM Algorithm)

The T matrix is next updated by a linear regression procedure, where the hidden variable $w(s)$ now plays the role of an explanatory variable. Kenny *et al* gives a very lengthy derivation in the Appendix of [2], under *Proof of Proposition 3*.

Kenny *et al* otherwise defines the new estimates of T (and Σ), as the solution of:

$$\sum_s (NN(s)TE[w(s)w^*(s)]) = \sum_s (FF(s)E[w^*(s)]) \tag{4.25}$$

where s represents each training speaker (remembering for T-matrices, different recordings from the same speaker are treated as if they were produced by a different speaker); and $E[w(s)]$, and $E[w(s)w^*(s)]$ are the first and second order moments of $w(s)$, calculated from the new posterior distribution expectation estimates.

To solve Equation 4.25, Kenny *et al* first note that $NN(s)$ is diagonal, and equate the ith row of the left hand side to the ith row of the right, for $i = 1, ..., CD$ (where C is the total number of components, and D the feature dimension order). This gives

$$T^i \sum_s \left(NN^i(s)E[w(s)w^*(s)]\right) = \sum_s \left(FF^i(s)E[w^*(s)]\right) \tag{4.26}$$

where T^i is the ith row of T, and similarly for $NN^i(s)$ and $FF^i(s)$.

They then go further, by making the observation that the zero'th order (diagonal) matrix $NN(s)$ values are the same, for each respective Gaussian mixture component. Letting i then be of the form $(c-1)D + d$, where $1 \leq c \leq C$ and $1 \leq d \leq D$, then $NN^i(s) = N_c(s)$, gives

$$T^i \sum_s (NN_c(s)E[w(s)w^*(s)]) = \sum_s \left(FF^i(s)E[w^*(s)]\right) \tag{4.27}$$

T is then solution of a (K_TxK_T) system of equations, where each row T^i is calculated independently in turn, and NN_c is effectively a scalar value for each Gaussian mixture component. First re-arranging for clarity, and noting that the bracketed left terms multiply out to a scalar allowing

$$\left[\sum_s (NN_c(s)E[w(s)w^*(s)])\right] T^i = \sum_s \left(FF^i(s)E[w^*(s)]\right) \tag{4.28}$$

The solution for T then is

$$T^i = \left(\sum_s (NN_c(s)E[w(s)w^*(s)])\right)^{-1} * \left(\sum_s \left(FF^i(s)E[w^*(s)]\right)\right) \tag{4.29}$$

where $E[w^*(s)] = \left(l^{-1}(s)T^*\Sigma^{-1}FF_X(s)\right)^*$, and $E[w(s)w^*(s)] = l^{-1}(s)$, which can be calculated from the new posterior distribution estimates from the expectation (EM) stage.

The proof of the system equation 4.25 is defined in the Appendix of [2] under *Proof of Proposition 3*, and involves constructing an EM auxiliary function, with w as the hidden variable. The theoretical construct involving Jensen's inequality is first reviewed, before then following the derivation given Kenny *et al* [2].

Following the EM tutorials by Andrzejewski [49], Mackey [50], and Ng [51], the objective is to maximise the likelihood of the observed data x given model parameters λ. Letting w represent a latent variable, then the conditional log likelihood can be defined as (dropping the reference to s):

$$log\left(p(X|\lambda)\right) = log\left(\sum_{w} p(X, w|\lambda)\right) \tag{4.30}$$

Defining then some auxiliary distribution $q(w)$ for w, this leads to

$$log\left(p(X|\lambda)\right) = log\left(\sum_{w} q(w)\frac{p(X, w|\lambda)}{q(w)}\right) \tag{4.31}$$

A lower likelihood is then formalised using Jensen's inequality for a concave function, given that it is not possible to directly maximise $p(X|\lambda)$ due to the latent model variable w

$$f(E[X]) \geq E[f(X)] \tag{4.32}$$

which then substituting Equation 4.31 into Jensen's inequality

$$\begin{aligned} log(p(X|\lambda)) &= log\left(\sum_{w} q(w)\frac{p(X, w|\lambda)}{q(w)}\right) \\ &\geq E_{w\sim q}\left[log\left(\frac{p(X, w|\lambda)}{q(w)}\right)\right] \\ &\geq \sum_{w} q(w) log\left[\frac{p(X, w|\lambda)}{q(w)}\right] \\ &\geq \sum_{w} q(w) log\left(p(X, w|\lambda)\right) - \sum_{w} q(w) log\left(q(w)\right) \end{aligned}$$

Thus the following observation can be drawn

$$log(p(X|\lambda)) \geq \underbrace{E_{w\sim q}\left[log\left(p(X,w|\lambda)\right)\right]}_{Expected\,Complete\,Log-Likelihood} + \underbrace{E_{w\sim q}\left[log\left(q(w)\right)\right]}_{Entropy\,H(w)} = A(w,\lambda) \tag{4.33}$$

where the lower defined limit comprises of the expected complete log-likelihood, with respect to w drawn from distribution q; and the entropy or uncertainty $H(w)$ of latent variable w. This lower bound is typically referred to as an auxiliary function, defined here as $A(w, \lambda)$.

The EM algorithm in its general form is precisely then coordinate ascent on the two variables w and λ, of the log-likelihood lower bound $A(w, \lambda)$ [50]

(1) **E-Step**: $q^{(i+1)}(w) = argmax_q A(w, \lambda^i)$

(2) **M-Step**: $\lambda^{i+1} = argmax_\lambda A(w^{(i+1)}, \lambda))$

where i represents the iteration number.

The M-Step is thus the maximisation of the auxiliary function $A(w, \lambda)$, defined by Equation 4.33 with respect to parameters λ. This maximisation is equivalent to maximising the expected complete log-likelihood (ECLL) under q, provided the entropy or uncertainty $H(w)$ can be minimised.

The expectation E-Step is used to derive an optimal value for $q(w)$, which the solution is

$$q^{i+1}(w) = p(w|X, \lambda^i) \tag{4.34}$$

This solution for $q(w)$ can be derived as follows

$$\begin{aligned} log(p(X|\lambda)) &\geq E_{w\sim q}\left[log\left(\frac{p(X,w|\lambda)}{q(w)}\right)\right] \\ &\geq E_{w\sim q}\left[log\left(\frac{p(w|X,\lambda)p(X|\lambda)}{q(w)}\right)\right] \\ &\geq log\left(p(X|\lambda)\right) - E_{w\sim q}\left[log\left(\frac{q(w)}{p(w|X,\lambda}\right)\right] \\ &\geq log\left(p(X|\lambda)\right) - KL\left(\frac{p(w|X,\lambda)}{q(w)}\right) \\ &\geq log\left(p(X|\lambda)\right) - KL\left(\frac{p(w|X,\lambda)}{p(w|X,\lambda)}\right) \\ &\geq log\left(p(X|\lambda)\right) \end{aligned} \tag{4.35}$$

The solution then for $q(w)$ involves minimising the Kullback-Leibler (KL) divergence, by setting $q(w) = p(w|X,\lambda)$. Substituting this solution for $q(w)$ leads to inequality becoming an equality. Given that the log-likelihood provides an upper limit for the auxiliary function for all $q(w)$, the equality result indicates that choice of $q(w)$ must be maximal.

Finally returning to Kenny *et al*'s derivation, they first begin by constructing the required Jensen's inequality for the EM algorithm. Referring to the earlier stated definition equations

$$\begin{aligned} f(E[x]) &\geq E[f(x)] \\ &\geq \sum_w q(w)^{(i+1)} log\left(\frac{p(w|\lambda^{(i)},X)}{q^{(i+1)}(w)}\right) \end{aligned}$$

and substituting in the required terms gives

$$f(E[X]) \geq \sum_s \int \left(log\frac{P_{T,\Sigma}(X(s),w))}{P_{T_0,\Sigma_0}(w|X(s))}\right) P_{T_0,\Sigma_0}(w|X(s))dw \tag{4.36}$$

where T_0 and Σ_0 represent the currents estimate for T and Σ.

Kenny *et al* then from this define their required auxiliary function to be maximised, with respect to T.

$$A = \sum_{s} E_{w\sim q}\left[logP_{T,\Sigma}(X(s)|w(s))\right] \tag{4.37}$$

The derivation of this can be explained by the KL equality derivation, Equation 4.35 previous, resolving to posterior probably of observation vectors X given model λ. This can be re-written in the earlier proof definition format

$$A = \sum_{s} E_{w\sim q}\left[logP\left(X(s)|\lambda(s) = \{w(s), T, \Sigma\}\right)\right]$$

Kenny *et al* then refer back to their previous general definition for the $logP_{T,\Sigma}(X|w(s))$ (in Lemma 1 of the Appendix), displayed previously here in Equation 4.15 as

$$A = \sum_{s} G_{\Sigma}(s) + \sum_{s} E_{w\sim q}\left[H_{T,\Sigma}(s, w(s))\right] \tag{4.38}$$

They first term does not contain any reference to T, and so will be disappear when differentiated with respect to T. Kenny *et al* thus focus on the second term defined as

$$\sum_{s} E_{w\sim q}\left[H_{T,\Sigma}(s, w(s))\right] = \sum_{s} E_{w\sim q}\left[w^*(s)T^*\Sigma^{-1}FF(s) - \frac{1}{2}w^*(s)T^*NN(s)\Sigma^{-1}Tw(s)\right] \tag{4.39}$$

resolving to a more suitable form for matrix differentiation, beginning with the second term

$$= \sum_{s}\left(E_{w\sim q}[w^*(s)]T^*\Sigma^{-1}FF(s)\right) - \left(\frac{1}{2}tr\left(T^*NN(s)\Sigma^{-1}TE_{w\sim q}[w(s)w^*(s)]\right)\right) \tag{4.40}$$

since $\frac{1}{2}w^*(s)Bw(s) = \frac{1}{2}tr(w^*(s)Bw(s))$ holds true because the end result is a scalar, where B represents here $T^*NN(s)\Sigma^{-1}T$, and $w(s)$ importantly is a vector. This is then further translated to $\frac{1}{2}tr(w^*(s)Bw(s)) = \frac{1}{2}tr([Bw(s)]w^*(s))$, since the trace is invariant with the order of summa-

tion (*property:* $tr(XY) = tr(YX)$, where X is a n x m matrix, and Y is a m x n matrix). Kenny *et al* then re-write the expectation within the second term, since both the trace and expectation are linear operators.

Next they re-write the first term to required differential form, whilst also factorising out the expectation and Σ, and changing slightly the ordering of the second term (again the *property:* $tr(XY) = tr(YX)$), giving the required form for differentiation

$$= \sum_s tr\left(\Sigma^{-1}\left(FF(s)E_{w\sim q}[w^*(s)]T^* - \frac{1}{2}NN(s)TE_{w\sim q}[w(s)w^*(s)]T^*\right)\right) \tag{4.41}$$

where the first term follows from

$$\begin{aligned} E_{w\sim q}\left[w^*(s)T^*\Sigma^{-1}FF(s)\right] &= tr\left(E_{w\sim q}\left[w^*(s)T^*\Sigma^{-1}FF(s)\right]\right) \\ &= tr\left(\left[E[w^*(s)]T^*\right]\left[\Sigma^{-1}FF(s)\right]\right) \\ &= tr\left(\left[\Sigma^{-1}FF(s)\right]\left[E_{w\sim q}[w^*(s)]T^*\right]\right) \\ &= tr\left(\Sigma^{-1}\left(FF(s)E_{w\sim q}[w^*(s)]T^*\right)\right) \end{aligned}$$

Kenny then differentiates Equation 4.41 with respect to T, setting the gradient to 0, to obtain the required system of normal equations (Equation 4.25 previous), which can be solved for the update of T.

Taking the matrix differentiation identities

$$\frac{\delta B\theta^*}{\delta\theta} = B \qquad\qquad \frac{\delta\theta B\theta^*}{\delta\theta} = \theta(B + B^*) \tag{4.42}$$

where B is a matrix that is not a function of variable θ.

Differentiating Equation 4.41 using these two identities gives,

$$= \sum_s tr\left(\Sigma^{-1}\left(FF(s)E_{w\sim q}[w^*(s)] - NN(s)TE_{w\sim q}[w(s)w^*(s)]\right)\right) \tag{4.43}$$

Kenny *et al* then write that if Y represents some matrix with the same dimensions as T, that the derivative with respect to T in the direction of Y is given by

$$\begin{aligned} &= \sum_s tr\left(\Sigma^{-1}\left(FF(s)E_{w\sim q}[w^*(s)] - NN(s)TE_{w\sim q}[w(s)w^*(s)]\right)Y^*\right) \\ &= \sum_s tr\left(M(s)Y^*\right) \end{aligned}$$

where $M(s)$ effectively represents a transformation matrix.

In order then for the gradient to be equal to 0 for all Y, then the following is a must

$$\sum_s \Sigma^{-1}\left(FF(s)E_{w\sim q}[w^*(s)] - NN(s)TE_{w\sim q}[w(s)w^*(s)]\right) = 0 \tag{4.44}$$

from which finally Equation 4.25 is derived. This by default implies that the matrix trace summation is redundant and can be dropped, since critically the gradient must be 0 for all Y, and not as the result of the summation.

4.2 T-Matrix Training Overview and I-Vector Extraction

In the previous section, the training of the T-matrix was explained in great detail, following the detailed Eigenvoice work by Kenny *et al* in [2]. In this section, the key steps are highlighted for ease of reference, and the i-vector extraction process, which can be easily defined out of this process. A useful implementation guide is also given by Lei in [45], from which this section is based.

(1) Calculate the required zero'th, first and second order statistics for each speaker (s), and Gaussian mixture component (c), where the sum extends over all t observation frames X_t respectively for each speaker s.

$$N_c(s) = \sum_t p(c|X_t)$$
$$\hat{F}_c(s) = \sum_t p(c|X_t)X_t$$
$$\hat{S}_c(s) = diag\left(\sum_t p(c|X_t)X_t^2\right)$$

(2) Centre the first and second order statistics relative to a pre-trained UBM, with mean μ_c.

$$F_c(s) = \hat{F}_c(s) - N_c(s)\mu_c$$
$$S_c(s) = \hat{S}_c(s) - diag\left(\hat{F}_c(s)\mu_c^* + \mu_c\hat{F}_c(s)^* - N_c(s)\mu_c\mu_c^*\right)$$

(3) Expand the statistics into matrices for computational ease.

$$NN(s) = \begin{pmatrix} N_1(s) * I & & \\ & \ddots & \\ & & N_C(s) * I \end{pmatrix} \quad FF(s) = \begin{pmatrix} F_1(s) \\ \vdots \\ F_C(s) \end{pmatrix} \quad SS(s) = \begin{pmatrix} S_1(s) & & \\ & \ddots & \\ & & S_C(s) \end{pmatrix}$$

where $NN(s)$ is the zero'th order diagonal matrix of dimension CDxCD, with C mixture components and feature dimension order D, and I is the identity matrix; $FF(s)$ represents the first order matrix of length CD, centralised to a speaker independent UBM mean offset; and $SS(s)$ represents the second order diagonal matrix of dimension CDxCD.

(4) Calculate the posterior expectation of the latent total variability factors $w(s)$, or as commonly referred to, i-vectors. This is the expectation stage of the EM training algorithm.

$$\begin{aligned} l(s) &= I + T^*(s)\Sigma^{-1}NN(s)T \\ w(s) &\sim Normal\left(l^{-1}(s)T^*\Sigma^{-1}FF(s), l^{-1}(s)\right) \\ \bar{w}(s) &= E\left[w(s)\right] = l^{-1}(s)T^*\Sigma^{-1}FF(s) \end{aligned} \tag{4.45}$$

where $l^{-1}(s)$ is the R_TxR_T covariance matrix of the posterior distribution of $w(s)$, Σ is the unadapted speaker independent covariance matrix derived from a UBM, $Normal$ implies a Gaussian normal distribution, and $\bar{w}(s)$ is the required expected mean value (or i-vector). On the first iteration T can be random initialised [45].

(5) Pre-calculate required accumulators.

$$\mathfrak{A}_{\mathfrak{c}} = \sum_{s} NN_c(s)E\left[w(s)w^*(s)\right] = \sum_{s} NN_c(s)l^{-1}(s)$$
$$\mathfrak{C}_{\mathfrak{c}} = \sum_{s} FF_c(s)E\left[w^*(s)\right] = \sum_{s} FF_c(s)\left(l^{-1}(s)T^*\Sigma^{-1}FF(s)\right)^*$$

where $E\left[w(s)w^*(s)\right]$ is given by $l^{-1}(s)$; and $E\left[w^*(s)\right]$ is the complex conjugate transpose of the newly calculated mean, of the posterior distribution of $w(s)$.

(6) Compute the new estimate of T, the total variability matrix. This is the maximisation stage of the EM algorithm.

$$T_c = \mathfrak{A}_{\mathfrak{c}}^{-1}\mathfrak{C}_{\mathfrak{c}}$$

(7) Repeat stages 4 to 6 approximately 20 iterations according to Lei [45], to give the final estimate for T.

(8) To calculate i-vectors during normal speaker enrolment, Equation 4.45 in stage 4 should be used.

4.3 Speaker Verification Scoring and PLDA

In the previous two sections, the training of the total variability matrix (T) was extensively reviewed, concluding with a summary implementation guide based on a publication by Lei [45]. The T-matrix defines a total variability subspace, that attempts to capture both the desired speaker, and undesired channel variabilities. Once trained, the T-matrix allows speech utterances to be translated into this low-dimensional subspace. The computed total variability factors are thus MAP point estimates of latent variables ($w(s)$ for speaker s) relative to a well defined UBM, more commonly referred to as i-vectors .

The training of the T-matrix is computationally intensive, involving the iterative application of the EM algorithm against a large cohort of background speakers. The maximum likelihood estimation process proposed by Kenny [2] is motivated by the often limited amounts of training data involved per speaker, making it difficult to estimate the supervector covariance matrix reliably. The covariance matrix of the supervectors (corresponding to TT^*) is therefore estimated by maximising across all the background training speakers.

Kenny further explains this in [2], by making such observations when reviewing the earlier work by Gales [52], that T (effectively V in [2]) should be the dependent regressed coefficients, and the i-vectors $w(s)$ ($y(s)$ in [2]) the explanatory variables. Gales [52] instead maximises with respect to $w(s)$, which Kenny objects to because T then cannot be reliably estimated if there is limited speaker training data per speaker. In addition, the use of factor analysis for dimension reduction also critically provides a common shared subspace T to compare speaker utterances.

In this section, probabilistic linear discriminant analysis (PLDA) [42] verification scoring of the i-vectors within the total variability sub-space is reviewed. PLDA has enabled state-of-the-art speaker verification performances [12] to be achieved. Initial formulations otherwise combined within class covariance normalisation (WCCN) [3, 12] with linear discriminant analysis (LDA) to compensate for undesired variabilities. The verification score was then calculated by way of

a cosine similarity applied between the compensated i-vectors.

4.3.1 Probabilistic Linear Discriminant Analysis (PLDA)

PLDA was originally proposed by Prince and Elder [42] for facial recognition, before then being adopted by the speaker recognition research community [10,41]. PLDA in the context of speaker recognition, ignores the process by which i-vectors are generated, and considers instead that they are generated by some probabilistic generative model [12].

Given then an i-vector $w_{i,r}$ corresponding to some utterance r spoken by speaker i, PLDA proposes the following model, taken from [12, 42]

$$w_{i,r} = m + \Phi\beta_i + \Gamma\alpha_{i,r} + \epsilon_{i,r} \tag{4.46}$$

where $S_i = m + \Phi\beta_i$ represents the between-speaker variability, which depends only the identity of the speaker and not the particular image (no dependence on r), with a speaker independent offset m; and the undesired channel component $C_{i,r} = \Gamma\alpha_{i,r} + \epsilon_{i,r}$, which is utterance r dependent, and describes the within-speaker variability (channel noise).

The PLDA factor analysis equation 4.46 therefore can be seen to strongly resembles Joint Factor Analysis (JFA) [46]. In fact Kenny defines in [41] PLDA as a special case of JFA that was independently developed, where PLDA limits to a single Gaussian component unlike JFA.

The columns of Φ define a basis for the between-speaker subspace (eigenvoices), with β a latent identity vector having a standard normal normal distribution, representing the position in the subspace. Similarly, Γ defines a basis for the within-speaker (eigenchannel) subspace, with α again a latent identity vector having a standard normal distribution, and ϵ represents a residual term, defined in [42] as Gaussian distributed with zero mean, and diagonal covariance Σ.

Within the field of speaker recognition Kenny [41] modifies the PLDA Equation 4.46, proposing

that the within-speaker factor term ($\Gamma\alpha_r$) can be removed to give the following form

$$w_{i,r} = m + \Phi\beta_i + \epsilon_{i,r} \tag{4.47}$$

Kenny writes that because the dimensionality of i-vectors is sufficiently low, then the covariance matrix Σ of the residual channel noise term $\epsilon_{i,r}$ can be robustly estimated, and made full covariance. He therefore proposes that $\Gamma\alpha_{i,r}$ can be removed because it does not then add any additional information, particularly in the case of telephony speech, where there is a large amount of labelled background training data available. The model parameters m, Φ, Σ are learnt during a pre-training phase similar to the training of the T-matrix, using the familiar EM algorithm, estimating their maximum likelihood point estimates [42].

PLDA verification scoring considers whether a claimed identity utterance, and a test utterance are from the same speaker or not. This resolves into a log-likelihood ratio test between the hypothesis H_s that they are from the same speaker, or the alternate hypothesis H_d that they are from different speakers.

Following the derivation in [12], if a test utterance is produced by the claimed identity speaker (hypothesis H_s), then the following generative PLDA model is assumed

$$\begin{bmatrix} w_{i,1} \\ \vdots \\ w_{i,K} \\ w_T \end{bmatrix} = \begin{bmatrix} m \\ \vdots \\ m \\ m \end{bmatrix} + \begin{bmatrix} \Phi \\ \vdots \\ \Phi \\ \Phi \end{bmatrix} \beta_i + \begin{bmatrix} \epsilon_{i,1} \\ \vdots \\ \epsilon_{i,K} \\ \epsilon_T \end{bmatrix} \tag{4.48}$$

which can be summarised as

$$n^{'} = m^{'} + \Phi^{'}\beta_i + \epsilon^{'} \tag{4.49}$$

where there are K potential i-vectors $w_{i,r}$ for claimed speaker identity i, with $r = 1, ..., K$; and a test utterance i-vector w_T.

Conversely, if the test utterance is produced by a different speaker (hypothesis H_d), then the i-vectors are assumed to follow

$$\begin{bmatrix} w_{i,1} \\ \vdots \\ w_{i,K} \\ w_T \end{bmatrix} = \begin{bmatrix} m \\ \vdots \\ m \\ m \end{bmatrix} + \begin{bmatrix} \Phi & 0 \\ \vdots & \vdots \\ \Phi & 0 \\ 0 & \Phi \end{bmatrix} \begin{bmatrix} \beta_i \\ \beta_j \end{bmatrix} + \begin{bmatrix} \epsilon_{i,1} \\ \vdots \\ \epsilon_{i,K} \\ \epsilon_T \end{bmatrix} \tag{4.50}$$

where $i \neq j$, and summarised as

$$n^{'} = m^{'} + \Phi^{''}\beta^{''} + \epsilon^{''} \tag{4.51}$$

Equation 4.48 illustrates how the i-vectors are all generated using the same latent identity variable β_i, compared with $\beta^{''} \in [\beta_i, \beta_j]^T$ in Equation 4.51.

Given then that the channel residual and latent identity variables are assumed to be Gaussian, along with the statistical independence assumption between the speaker and channel components, Garcia-Romero [12] defines the verification score as the ratio of two Gaussian distributions

$$score = log\, p\left(\{n_{i,r}\}, n_T|H_s\}\right) - log\, p\left(\{n_{i,r}\}, n_T|H_d\right) \tag{4.52}$$

$$= logN\left(w^{'}|\{m^{'}, \Phi^{'}\Phi^{'T} + \Sigma^{'}\}\right) - logN\left(w^{'}|\{m^{'}, \Phi^{''}\Phi^{''T} + \Sigma^{'}\}\right) \tag{4.53}$$

where $\Sigma^{'}$ is a block diagonal matrix with $K+1$ copies of the channel noise covariance matrix Σ in the diagonal.

Equation 4.53 illustrates how the log-likelihood ratio therefore involves two Gaussian distributions with the same mean, but two different covariance matrices.

Critically also, the computed verification score is not based on point estimates of the latent identity variables, but by the marginalisation over the latent variables. Prince and Elder [42] write that they recognise the inherent uncertainty, given that the claim identity might be observed under noisy conditions. An interesting side effect of the marginalisation according to Prince and Elder, is that it is then also valid to compare models with different numbers of identity variables, or i-vectors in this case, as illustrated by Equations 4.48 and 4.50. Otherwise specifically then, they calculate the probability that the i-vectors representing the claimed identity, and the test i-vector, are all generated from the same person or not, irrespective of who the actual identity is.

4.3.2 The Statistical Independence Assumption

PLDA assumes (i) statistical independence between the between-speaker (S) and channel (C) components, and (ii) that S and C have Gaussian distributions, which are both questionable assumptions. Kenny in [41] highlights for example, that it is well known that gender-dependent

eigenchannel modelling is more effective than gender-independent modelling, thereby illustrating the limitations of the statistical independence assumption. However despite the relationship between speaker and channel effects yet to be properly understood, the application of PLDA within the low dimensional total variability space, is said to aid such statistical independence assumptions.

By limiting the dimension order of the between-speaker covariance matrix $Cov(S, S)$, and the within-speaker (channel) covariance matrix $Cov(C_r, C_r)$, Kenny [41] writes that they can be treated as full rank. Remembering that the total variability matrix notionally corresponds to the principle eigenvectors of the supervector covariance matrix, this then presumably aids the further statistical independent decomposition into S and C, enabling them to be then assumed to be full rank due to the low dimensionality.

Interestingly, Kenny continues in [41] by hypothesising that there might in fact be principle axes of channel variability, which are speaker dependent. Observing scatter plots produced by Tang *et al* [53] of effectively the first two i-vector components, the plots are high directional with respect to the speakers. Kenny therefore concludes that this must not only explain the success of cosine scoring, but also that the channel effects are speaker dependent. This also then leads him to conclude that the statistical independence assumption is fundamentally flawed.

Whilst the conclusion drawn by Kenny around the success of cosine scoring appears to be valid, his following hypothesis that the statistical independence assumption is fundamentally flawed is possibly over-stated. The channel effects in the utterance recordings used by Tang *et al* [53] might not be as dominant as the speaker effects, potentially masking the nature of the channel effects. Tang *et al* [53] explains the highly directional speaker dependent nature of their scatter plots, being attributed simply to the intentional MAP adaptation applied. The precise recordings used by Tang *et al* [53] are not listed, but it would seem likely they are derived from the GALE Mandarin broadcast news database used in their experiments.

It would seem therefore, despite the extensive research into data-driven modelling techniques, much further research is needed to understand the limits of the statistical independence assumption [41], between the speaker and channel effects. This problem according to Kenny is likely to be difficult to solve, if not otherwise potentially highly non-linear if there are multiple different compounding sources of channel noise.

4.3.3 Gaussian Assumptions: Heavy-Tailed PLDA

Accepting then the difficulties with solving the statistical independence assumption, Kenny instead focused his efforts on addressing the Gaussian assumption in [41], proposing the use of the heavy tailed Students' t-distribution. His findings, on the NIST 2008 SRE evaluation data, demonstrated a substantial 30% relative improvement over Joint Factor Analysis. Inspired by this, Garcia-Romero [12] subsequently developed an i-vector length normalisation procedure, addressing computational issues around Kenny's work, whilst achieving similar performance.

Kenny describes how the Gaussian assumption effectively prohibits large deviations from the mean, but outlier speaker and channel case effects do arise. He gives such speaker examples as non-native language speakers, and channel effects arising from gross distortions, particularly in the case of non-telephony microphone sourced speech.

Motivated by these observations, Kenny in [41] proposes that the latent identity prior β_i, and the residual channel noise $\epsilon_{i,r}$ in the PLDA model equation (Eq. 4.46), are Student's t distributed rather than Gaussian. He refers to his proposed form of PLDA as Heavy-Tailed PLDA (HT-PLDA).

Importantly, the weight of the tails of the Student's t distribution are controlled by the degrees of freedom. The smaller the number of degrees of freedom, the heavier the tails. Conversely, as the number of degrees of freedom is increased to infinity, then the Student's t distribution

converges to a Gaussian distribution. Both the latent identity prior β_i, and the residual $\epsilon_{i,r}$, have individual respective degrees of freedom parameters, which must be estimated.

Unfortunately however, the use of the heavy-tailed distribution results in the log-likelihood ratio for speaker verification scoring (Eq. 4.53), to no longer have a closed form solution according to Kenny [41]. Calculating the two required likelihoods involves marginalising (i.e. the evidence in Bayesian theory) over the respective latent variables.

Kenny states in [41], that this tends to be intractable, because it generally difficult to factorise out the latent variable priors from the joint distribution (over the claimed identity, and test utterance i-vectors). This suggests that the distribution of the latent variable priors over the i-vector must be complicated, making it difficult to separate out the latent priors. To solve this problem, Kenny in [41] uses variational Bayes to compute a lower bound, as a proxy for the marginal likelihoods in the log-likelihood ratio (Eq. 4.53), at expense to the amount of computation required.

4.3.4 Gaussian Assumptions: I-Vector Length Normalisation

The use of variatonal Bayes requires much more computation according to Garcia-Romero [12], when compared with the original Gaussian PLDA. In an effort to rectify this, Garcia-Romero in [12] investigates the use of non-linear transforms, to map from the heavy tailed student's t distribution, back to the preferred, computationally efficient use of Gaussian distributions.

In [12], Garcia-Romero proposes dropping the statistical independence assumption between the speaker identity,and channel variations. Based on Kenny's observations in [41], that there must exist a principle axis of channel variation that is speaker dependent, he proposes the following generative i-vector model

$$w = m + \Omega z \tag{4.54}$$

where the latent variable z instead represents both the speaker and channel factors, and follows the Student's t distribution. Observing then that the Student's t distribution belongs to the family of Elliptically Symmetric Densities (ESD), he investigates the use of a non-linear transformation technique called Radial Gaussianization (RG), proposed by Lyu and Simoncelli in [54].

Radial Gaussianization can be used to transform non-Gaussian ESD into spherically symmetric Gaussian distributions, rendering appropriate the Gaussian and statistical independence assumptions [12]. Lyu and Simoncelli point out in [54] that conventional linear transforms, like for example the second-order decorrelation method PCA-reliant on covariance, have no effect on removing the dependencies beyond second order for ESD. Therefore instead, observing that ESD are inherently symmetric, they propose a non-linear function that acts radially, mapping to a Gaussian distribution, and thus preserving spherical symmetry.

RG is a two-step process. The first, is to transform the ESD to a Spherically Symmetric Density (SSD)-removing the second-order dependencies, by removing the mean, and applying a linear whitening transform learned from i-vector data samples. In the second stage, the centred and whitened i-vectors, are then length normalised, by applying a non-linear histogram warping of the length distribution.

Following [54], the *radial* marginal distribution (lengths) of the *whitened* source i-vector ($r = \|w_{wht}\|$), in terms of the ESD generating function $f(\cdot)$, follows by standard form a Chi distribution

$$p_r(r) = \frac{r^{d-1}}{\beta} f\left(-r^2/2\right) \tag{4.55}$$

where β is a normalising constant that ensures that the density integrates to one, and d is the number of degrees of freedom in r.

The *radial* marginal distribution of a spherical Gaussian density, with *unit* component variance,

similarly follows a Chi distribution, with d degrees of freedom

$$p_x - \frac{r^{d-1}}{2^{d/2-1}\Gamma\left(d/2\right)} exp\left(-r^2/2\right) \tag{4.56}$$

where $\Gamma\left(\cdot\right)$ is the standard Gamma function.

The RG transform is thus specifically defined as

$$g(r) = F_x^{-1} F_r\left(r\right) \tag{4.57}$$

where F_x^{-1} is the normalising inverse cumulative density function (CDF) of p_x, multiplied by the cumulative density function of p_r, where $r = \|w_{wht}\|$.

Garcia-Romero [12] points out that the CDF F_r, in practice has to be estimated from the data, which poses problems for when competing in the NIST-SRE academic trials. The rules of the evaluation only allow each trial verification score to be derived from the two i-vectors specified, where RG requires training across a large portion of the evaluation set to be robust. It does seem strange however, why previous trial data is not representative, given that PLDA inherently needs representative multiple speaker session data. It may perhaps be that RG is highly susceptible to speaker and channel variations.

Irrespectively, Garcia-Romero continue nevertheless, simplifying the second stage length normalisation, by proposing to simply scale the lengths of each centred and whitened i-vector, to unit length. Thus the RG transform is modified to

$$g_{LN}(r) = \frac{r}{\|r\|} \tag{4.58}$$

where $r = \|w_{wht}\|$. This effectively projects r onto the unit hypersphere.

He justifies his approximation in [12], hypothesising that most of the probability mass of a

standard Gaussian distribution will be concentrated within a thin shell located close to its mean. If the dimension order is increased, the thickness of the shell will as a consequence decrease. They argue therefore that, projecting the data onto a hyper-sphere, whose radius depends on the dimensionality of the space (from the earlier relation of the Gaussian distribution to the Chi distribution in Equation 4.55), is therefore sufficient. They expect this approximation to become more exact as the dimensionality of the space then increases.

Garcia-Romero in [12] also writes on a practicality note, that the precise choice of radius for the hyper-sphere is unimportant, and that the unit radius is simply chosen for convenience. Using a different radius can be seen to just introduce an offset in the verification score, based on Equation 4.53, which is dependent effectively on comparisons between covariance matrices.

He summarises the i-vector length normalisation transformation in [12], learned from background development data as:

(1) Centre and whiten
- Compute mean and sample covariance matrix ($\hat{m}$, $\hat{S}$) from development data.
- Use singular value decomposition to obtain whitening transform $A = D^{1/2}U^T$, where $\hat{S} = UDU^T$.
- Centre and whiten i-vector: $w_{wht} = A(w - \hat{m})$.

(2) Scaling or normalisation
- Project onto unit sphere: $w_{LN} = \frac{w_{wht}}{\|w_{wht}\|}$.

The transformation can then be applied during verification, once learned from background development i-vector data.

Garcia-Romero and Wilson [10] find that using this heavy-tailed Student's t to Gaussian transformation, they are able to achieve similar state-of-the-art performances to Kenny's HT-PLDA

if not better, on NIST-SRE 2010 evaluation data.

Table 4.1 is taken from [10], and shows applying i-vector length normalisation (G-PLDA), they are able to achieve 1.29% equal error rate (EER), compared with HT-PLDA at 1.48% for male speech. They find a similar improvement over HT-PLDA for female speech, with the percentage EER improving from 2.21% to 1.97%. Interestingly, they also find that applying length normalisation to the HT-PLDA, improves the performance to slightly better than that of G-PLDA + length normalisation, at 1.28% EER for male and 1.95% EER for female.

Rank		Male		Female	
		EER%	minDCF(new)	EER%	minDCF(new)
4	G-PLDA	3.08	0.4193	3.41	0.4008
3	HT-PLDA	1.48	0.3357	2.21	0.341
2	G-PLDA + Length	1.29	0.3084	1.97	0.3511
1	HT-PLDA + Length	1.28	0.3036	1.95	0.3297

Table 4.1: I-vector compensated results from Garcia-Romero and Espy-Wilson [10] comparing Gaussian (G) and heavy-tailed (HT) PLDA, with length normalisation on the NIST-10 telephony core 5 condition.

4.3.5 PLDA Conclusions

PLDA was originally proposed by Prince and Elder [42] for facial recognition, before being adopted for speaker recognition by Kenny [41]. Like JFA, PLDA assumes that the speaker and channel effects are statistically independent, and that they are Gaussian distributed. Kenny questions the validity of both assumptions, but without an obvious solution to the statistical independence assumption, proceeds to otherwise investigate the use of the heavy-tailed Student's t distribution.

Kenny's original motivations for proposing the Student's t distribution, were to allow for outlier speaker and channel effects, which are effectively prohibited by the use of Gaussian distributions.

Within the domain of telephony speech, he shows that HT-PLDA leads to a substantial 30% relative improvement compared with JFA [41], and similarly if not more when compared with the original Gaussian-PLDA (G-PLDA) form [10]. However the use of the heavy tailed Student's t distribution comes at the increased expense in computation, requiring the use of variational Bayes.

Garcia-Romero [12] in response to this proposes an i-vector length normalisation procedure, based on a non-linear transformation procedure [54]. Taking advantage of the fact that the Student's t distribution is an elliptical symmetric distribution, the non-linear transform radially learns a mapping to the computationally efficient Gaussian distributions, whilst attempting to preserve spherical symmetry. With Wilson in [10], they show that similar, if not better state-of-the-art of the performances can be achieved on telephony speech, using length normalisation.

The success of these two procedures raises the question though, of what are exactly these outlier effects attributed to principally, and are they that common? It would seem for English telephony, which comprises most of the NIST-SRE 2010 data [55], the margins for further improvement are small. The application of PLDA is perhaps then no more then than fine tuning in this scenario, but perhaps PLDA or JFA may prove useful in detecting or understanding speech variations with non-native speakers.

However for interview or microphone derived speech, Kenny [41] finds that HT-PLDA modelling of channel effects, if left to its own devices degenerates. He concludes that there must be extreme non-Gaussian effects present. An alternative observation, is perhaps simply that the underlying UBM (which provides probabilistic speaker independent reference), must be derived with a sufficient amount of background interview speech data. Attempting otherwise to apply HT-PLDA channel corrections, to what is maybe a fundamentally telephony based recogniser, would seem possibly then futile. Kenny unfortunately does not state what he used, other than he used a large corpus of background data.

An interesting area for further research thus, is maybe to try to investigate the statistical independence assumptions between speaker and channel effects, but under strict controlled recording conditions. Is it for example possible, to interpret the i-vector changes when the microphone, vocabulary, and speakers recordings are made under strict controls? Kenny's findings in [41] are derived from open NIST data, where such control is not possible, making his strong conclusions then that the statistical independence assumption are flawed, potentially over-stated. He also does not consider the impact of any speech detection, and speaker segmentation.

It also may simply not be possible to easily understand the relationship, which might be highly non-linear. The use of highly non-linear deep neural networks may then be another interesting area of research. An initial investigation into the use of deep networks for speaker identification is covered later in this thesis.

Chapter 5

Performance Tuning Experiments

In the previous two chapters, an extensive review of the major advancements in automatic speaker verification (ASV) was presented. The review covered in particular the pioneering works by Reynolds *et al* [1, 16] on GMM-UBMs from the mid-'90s, which led eventually to the development of i-vectors by Dehak *et al* [3] in 2010. Current state-of-the-art ASV systems are still essentially underpinned by i-vectors, with only in recent times seen their partial fusion with deep learnt automatic speech recognition (ASR) [22, 56].

In this chapter a series of practical performance tuning experiments are presented. Often many publications leave out specific implementation details, making it difficult to sometimes repeat if not understand the practical limits of published findings. Dehak *et al* in [3] for example, writes only that they used whole NIST-SRE 2004 to 2006 data sets for when training their UBM and T-matrix, with few details on any pre-screening and on their feature calculation process. Bimbot *et al* [29] describe their feature extraction process in detail as part of their literature review, but they do not advocate the importance of cepstral mean subtraction (CMS). This is shown by Reynolds in [35] to be critical to achieving good performance. The cepstral features can also be variance normalised (CMV), or reduced.

The experiments presented therefore attempt to answer basic questions such as:

- How many EM training iterations should be generally used when training a UBM?
- At what stage in the cepstral feature calculation process should the speech detector and CMV be applied?

- What is the effect of adding more training data to the UBM on performance?
- What is the effect of increasing the size of the UBM on GMM-UBM and i-vector recognition performance?

This chapter is structured as follows, beginning with first a description of the experimental corpora used, and the scoring procedure adopted. Following this, an architectural type description is given, explaining how each of the components in the ASV system have been implemented. The experiments then presented are effectively split into two parts, with the first part investigating different GMM-UBM configurations and hyperparameter settings, and the latter exploring T-matrix training with i-vectors.

In order to allow for comparison between different speaker verification configurations, a fixed reference test set is used throughout. It is hoped that this comparable set of findings, exploring GMM-UBMs and i-vector based verification systems, provides useful information to the research community. The chapter then ends with some conclusions.

5.1 Experimental Corpora and Toolboxes

The speech corpora used throughout the experimental work presented here, are taken from the standard NIST-SRE 2004 to 2005 [57, 58] evaluations, and the Switchboard-2 Phase I and II sets [14, 59].

Table 5.1 shows statistics of the corpora used for training the UBMs and T-matrices, whilst 5.2 shows the test set used throughout for consistency. For reference, the total number of hours is also included. Only male speech is used throughout the experiments presented, to try to avoid additional issues found with gender dependencies [41].

All audio is 8kHz, 8-bit μ-law American English telephony. The term '3conv4w' means that three conversations are used to train each respective speaker model, each approximately 5 minutes gross audio duration, and four wire (speakers are channel separated). The term '1conv4w' refers to the test list, defining the list of 5 minute conversational segments to be verified against a specified model respectively.

Corpora	#Channels	Overall Total	Total Male	Cumulative Male Total
NIST-SRE04 Training	1	440	181	181
SWB2-P1	2	606	267	448
SWB2-P2	2	745	364	812
FE-P1	2	1948	880	1692
FE-P2	2	1966	846	2538

Table 5.1: List of corpora used for training the UBMs and T-matrices in hours (SWB2-P1=Switchboard 2-Phase 1, FE-P1=Fisher English-Part 1).

Corpora	Trial Set	#Channels	Total Hours	Total Male	#Male Trials
NIST-SRE 05	3conv4w-1conv4w	2	184	84	12161

Table 5.2: The reference test set used, which is selected out of the NIST-SRE 2005 evaluation standard trial conditions, where '3conv4w' is the model training list, and '1conv4w' the test list.

5.2 Accuracy Scoring

The standard NIST-SRE 2005 [58] scoring procedure is used in line with the '3conv4w-1conv4w' trial condition, as referred to in Table 5.2. Accuracy scores are computed in the form equal error (EER) percentage rates and a cost score, with results often presented in the form of detection error trade-off (DET) plots [58].

An example DET curve is shown in Figure 5.1, with an EER score corresponding to 7.6%, and cost 0.033. The performance profile is derived by progressively increasing or decreasing the detection threshold applied to the output log-likelihood ratio score, defined earlier in Equation

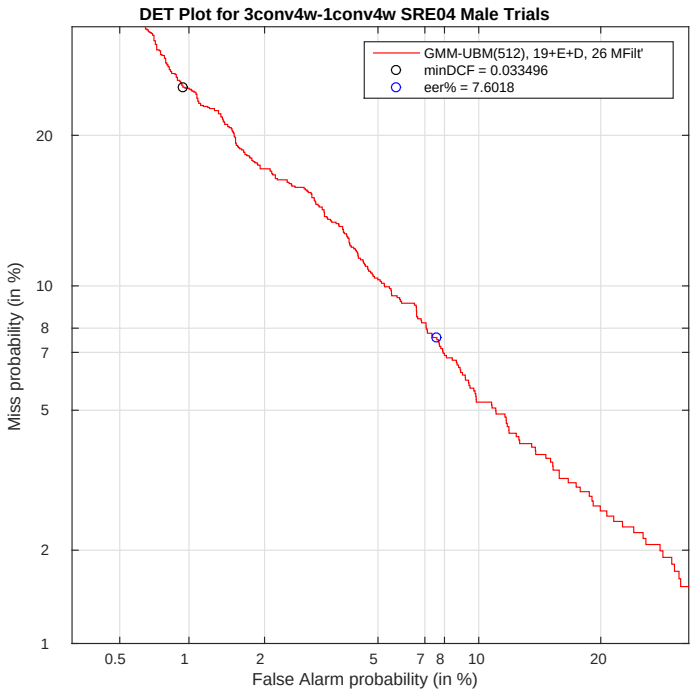


Figure 5.1: Example detection error trade-off (DET) curve.

2.1, but re-stated again here for convenience

$$\wedge = log\left(\frac{p(Y|H_0)}{p(Y|H_1)}\right)\begin{cases} >= \theta & accept H_0 \\ < \theta & reject H_0 \end{cases} \tag{5.1}$$

where $p(Y|H_0)$ is the probability density function for the true hypothesised speaker H_0, evaluated for observed speech segment Y, and respectively H_1 is for Y not being the hypothesised speaker, with θ the decision threshold applied.

The cost score is defined by NIST [58] as:

$$C_{Det} = \left(C_{Miss} \cdot P_{Miss|H_0} \cdot P_{H_0}\right) + \left(C_{FA} \cdot P_{FA|H_1} \cdot (1 - P_{H_0})\right) \quad (5.2)$$

where C_{Miss} and C_{FA} are the relative miss and false alarm costs, $P_{Miss|H_0}$ and $P_{FA|H_1}$ are the a-posteriori probabilities of a miss given the hypothesised speaker being present, and false alarm given it is not the hypothesised speaker present, and P_{H_0} the a-priori probability of the hypothesised speaker.
The three parameters are defined as follows by NIST:

C_{Miss}	C_{FA}	P_{H_0}
10	1	0.01

Table 5.3: Cost scoring equation parameters as defined by NIST-SRE in 2005 [58].

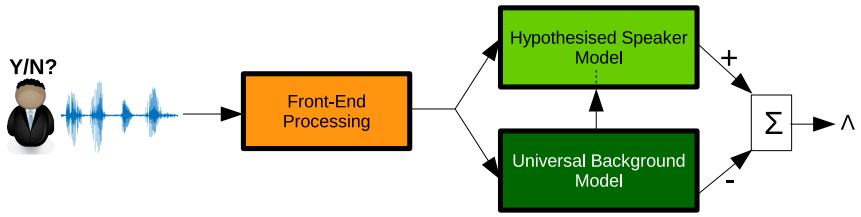
The cost equation effectively biases the optimum score threshold to be much higher than one would normally expect. Whilst the cost of every miss is high at '10', the prior probability of the hypothesised speaker being present is set very low at 1%. This places the operating point on the DET curve in the top left of the plot, as can be seen in Figure 5.1, at over 20% miss and just under 1% false alarm. Normally one would expect the minimum cost point to fall in the bottom right of the DET plot, corresponding with a low miss probability.

This effectively shows how NIST is prioritising applications that have to process a significant amount of telephony audio, with a very low probability of their target speaker, where potential excessive false alarms are distracting to an operator.

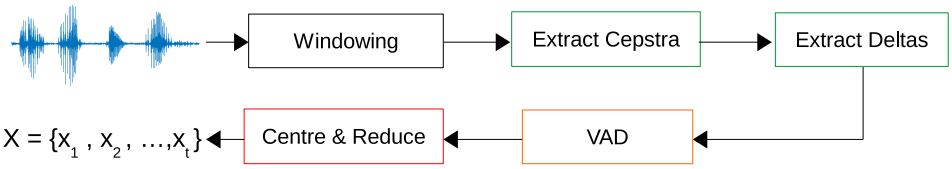
A number of slight modifications have since been made by NIST since 2005, but Equation 5.2 represents the underlying vanilla cost formulation, and the three cost equation parameters listed in Table 5.3 are used throughout for all experiments presented.

5.3 ASV Hyperparameter Settings and the Voice Activity Detector

Figure 5.2(a) highlights the main components found in an ASV system, taken from [16]. Directly underneath in (b) is an expanded diagram of the sub-components typically found in the 'Front-End Processing' component. As described in [16], the objective of the 'Front-End Processing' is to attempt to extract features from the speech signal that convey speaker dependent information. The output of this process is usually a sequence of feature vectors $X = [x_1, ...x_T]$ representing the speech utterance, where x_t is a feature vector at short-time frame or segment t. The front-end processor usually also contains additional processes, including a speech or voice activity detector (VAD) to attempt to remove non-speech periods, and further filter type processes to try to remove nuisance variabilities.



(a) Likelihood-ratio based speaker verification set up.



(b) Front-End Processor.

Figure 5.2: Main automatic speaker verification processing stages.

Figure 5.2(b) highlights the standard *default* configuration used throughout the experiments, with the exception of the feature extraction experiment presented in Sub-Section 5.4.2. The specific cepstral feature extraction hyperparameter settings chosen are listed in Table 5.4, and

computed using the open source Voicebox toolkit written by Brookes [36] in Matlab.

Component	*Hyperparameter*	*Value*
Windowing	Frame Size	32ms
	Frame Increment	16ms
Cepstra Extractor	Filterbank	26 triangular mel-spaced from 0 to 4kHz
	Cepstra Order	19 coefficients (+ Δ + $\Delta\Delta$ + energies as specified)

Table 5.4: Feature extraction hyperparameter settings.

For the speaker processing stages highlighted in Figure 5.2(a), namely the model training and verification decision process, the open source Microsoft Research (MSR) Identity Toolbox [38] written in Matlab was used. The MSR toolbox is found to be very efficient, with the implemented code highly parallelised.

A successful attempt was made at implementing the training procedure for the T-matrix, but it was found to be too slow to be of practical use in the first instance. The training of the T-matrix takes a considerable amount of computational processing, due to the large amounts of background data often used for robustness [3].

Unfortunately both the Voicebox [36] and MSR Identity toolboxes [38] do not include a voice activity detector (VAD), which is important for achieving good speaker verification performance. A VAD was implemented, effectively following the similar speaker verification process depicted in Figure 5.2(a), but the decision output changed to be 'speech' or 'not speech' instead.

The GMM model training libraries without MAP adaptation from the MSR Identity toolbox [38] were used. Two GMM models were effectively trained, representing respectively the 'speech' and 'non-speech' utterance periods. The models were trained on all the NIST-SRE 2004 [57] male training data. The supplementary 'ctm' transcript files were used as the ground-truth voice activity labels. The cepstral features were extracted similarly using the Voicebox toolkit [36]. Table 5.5 lists the specific feature extraction and model hyperparameter settings.

In degraded and more variable conditions, the robustness of the VAD is likely to prove more

Hyperparameter	Values
Windowing	32ms frames, 16ms increment
Filterbank	26 triangular mel-spaced from 0 to 4kHz
Cepstral order	19 coefficients + Δ + energies (=40 in total)
Gaussian Components	2

Table 5.5: Voice activity detector feature extraction and model hyperparameter settings.

critical. Fauve *et al* for example, from an ealier publication in 2007, used only the energy distribution in [17], following earlier work from [60]. In a more recent publication however, Novotnỳ *et al* [61] use a Czech phoneme recogniser, dropping all frames that are decoded as silence or noise. For noise robustness, they purposely trained their phoneme recogniser on speech that they artificially degraded with additive noise.

To also avoid complications with developing a robust speaker segmentation, all audio data is limited to two channel recordings. The focus of the experiments presented here is specifically on understanding in the first instance, the robustness of GMM-UBM and i-vectors based speaker verification. Interestingly, speaker segmentation and diarization is a theme of current research, with the recent 'Speakers in The Wild' challenge [62], where proposed techniques are based on combined type clustering with factor analysis [61].

Last, the use of score normalisation is also not investigated here. The use of Z-Norm' and T-Norm' [29] invariably should lead to improved performance. The objective of this experiments here is to first understand the fundamental GMM-UBM verification performance, prior to the additional normalisation stages.

5.4 GMM-UBM Experiments

Three experiments are presented investigating hyperparameter sensitivity of the GMM-UBM verification system, and the effect on performance due to when specifically the VAD, and the

cepstral mean variance (CMV) normalisation are applied within the front-end feature extraction process. The hyperparameters investigated specifically, includes the number of EM UBM training iterations required to train a UBM, and second the joint performance with respect to the use of acceleration features, number of Gaussian components, and amount of UBM speech training data.

5.4.1 EM UBM Training Iterations

The purpose of this experiment, is simply to understand the number of EM iterations required to successfully train a UBM. A 512 Gaussian component UBM is trained on the NIST-SRE 2004 male 'conv4w' speech training data, with EM iterations up to a maximum of 40. The cepstral feature order nominally used is 40, consisting of 19 cepstra, the delta-cepstra, and both energies.

Figure 5.3 shows the result, with the number of EM iterations with respect to the UBM log-likelihood training criterion $log\,(p(X|\lambda_{UBM}))$ (Eq. 3.6), where $X = x_t, ...x_T$ represents feature observation vectors at short-time frame t, and λ the UBM.

The plot shows that 20 EM iterations appears generally sufficient for training the UBM, with the log-likelihood probability effectively converged on a maximum value. It should be noted that the MSR Identity Toolbox [38] uses a binary split to initialise the Gaussian components.

5.4.2 Application of the VAD and CMV in the Front-End Process

The front-end process involves the calculation of the cepstral features, in preparation for GMM model training or verification scoring. Front-end processing also usually includes a number of other stages that can be critical to achieving good verification performance.

In this experiment, the effect on performance of the location of the VAD, and the cepstral mean subtraction with variance (CMV) normalisation, within the feature extraction process is

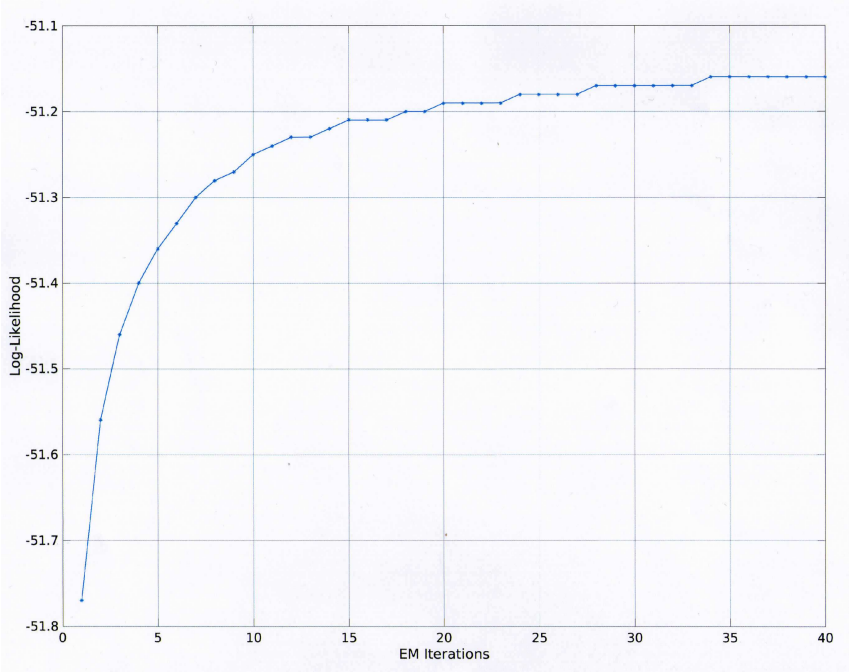
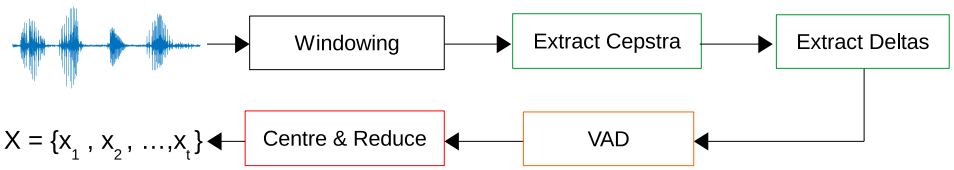


Figure 5.3: UBM log-likelihood probability $p(X|\lambda)$ with respect to EM training iterations, on the male SRE04 training data 'X', with 19C+E+19Δ+ΔE = 40 cepstra, and 512 Gaussian components.

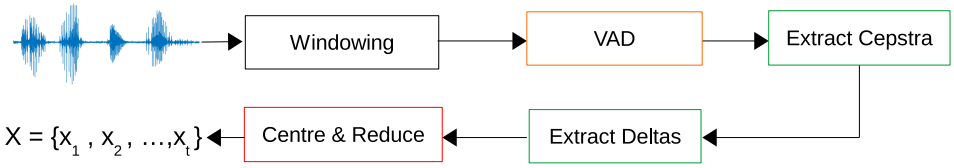
investigated. The VAD is important for removing silence or non-speech frames. CMV is used to remove convolutive type noise by subtracting the cepstral means, and can then be variance normalised.

Figure 5.4 shows the three front-end configurations considered, with (a) defined as the 'default' reference. In configurations (b) and (c) the VAD is moved to before the cepstral extraction, with (c) also additionally moving the CMV to before the delta feature extractor. Configuration (b) potentially corresponds to that used by Reynolds *et al* in [16].

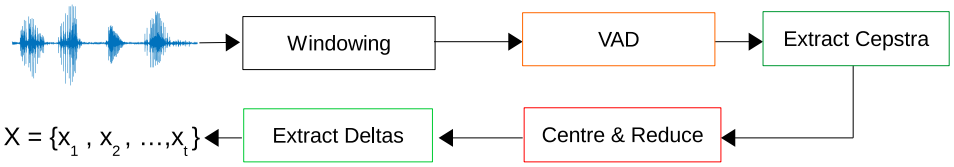
The precise cepstral feature order totals 38, consisting of 19 plus the delta coefficients, and excluding the energy coefficients. The use of zero'th energy coefficients was subsequently found in other experiments not reported here to not add any additional value, and so was discarded. The UBM used was trained again on the NIST-SRE 2004 male training data, consisting of 512



(a) Default chosen configuration (a).



(b) Configuration b.



(c) Configuration c.

Figure 5.4: The three different front-end configurations considered, with respect to the locations of the VAD and cepstral mean variance (CMV) normalisation.

Gaussian components with 20 EM iterations.

Figure 5.5 shows the resultant percentage EER (a) and minimum cost scores (b). Clearly in both plots, the default configuration 'a' can be seen to produce the best performance at close to 7.1% EER, and 3.1 cost score. The EER improvement is more than 0.5% relative.

A possible explanation for this result, is that applying the VAD prior to the cepstral and delta feature extractors might lead to 'glitches', as groups of speech frames are blindly concatenated together. If proven true, then this effect is expected to be more likely pronounced within the delta features.

It is therefore interesting that Reynolds *et al* [16] potentially uses front-end configuration (b).

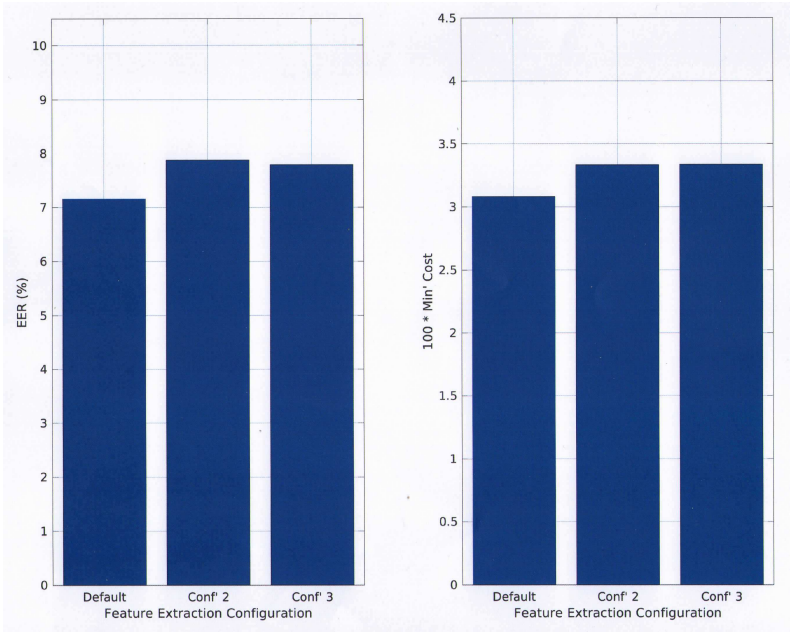


Figure 5.5: Respective %EER and cost performance scores with respect to the specific application of the VAD and CMV within the front-end feature extraction process. The three configurations are defined in Figure 5.5.

Their motivation is to remove silence periods prior to cepstral feature extraction. This remains an area of further research to understand the reasons for this, that is if in fact Reynolds *et al* [16] did configure their front-end in this form. An interesting further experiment for example might be to explore the dependency with the VAD threshold, in the form of DET curve.

Despite this observation, the results in Figure 5.5 indicate that for the specific experimental conditions used for this work, the VAD and CMV should be reasonably optimal. The performances reported in Figure 5.5, are also similar to the leading results reported for this type of configuration [17]. The 'default' front-end configuration is used in all subsequent experiments.

5.4.3 Acceleration Features, Background Training Data, and Model Size

In this experiment the use of acceleration cepstral features, amount of background speech training data used to train the UBM, and the number of Gaussian components is investigated. It is invariably expected that using larger amounts of training data should lead to better performance and robustness, provided the data can be learned effectively.

The percentage EER and minimum cost scores are compared with and without the use of acceleration features. The feature order is respectively 38 (19C+19Δ) without the acceleration features, and 57 (19C+19Δ+19$\Delta\Delta$) with. The UBM training data is incrementally increased, from the initial NIST-SRE 2004 male training set, to including all of the male audio from Switchboard II-Phase 1, and then similarly with Switchboard II-Phase 2. The UBM is equally increased in Gaussian component size, from the initial total of 512 to 1024, and then 2048. The number of EM iterations used to train the UBMs is fixed at 20 for all cases. Again, the comparable reference test set from the NIST-SRE 2005 '3conv4w-1conv4w' is used.

Figure 5.6 shows the %EER and minimum cost results for this experiment, with four bar charts displayed, for without acceleration features (left), and with (right). Comparing first the performance benefit with using acceleration, it is clear to see that there is a general overall improvement between the left and right plots. For example, without acceleration features with a UBM for 1024 Gaussian components, the EER is approximately 7.35% and cost 3.17. This decreases with acceleration features to approximately 6.8% EER and cost 2.98.

The slight exception to this, is with the %EER results when Switchboard II-Phase 2 data is added to the 1024 Gaussian component UBM. The performance can be seen to become marginally worse at 8.25% compared with 8.3%, if not effectively remaining unchanged. Critically however, the corresponding cost scores decrease, implying that within the operating threshold performance still improves.

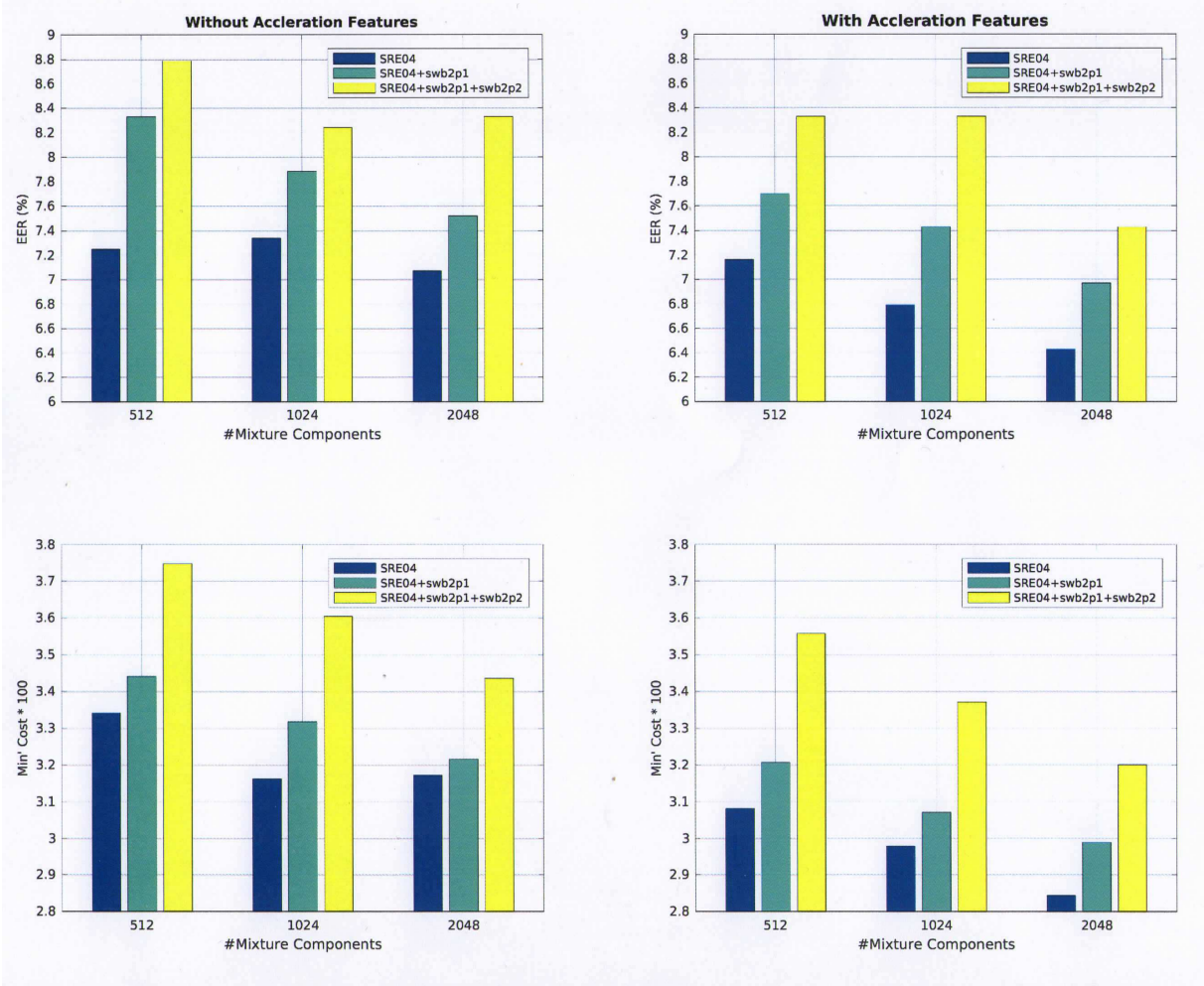


Figure 5.6: Percentage EER and cost scores with respect to with, and without acceleration cepstral features, and the amount of UBM training data (horizontal axis). No energy coefficients are used.

Focusing next on increasing the amount of data used to train the UBM, in general it would seem that adding Switchboard data unexpectedly degrades performance. For example the %EER at 512 Gaussian components without acceleration features increases from approximately 7.2% to 8.25%, and the cost 3.34 to 3.74. Increasing the number of Gaussian components though, appears to temper this degradation in general. The would suggest that UBM models with more than 512 Gaussian components are required to effectively learn the larger amounts of background data. This pattern also appears indicative with the use of acceleration features.

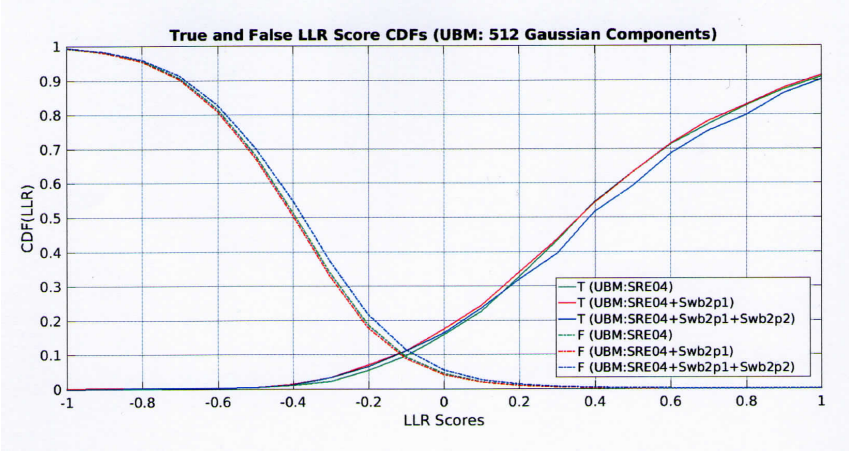
However the one significant exception to this is again with the Switchboard II-Phase 2, where comparing the %EER with acceleration features does not decrease at 1024 Gaussian components compared with the 512, essentially remaining fixed at 8.29% and 8.30% respectively. Further research is needed to understand this phenomenon with the %EER scores, but again the corresponding costs do nevertheless decrease as expected.

To try to understand the more significant issue, with the apparent progressive degradation in performance when Switchboard II data is added to the UBM, the corresponding true and false trial LLR cumulative distributions plots are shown in Figure 5.7.

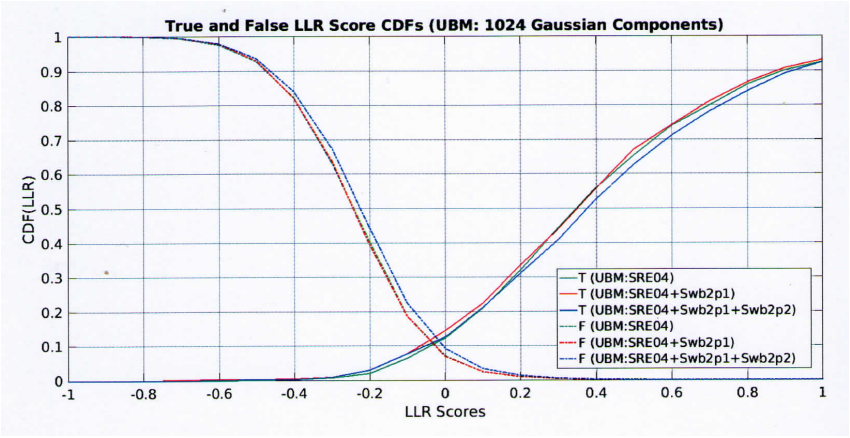
Three cumulative distribution plots are shown in Figure 5.7, corresponding respectively to the 512, 1024 and 2048 trained UBMs. Each plot contains six profiles, corresponding to both the true and false trials, with the three UBMs trained incrementally with SRE04 (training), and Switchboard II-Phases 1 and 2 data respectively.

Comparing all three plots in Figure 5.7, it is clear that there is an increasing positive LLR score offset as more background data is added. The 512 Gaussian component profiles in (a) can be seen to be centralised around a LLR score of -0.1, but this increases to approximately -0.03 in (b) with 1024 components, to then 0.02 in (c) with 2048 components. This would suggest that in the first instance, that some form of score normalisation is needed.

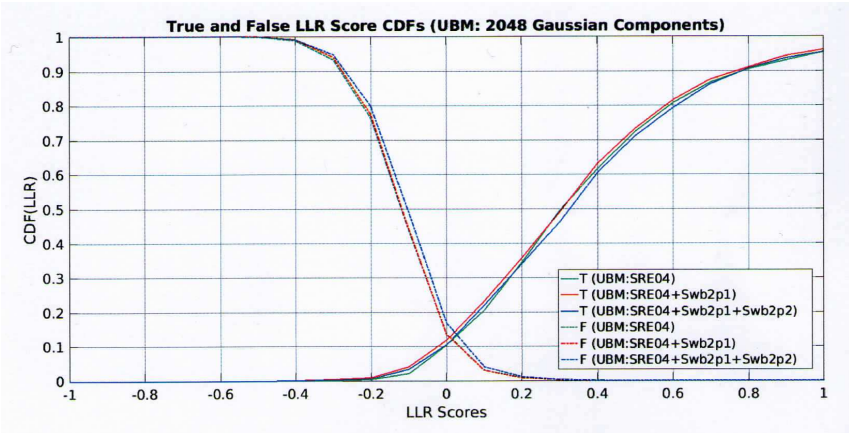
Examining next the individual plot profiles, the increase in verification error can be examined by comparing the regions of overlap between the true and false trial profiles. For the 512 Gaussian component UBMs in plot (a), the UBM only trained on the NIST-SRE 2004 data (green), clearly has the smallest region of overlap, with the true trial profile having a noticeably lower number of LLR score instances below 0.0, compared with the two UBM true profiles trained on Switchboard II data. Interestingly, the UBM trained on additionally both Switchboard II-Phases 1 and 2, appears to have an apparent positive offset, with both the false and true profiles. The shift is particularly consistent across the whole of the false profile (dashed blue), increasing the region



(a) 512 Gaussian components



(b) 1024 Gaussian components



(c) 2048 Gaussian components

Figure 5.7: True and false trial-cumulative distribution LLR score plots, at 512, 1024 and 2048 Gaussian components. Each plot contains the true and false score profile pairs for the UBMs trained incrementally with SRE04 (training), and Switchboard II-Phases 1 and 2 data.

of overlap, and hence the higher error rate.

The subtle frequency changes in the true and false LLR score distributions, when Switchboard II data is added to the UBM, indicates possibly that the GMM-UBM system is quite sensitive to speaker and channel variations. The Switchboard II data likely has a high degree of variation, because it intentionally contains a large number of speakers, and speaking over many different telephone channels. It is also according to Reynolds *et al* in [16], in their pioneering GMM-UBM MAP paper, well known that variations in telephone handsets can cause significant degradations in performance.

Intuitively, the EM training of the UBM involves essentially an undirected blind fitting of Gaussian distributions to the background data. There is effectively no allowance or compensation made for unwanted combinations of speaker and channel variations, which may in themselves be sparse or not that common. Adding then more background training data, which potentially contains such high degrees of speaker and channel variation, without allowing for this fact, is possibly the reason for the decrease in performance found. Further research is needed though to prove this hypothesis.

5.5 I-Vector Experiments: T-Matrix Training

The T-matrix is required for the extraction of i-vectors from speech utterances, mapping from the high dimensional supervector space to the low dimensional total variability space. In this experiment the training of the total variability matrix (T-matrix) is investigated, with respect to the amount of background speech data used, and also the number of Gaussian components in the UBM.

Multiple whole sets of corpora are typically used when training the T-matrix for robustness. In the principle i-vector paper by Dehak *et al* [3] for example, they used all of the Switchboard

Descriptor	*Configuration*	*Value*
UBM	Gaussian components	1024 & 2048
T-matrix	Rank Order	400
Features	Order	19C+19Δ+19ΔΔ=57
	Filterbank	26 Mel-Spaced Filters

Table 5.6: The UBM (Gaussian components), T-matrix (rank order), and the feature order hyperparameter values. The number of Gaussian components in the UBM is investigated at both 1024 and 2048 components. The number of EM training iterations for both the UBM and T-matrix is set at 20.

Descriptor	*NIST-04 (training)*	*SwbII-P1*	*SwbII-P2*	*FE-P1*	*FE-P2*
UBM (fixed data)	X	X	X		
T-Matrix (incremental data)	X	X	X	X	X

Table 5.7: The background male data used to train the UBM, and to estimate the T-matrix. The UBM data is fixed throughout the experiment, with three corpora used. The T-matrix is incrementally increased, starting with NIST-SRE 2004 (training), then adding Switchboard II-Phases 1 and 2, with last the two Fisher English-Parts 1 and 2.

corpora, NIST 2004 and 2005, and both part 1 and 2 of the Fisher English database. It is expected that using more training data should lead to both better, and more robust performance. However, it was observed in the previous GMM-UBM experiment presented in Sub-section 5.4.3, that unintentionally adding large amounts of potentially highly variable data in an effectively blind manner using the EM algorithm, can lead potentially to a degradation in performance. Thus in this experiment, the incremental increase in the amount of training data used to learn the T-matrix, and its component size via the UBM, are investigated.

Table 5.6 lists the UBM, T-matrix, and feature order hyperparameter values. As part of the T-matrix training experiment, the number of UBM Gaussian components is investigated at both 1024 and 2048. The T-matrix rank order, and the feature orders are fixed throughout. The number of EM training iterations used to learn the UBM and T-matrix is also fixed at 20.

The male background data used to train the UBM and T-matrix are listed in Table 5.7. The UBM data is fixed throughout the experiment, utilising the three male corpora indicated (NIST-

04 training, and Switchboard II-Phases 1 and 2). The training data for the T-matrix though, as part of the experiment, is incrementally increased to analyse the effect on ASV performance. In total five corpora are used, including the Fisher English-Parts 1 and 2.

The scoring procedure used through this experiment is simply the cosine distance, which is defined in [3] as

$$score\left(w_{hypothesised}, w_{test}\right) = \frac{\langle w_{hypothesised}, w_{test}\rangle}{\|w_{hypothesised}\|\|w_{test}\|} \underset{<}{\geq} \theta \tag{5.3}$$

where θ represents the threshold by which a verification decision is made, and w the respective hypothesised speaker and test utterance i-vectors to be compared. The intention in this experiment, is to explore the effect of the training of the T-matrix on ASV performance in the first instance, with the obvious next step to then investigate PLDA scoring.

5.5.1 Results and Analysis

Figures 5.8 and 5.9 show the respective DET performance plots, with incrementally increasing the amount of male training data used to estimate the T-matrix. Figure 5.8 shows the respective performances with (a) 1024 , and (b) 2048 Gaussian Components. In Figure 5.9, the two component total profiles with the larger amounts of Switchboard II data are plotted together for comparison, and to also analyse the effect of the adding even more T-matrix training data, with the Fisher English (Parts 1 and 2) corpora.

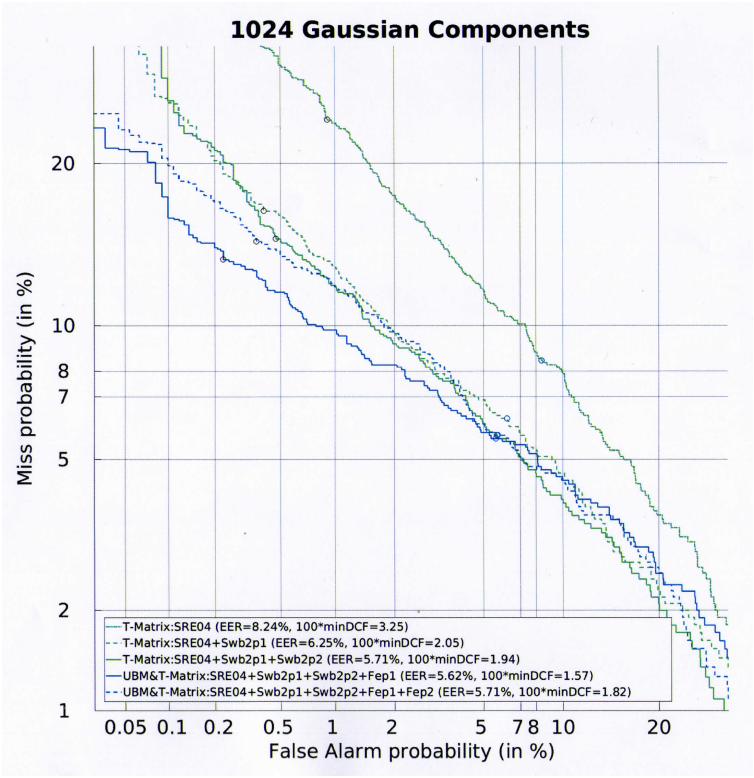
Analysing the 1024 and 2048 Gaussian component plots in Figure 5.9(a), it can be observed that using larger amounts of training data to estimate the T-matrix is required, to achieve performance beyond that of the GMM-UBM systems presented previously in Sub-Section 5.4. At 1024 components (a), using only the NIST-SRE 2004 (training) data to train the T-matrix, results in an EER performance of 8.24%, and cost of 3.25. Adding both Switchboard II-Phases

1 and 2 corpora then improves this to 5.71%, and cost 1.94. Comparing with the 1024 GMM-UBM results in Figure 5.6, the older GMM-UBM can be seen to outperform the newer i-vector approach when only using NIST-SRE 2004 data to train the UBM, scoring approximately 6.8% EER and with cost 2.98.

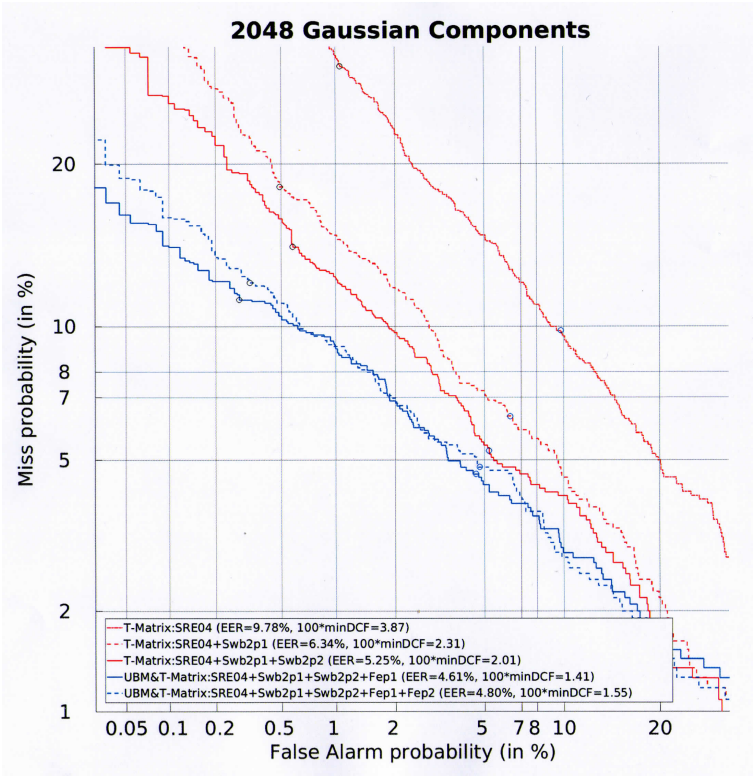
A possible explanation for this, is that first the GMM-UBM makes no attempt to manage unwanted increases in speaker and channel variabilities, which are not representative of the test conditions. The T-matrix in comparison, is specifically intended to be used to try to capture the principle speaker (and channel) dimensions of variability. Using more training data to estimate the T-matrix, should help then in general with the estimation of the principle component dimensions of total variability. However, one would expect that using larger amounts of data that is representative of the test conditions to train the T-matrix, is more likely to give the better performance.

The presence of unwanted variabilities in the T-matrix training corpora, is perhaps the reason for the slightly unexpected behaviour with the Fisher English-Part 1 and 2 (Fep1, Fep2) corpora. For both the 1024 and 2048 Gaussian component plots, adding Fisher English-Part 1 leads to a significant increase in performance, particularly with the 2048 components. For example, with 1024 Gaussian components, the cost score improves from 1.94 to 1.57 with adding Fisher English-Part 1, corresponding to a 19% improvement. With the larger complexity 2048 Gaussian component model, there is further expected improvement, with the cost score improving from 2.01 to 1.41, corresponding to an almost 30% improvement. However both the 1024 and 2048 DET plots indicate that adding Fisher English-Part 2 leads somewhat surprisingly, to a slight degradation in performance.

With 1024 Gaussian components, Figure 5.8(a) shows that adding Fisher English-Part 2 to the existing T-matrix training data, degrades the EER from 5.62% to 5.71%, and the cost score from 1.57 to 1.82. Similarly, with 2048 Gaussian components directly underneath in plot (b), the EER



(a)



(b)

Figure 5.8: DET plots showing the ASV i-vector performance with cosine scoring at (a) 1024 and (b) 2048 Gaussian components, on the male NIST-SRE 2005 (3conv4w-1conv4w) reference set, with respect to incrementally increasing the amount of male training data used to estimate the T-matrix.

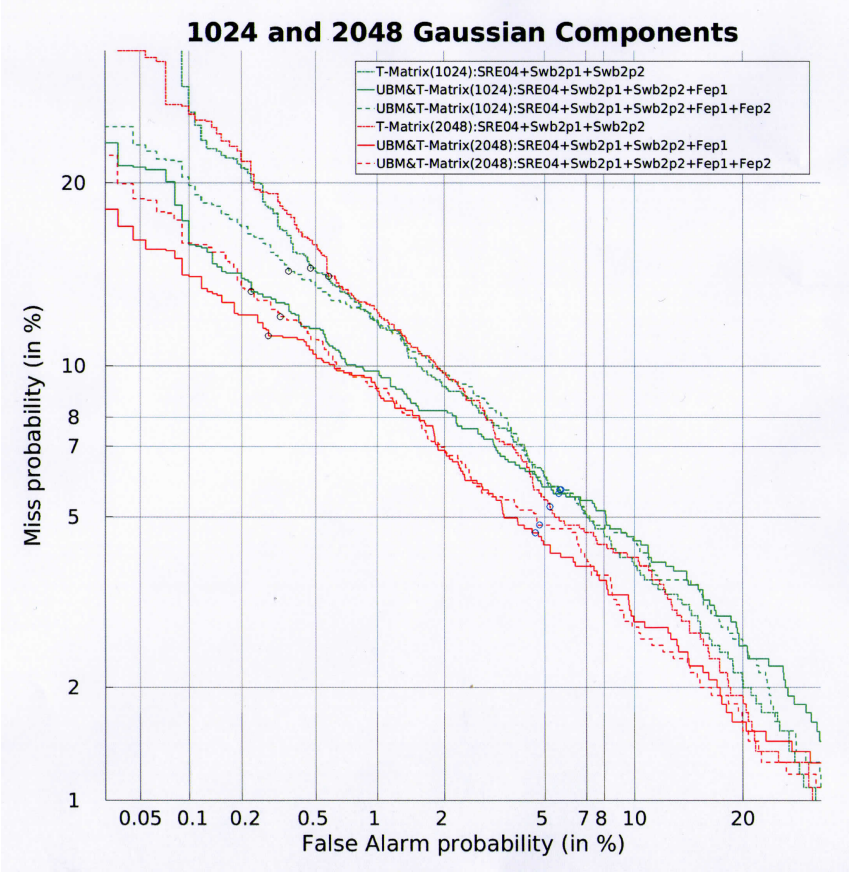


Figure 5.9: Combined 1024 and 2048 Gaussian component DET plot for ease of comparison, and for comparing with and without the use of Fisher English male training data in the training of the T-matrix, again on the male NIST-SRE 2005 (3conv4w-1conv4w) reference set with cosine scoring.

increases from 4.61% to 4.80%, and the cost 1.41 to 1.55. The decrease in ASV performance suggests that by adding the Fisher English-Part 2 data, this may have introduced unwanted variabilities during the training of the T-matrix. Further research is needed to try to isolate the source of this variability, but the Part 2 set appears to contain a larger number of speakers at 562 compared with 418, for approximately the same number of recordings. The smaller amount of training data per speaker with the Part 2 set, coupled with the larger number of speakers, is maybe then not as helpful.

The use of 1024 and 2048 Gaussian components is next specifically compared, with the combined DET plot shown in Figure 5.9. In total six profiles are shown, corresponding to the T-matrices trained with all the specified Switchboard II data, and with and without the two Fisher English corpora.

In general, Figure 5.9 appears to indicate that using 2048 Gaussian components does not begin to add benefit, until all of the NIST, Switchboard, and Fisher English corpora are included when training the T-matrix.

Examining the locations of the minimum cost points, it appears that only when Fisher English-Part 1 data is added on top of the NIST-SRE and Switchboard corpora used, does the use of 2048 components significantly advance upon the three 1024 component profiles. The 2048 minimum cost point improves to approximately 0.3% false alarm, with 12% miss.

Using otherwise 2048 Gaussian components without the Fisher English training data, leads it would appear to a poorer performance, compared with to using 1024 Gaussian components. At the minimum cost points, both the 1024 and 2048 Gaussian component profiles correspond to approximately 15% miss, but the false alarm probability becomes worse. Using 2048 Gaussian components, the false alarm probability corresponds to approximately 0.55%, compared with the lower value of approximately 0.45% with 1024 Gaussian components.

5.6 Experimental Conclusions

In this chapter a series of practical performance tuning experiments were presented. The experiments were motivated by the often limited amount of practical implementation details usually included in published works, which can make it difficult to repeat findings, and or understand the limits of leading performances. The experiments presented were split effectively into two parts, the first explored the earlier GMM-UBM ASV system, whilst the latter experiment focused on

i-vectors.

The GMM-UBM experiments investigated such practicalities, that included:

(1) The number of EM training iterations required to successfully train a UBM,

(2) The application of the VAD, and cepstral mean variance (CMV) normalisation, within the Front-End feature extraction process,

(3) And the effect on performance of acceleration features, amount of UBM training data, and model size (total Gaussian components).

The i-vector experiment presented, investigated the training of the T-matrix, which is required for the extraction of i-vectors from speech utterances. The T-matrix is typically trained across several large corpora for robustness [3]. The requirement of using such large amounts of data was explored, with incrementally increasing the amount of data used, and with respect to the number of Gaussian components defined by the UBM.

Summarising the GMM-UBM experiments in turn, it was found that 20 EM iterations appeared generally sufficient to successfully train a 512 Gaussian component UBM, on the NIST-SRE 2004 (training) male training data, with the log-likelihood training criterion effectively converged.

Probing next the specific application of the VAD and CMV, within the Front-End feature extraction process, it was found that applying the VAD after the delta feature calculation was best, and then the CMV normalisation. This led to a relative improvement of 0.5% EER (achieving 7.2%) over the two other configurations considered. It was also found during other experimentation, the use of energy cepstral features did not add any additional value to ASV performance.

However in slight contradiction, it was observed that Reynolds *et al* [16] potentially apply their VAD before their delta feature calculation, which was found here to be not as effective. Their

motivation for this, was to first remove unwanted silence time-frames. A possible reason for this difference, is perhaps that there is a dependency on the VAD threshold. It would therefore be interesting to investigate this further, but for the experimental conditions presented here, the EER and cost score findings are believed to relatively well optimised. The VAD trigger threshold is set at a more confident LLR score of 2.0.

It was observed that including the additional acceleration cepstral features led usually to an improvement in performance, by as much as 0.5% EER. Contrary though to expectation, using more data (Switchboard II) to train the UBM, in addition to the initial NIST-SRE 2004 (training) training set, was found to progressively degrade performance as more was added. Adding both Phases 1 and 2 of the Switchboard II datasets, decreased performance by as much as 1.5% EER.

Cumulative distributions of the true and false trial LLR scores were analysed, to try to understand why, adding more training data was leading to a decrease in ASV performance. The cumulative distribution plots highlighted initially that the LLR scores were becoming increasingly offset, as more Switchboard II data was added to the training of the UBM. This suggested that basic score normalisation, such as Z-norm or T-norm [29] should be applied in the fist instance, to try to compensate for any unwanted variability or train-test domain data mismatch.

A closer examination was then made of the regions of overlap between the true and false LLR score distributions, in an attempt to understand more the increased degradation in performance found. Subtle frequency variations in the cumulative true and false score distributions, when Switchboard II data was added, gave further indications that the GMM-UBM system might be again quite sensitive to unwanted variabilities.

A further observation was made, that the GMM-UBM ASV system does not effectively make any attempt to compensate or manage unwanted variabilities. The iterative EM algorithm used to train the UBM, essentially just attempts to best fit Gaussian component densities to the training

feature observations, without any specific prior speaker knowledge. It is therefore perhaps not entirely surprising then, that adding larger amounts of potentially variable Switchboard II data, in an effectively uncontrolled manner, degrades performance. The sensitivity to channel effects was also highlighted previously by Reynolds *et al* [16], in the context of handset variability, writing that this has been widely observed in the literature. The use then of 'Switchboard' type data, presumably alludes to similar channel difficulties.

Having thus established a reasonably robust GMM-UBM baseline system, the i-vector approach was next investigated. The T-matrix is a significant aspect of the i-vectors, mapping speech utterances from the higher dimensional supervector space to the lower dimensional total variability space. The training of this matrix was subsequently explored, in respect of the amount of training data used, and the number of Gaussian components in the UBM. The scoring procedure adopted, was the simple cosine distance scoring only.

It was found that larger amounts of training data were required to train the T-matrix, such that performance actually improved beyond that of the GMM-UBM experiments. Larger amounts of training data (i.e., including Switchboard II data on top of the initial NIST-SRE 2004 (training) is presumably required to better estimate the principle component dimensions of speaker (and channel) variability.

Intuitively as well, using 2048 Gaussian components over 1024, was not found to be helpful until all of the NIST-SRE 2004 (training), Switchboard II-Phases 1 and 2, and the Fisher English data was used. Performance in fact was marginally worse compared with limiting to 1024 Gaussian components, if only the NIST-SRE 2004 and Switchboard II data was used (i.e., without the Fisher English).

However whilst adding more data was generally found to be better, it was also found that potentially adding data that is maybe highly variable, and or not representative of the test conditions, can degrade performance. This was observed in particular when adding the Fisher

English-Part 2 corpora. It was hypothesised that adding this data degraded performance because it contains an additional 144 speakers, compounded with there also not being any more recording examples available to that found in the Part 1 corpora.

In conclusion, a performance of 4.61% EER, with a cost score of 1.41 was achieved. The experiments partially corroborate the opening observations made by Garcia-Romero [12], that ASV advancement has been made by the development of statistical models, which can leverage large amounts of data, and are efficient at adapting to scenarios with limited amounts of data. This is evident in particular with the training of the T-matrix with i-vectors. However these experiments have also highlighted the importance of using data that is representative of test conditions, and that they should contain a sufficient amount of information or examples of the variabilities they exhibit.

In terms of future research, an interesting area would be to examine the linear independent speaker and channel statistical model underpinning PLDA scoring. Kenny [41] in particular questions this assumption.

The application of PLDA typically leads to state-of-the-art performance [12] on telephony speech. However a better understanding of the relationship between speaker and channel variabilities will perhaps be required, if speaker recognition is to move truly beyond telephony conditions, and also English language; The focus on English has been driven particularly in recent times by the NIST-SRE trials [63]. This now motivates the initial study into the use of non-linear deep neural networks, which is presented in the next chapter.

Chapter 6

Deep Learning

The work presented thus far, has covered extensively the developments in automatic speaker verification (ASV) up to approximately 2012, with the advent of i-vectors [3], and the subsequent research into the use of PLDA scoring [12, 41, 42]. I-vectors coupled with PLDA still largely remain as the state-of-the-art in ASV [26].

More widely however, machine learning has been witnessing significant advancements made through the use of deep learning [18], particularly in image object recognition [19], and ASR [64]. Inspired by LeCun *et al* [18], they describe deep learning as, "Allowing computational models that are composed of multiple processing layers, to learn representations of data with multiple levels of abstraction."

Deep learning is typically encapsulated by the use of a multilayer neural network, with the layers of neurons attempting to progressively increase both the selectivity, and invariance of the required representation of chosen output classes. Increasing the number of layers increases the learning ability of the network. The term 'deep' refers to more than one hidden layer.

Deep learning with neural networks, can be conceptually viewed as attempting to extract robust features or representations of the data for classification, by way of a data-driven approach. Observation data is fed into the first layer, with the output layer corresponding to the classes to be learnt. Each layer then attempts to automatically learn representations that help to discriminate between the classes. Each subsequent higher layer notionally corresponds to a higher, more abstractive level of information about the classes.

A deep neural network (DNN) in practice is a multi-layer perceptron with multiple hidden

layers, randomly initialised, and trained using a form of stochastic gradient descent to learn the weights [65]. In speech processing, the inputs to the DNN are typically stacked spectral features, such as mel-frequency cepstral coefficients (MFCCs), extracted from short-time frame intervals in the lower tens of milliseconds. Typically also a wider context of up to in the order of 10 frames (+/-5 frames) is used [65, 66]. The outputs of the DNN are the posterior probabilities of the target classes with respect to the current input observation data.

Much of the work of late in ASV appears to be focused around the use of DNNs for ASR [22, 25, 56]. This is likely motivated by the larger amounts of speech data available for training ASR systems. As a probable consequence of this, it appears that there has been less research published around the use of deep networks defined with output classes, which are specifically speaker related. Variani *et al* [23] is probably one of the first published such works, where they consider a standard feed-forward DNN for ASV, but they limit to a text dependent application. However, it would seem that research interest in the direct training of DNNs for ASV is increasing, with in particular eminent work by Snyder *et al* [26].

The effective use of deep learning within ASV to discover new robust representations, can be regarded therefore still as largely an open question. This motivates the initial study into the use of deep learning presented in this chapter, which begins by first reviewing such research to date.

The chapter then presents some initial experimental work into the development of a deep convolutional neural network (CNN) for speaker identification. The use of CNNs is motivated by the success found in image object recognition [19], where filterbank spectra can be also perceived as images. It is hoped that this work in the future might lead to a better understanding of the complex relationship between speaker and channel variabilities, highlighted previously with PLDA in Section 4.3.2, and to more robust features for ASV. The chapter then ends with some conclusions.

6.1 The Indirect DNN-ASR Approach for ASV

It is fairly apparent from the literature that there are two main approaches adopted within ASV for applying deep learning. The first, is one in which a pre-learnt DNN for ASR is used [65,67]. The second involves the more direct approach, where the output classes are speaker related [23,26,68]. In this section a brief account of the indirect approach, using a pre-learnt DNN for ASR is first presented, followed by a slight interjection of McLaren *et al*'s [25] extended work on convolutional neural networks (CNNs) for noisy conditions.

The use of a pre-learnt DNNs for ASR is originally inspired by Lei *et al* [22]. They recognised that unlike in other speech-related fields, there is often limited amounts of speaker dependent training data, making they claimed the direct transition to automatic speaker recognition challenging. To circumvent the issue of limited data, they instead proposed the use of a pre-learnt DNN for acoustic modelling in ASR, to form an alternate phonetically aware UBM to help guide ASV.

Following Lei *et al* [22], the i-vector model assumes that the t-th observation vector x_t is generated by the GMM defined by

$$x_t \sim \sum_c \gamma_{ct} N\left(\mu_c + T_c w, \Sigma\right) \tag{6.1}$$

where c represents the Gaussian mixture component; T_c the total variability matrix representing a low rank subspace (the total variability subspace) by which the means of the Gaussians are adapted to a speech recording; w is a normally-distributed latent vector that is recording specific (the i-vector); μ_c and Σ_c are the speaker independent mean and covariance matrix of the c-th Gaussian of the speaker population; and γ_{ct} is the posterior of the c-th Gaussian, given by

$$\gamma_{ct} = p\left(c|x_t\right) \tag{6.2}$$

γ_{ct} therefore defines the alignment of observation vector x_t to Gaussian component c at time t.

As pointed out in [22], the Gaussians in a UBM are traditionally used to define the classes c in Equation 6.1. Instead Lei *et al* in [22] propose the use of senones, defined by the leaves of an ASR phone decision tree. This does however make the assumption that each of the senones can be accurately modelled by a single Gaussian. Lei *et al* [22] advise that whilst this is a strong assumption, the results from their work show that it is reasonable for ASV.

Senones effectively correspond to the underlying tied-triphone states of the hidden Markov model (HMM) states [69]. In ASR, usually a set of simple pre-defined phonetic left and right context questions is used to grow a decision tree, by splitting the triphones of the same central phoneme into sub-groups so as to maximise the likelihood of the training data. The clustering of the triphones to produce tied-state variants, is performed to mitigate training data sparsity issues that commonly arise during training [70]. Due to there often being many triphone states present, inevitably many states will usually have an insufficient number of training examples available to estimate them robustly.

Lei *et al*'s [22] motivation for defining the GMM classes in Equation 6.1, according to the posteriors of tied-triphone states in an ASR system, is to attempt to embed phonetic alignment information into the speaker models. They draw attention to how in a classic UBM, the Gaussian classes are only defined so as to maximise the likelihood of the model. The classes do not they point out, have any inherent meaning, other than covering parts of the acoustic feature space.

However if the classes in Equation 6.1 correspond to phonetic senones, and the posteriors of the senones can be accurately estimated, then conceptually the utterance short-time frames can be aligned by phonetic (senone) content prior to computation of the i-vector. Notionally, speakers can thus be compared, aligned by their pronunciations of phonemes relative to the general population. In practical terms, observation frames are statistically assigned to their corresponding senone classes, which restricts the MAP adaptation shift of the means from the population mean μ_c in Equation 6.1, to only the aligned senone classes.

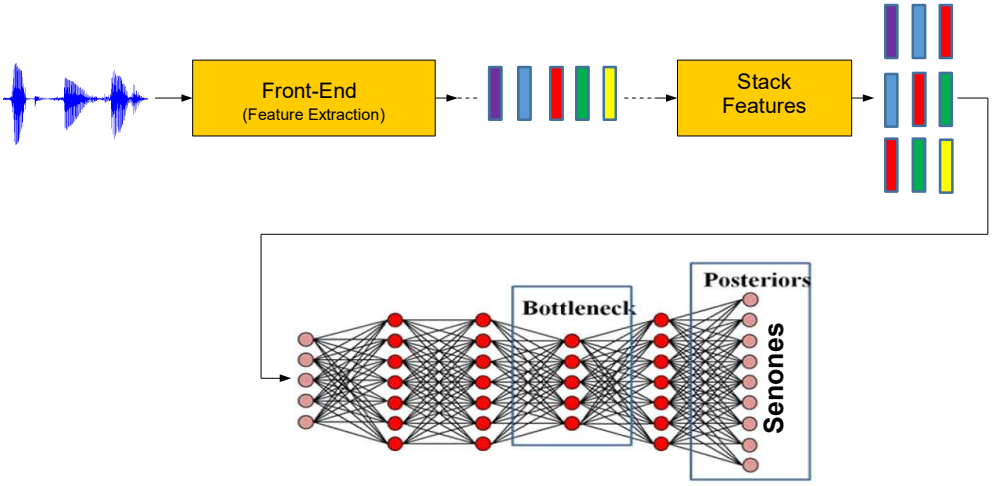


Figure 6.1: Illustration of the indirect DNN-ASR architecture taken from [71], where the output classes of the DNN are defined as the phonetic senone states, which are also effectively the tied-triphone states of an ASR-HMM. The acoustic features are stacked around the current input frame (in practice +/-5 frames [66, 71] context) for input into the DNN. Bottleneck features can also be extracted by restricting the number of nodes in one of the hidden layers, and taking its output as features [56, 72].

Taken from [71], Figure 6.1 illustrates the indirect DNN-ASR architecture proposed, where the output classes to the DNN as indicated are defined to be the phonetic senone states. A speech utterance produced by a speaker is first processed by a front-end module, which extracts typically short-time frame cepstral acoustic features [66]. The feature vectors are then stacked around the current input frame, with typically a wide context of +/-5 frames [66, 71] for input into the DNN.

Figure 6.1 also highlights, how 'bottleneck' features can be extracted from the DNN, by restricting the number of nodes in one of the hidden layers and taking its output. The use of bottleneck features for ASV is investigated by Lei *et al* in [72], motivated by earlier work in language identification [73].

In the first instance, Lei *et al* [22] trialled simply replacing the standard acoustic UBM with their new phonetic senone derived UBM trained on transcribed speech data, where the Gaussian

components are given by

$$\begin{aligned} \gamma_{ct} &\approx p(k|x_t) \\ n_c &= \sum_t \gamma_{ct} \\ \mu_c &= \frac{1}{n_c} \sum_t \gamma_{c,t} x_t \\ \Sigma_c &= \frac{1}{n_c} \sum_t \gamma_{c,t} x_t x_t^* - \mu_c \mu_c^* \end{aligned} \tag{6.3}$$

and the posteriors γ_{ct} for Gaussian component c for observation features x_t, are defined by the ASR system (using Bayes rule to convert to required posteriors). The terms μ_c and Σ_c, represent the means and covariance matrices for each of the Gaussian components c in Equation 6.1.

Traditionally a GMM was used to model the observation probability $p(x|q)$ for ASR decoding. According to McLaren *et al* [56], once the set of senones was derived from the decision tree, a Viterbi decoder was used to align the training data to the corresponding senones. The resultant alignments were then used to estimate the observation probability $p(x|q)$.

Unfortunately Lei *et al* [22] found this approach did not lead to significant improvements compared to just using a standard UBM, without knowledge of the phonetic knowledge. They concluded that this was due to the poor phonetic recognition accuracy of the GMM-based method; more precisely, if an observation frame is assigned to the wrong senone, then it is no better to having just used a conventional unsupervised UBM.

Lei *et al* [22] further point out that in ASV, context is only captured usually by way of delta and double delta cepstra features. They argue that this is not sufficient for predicting accurately phonetic content, and with using only a single GMM with a few thousand Gaussian components.

Fortunately, state-of-the-art ASR systems now typically use a standard feed-forward DNN, to more accurately estimate the senone class posteriors ($p(q|x)$) for observation features (x). However the initial training of the DNN still does rely on a pre-trained HMM ASR system, with GMM states to generate the training alignments.

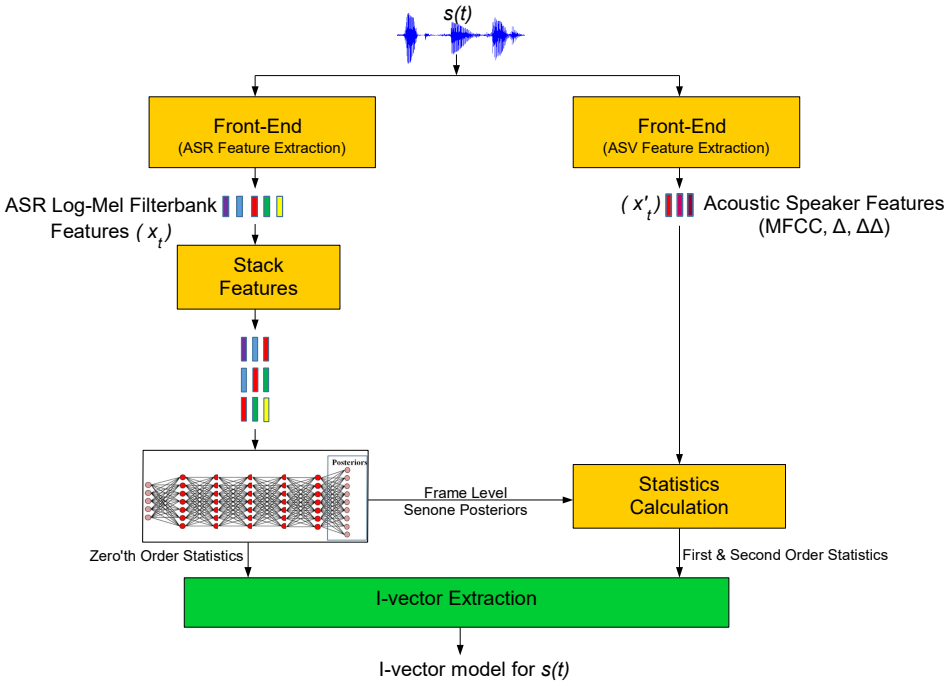


Figure 6.2: The proposed 'DNN/i-vector' framework proposed by Lei *et al* [22], where the ASR trained DNN is used to more accurately estimate the zero'th order utterance level statistics, and the frame level senone posterior probabilities for alignment. The diagram also illustrates how the features for the ASR-DNN (log-mel filterbanks = x_t), are not incumbent on the features used for ASV (e.g., MFCCs + Δ + $\Delta\Delta$ = x'_t).

Figure 6.2 shows the 'DNN/i-vector' framework proposed by Lei *et al* [22], where the DNN trained for ASR, is used to estimate the zero'th order, and frame level senone posterior probabilities ($\gamma_{c,t} \approx p(c|x_t)$ at time t and Gaussian component c). Figure 6.2 also illustrates how the features used by the DNN are not incumbent on the features used for ASV. Lei *et al* [22] for example, use the log outputs from the Mel-filterbanks (x_t) for their DNN, but MFCCs+delta coefficients (x'_t) for the end ASV task. The frame level posteriors, which provide phonetic senone alignment, are then used to estimate the first and second order utterance level statistics required for i-vector extraction (Equations 6.3), using cepstral features x'_t.

The i-vector for utterance $s(t)$ can be thus computed using all required Baum-Welch statistics,

defined in [22] as

$$N_c = \sum_t \gamma_{ct}$$
$$F_c = \sum_t \gamma_{ct} x'_t$$
$$S_c = \sum_t \gamma_{ct} x'_t x'^*_t$$

where N_c, F_c, and S_c are the required zero'th, first and second order statistics; and $gamma_{ct}$ the posterior for the c'th Gaussian at time frame t.

The indirect DNN-ASR approach inherently requires the pre-training of a (HMM) ASR. However once trained, Figure 6.2 illustrates how the HMM ASR decoding component is no longer required. Lei *et al* [22] find the DNN to provide much more accurate estimation of the senone posterior probabilities, likely they speculate due to both the larger time frame context used (+/-7 frames), and DNN based discriminative training.

6.1.1 Performance of the DNN/i-vector Framework and Bottleneck Features

Two ASV methods are effectively highlighted from the indirect DNN-ASR approach. The first is naturally the underpinning 'DNN/i-vector framework', of substituting the acoustic feature derived UBM for the phonetically derived senone based UBM, and using the DNN for accurate estimation of the senone posteriors.

The second method is the use of 'bottleneck' features [56, 72], extracted from the DNN by restricting the number of hidden nodes in one of the layers. The outputs from the bottleneck layer are then taken as a new set of features for each acoustic feature frame, and used for ASV by modelling the features using the standard UBM/i-vector or the DNN/i-vector frameworks. The use of bottlenecks features was motivated by earlier usage in language identification [73].

According to [56], the bottleneck features can also be appended to the conventional cepstral features, where they found provides significant performance gains.

McLaren *et al* [56] typically used the second-to-last layer as the bottleneck, with the number of hidden nodes reduced to 80 from 1200 nodes. Their DNN otherwise consists of 5-layers with 3494 senones, which they train on 800 and 1300 hour of microphone and telephone speech respectively. For the input to the DNN, they use the log mel-filterbank outputs from 40 filterbanks. The feature vectors are stacked across 15 consecutive frames, producing a 600-dimensional contextualised input to their DNN.

Table 6.1 shows a summary of the findings from McLaren *et al* [56], where they compare different configuration performances relative to the classical i-vector based system. Their results are derived on the NIST-SRE 2012 extended clean telephony (core 2), and clean microphone (core 1) conditions, which contain an upper limit of 100M trials. The performance figures in Table 6.1 correspond to the NIST-SRE primary cost score, which is the weighted sum of two different cost operating points, based on a lower (0.001) and a higher (0.01) pre-defined target prior [63].

In total six combinations are shown, with '*UBM (MFCC)*' referring to a classically trained UBM/i-vector framework trained on mel-frequency cepstral coefficients (MFCC). '*DNN (BN)*' refers to the new DNN/i-vector framework with the UBM instead derived from senone posteriors, and '*BN*' the use of bottleneck features during i-vector extraction. The term '*pcaDCT*' refers to an alternative acoustic feature to MFCCs investigated in [56].

Table 6.1 illustrates how in telephony clean conditions, using instead the *DNN (MFCC)* at 0.184, the *UBM (BN)* at 0.165, or the *DNN (BN)* configuration at 0.189, leads to significant improvement over the *UBM (MFCC)* baseline at 0.257. The largest improvement can be seen to correspond to the *UBM (BN)* configuration. McLaren *et al* then investigated fusing multiple combinations, which led to even further improvements, with 0.143 using *UBM (MFCC+BN)*, and 0.137 with *DNN (pcaDCT+BN)*.

Channel	*UBM (MFCC)*	*DNN (MFCC)*	*UBM (BN)*	*DNN (BN)*	*UBM (MFCC+BN)*	*DNN (pcaDCT+BN)*
Tel-Clean (c2)	0.257	0.184	0.165	0.189	0.143	0.137
Mic-Clean (c1)	0.187	0.176	0.172	0.207	0.134	0.156

Table 6.1: Comparative NIST-SRE primary cost scores taken from [56] on the NIST-SRE 2012 extended clean telephone (core 2) and microphone (core 1) conditions, comparing indirect DNN-ASR ASV performances using different combinations of the DNN-senone derived UBM and bottleneck features, relative to the classical UBM Mel-cepstral feature (MFCC) i-vector baseline. pcaDCT are alternate acoustic features to MFCC investigated in [56].

Their findings suggest for matched telephony train-test conditions, the use of the new DNN-ivector framework and bottleneck features demonstrate clear ASV performance gains over the classical i-vector framework using MFCC features. Interestingly the further improvement in performance found by fusing, also appears to highlight how the information derived using acoustic features (MFCC or pcaDCT), is notionally orthogonal to that derived via the new DNN i-vector framework. This also perhaps explains the 0.165 lower score found with *UBM (BN)* configuration, compared with the *DNN (MFCC)* at 0.184, and *DNN (BN)* at 0.189.

Analysing next their microphone derived results in Table 6.1, it immediately appears that the recording conditions are extremely beneficial for the baseline *UBM(MFCC)* system, with an improvement from 0.257 to 0.187. The specific sample rate is not specified in the NIST-SRE 2012 evaluation plan [63]. However the bit depth used for the microphone derived speech is double that of the telephony, at 16 bits. This would suggest that the recordings contain in the very least, less quantisation noise and are of a higher quality.

Again in general, with the exception of the *DNN(BN)* scoring 0.207, the use of the new DNN i-vector framework and bottleneck features leads to an improvement over the baseline. Fusing again leads to further significant improvement, with the *UBM (MFCC+BN)* configuration returning the best result overall at 0.134.

McLaren *et al* [56] speculate that anomalous lower performance found with the *DNN(BN)* at

0.207, is likely an artefact of using DNNs not well suited to the microphone characteristics. Given the also noticeable poorer performance of the *DNN (pcaDCT+BN)* at 0.156, relative to the *UBM (MFCC+BN)* at 0.134, might suggest that the DNN is quite susceptible to channel condition changes or over-tuning.

6.1.2 The CNN/i-vector Framework for Noisy Conditions

Inspired by the use of CNNs for ASR on noise degraded speech [74] and, they claim, their successful application of an alternate CNN/i-vector framework to language identification [75], McLaren *et al* [25] also investigated the use of convolutional neural networks (CNN) instead in their DNN/i-vector framework for noisy conditions.

Figure 6.3 illustrates their CNN-ASR deep network, trained similarly to estimate phonetic senone class posteriors. The difference with their CNN deep network compared with the DNN, is that they substitute the first layer for a convolutional one. However the rest of their deep network remains unchanged, consisting of between 5 to 7 fully connected feed forward layers as before in [25].

CNNs are a biologically inspired variation of multi-layer perceptrons (DNNs) [76], following similar observations to the early work of Hubel and Wiesel [77]. In their research into understanding the visual system, Hubel and Wiesel [77] discovered that regions of neurons in a cat's visual cortex are sensitive specifically to localised regions of the visual field (receptive field), and also that some neurons are orientation selective. CNNs in a manner similar, are constructed such that they are sensitive to specific localised patterns, but also then invariant to their precise location. These localised patterns or features, can then be combined in subsequent higher layers to derive higher-order semantic level features [78]. The use of CNNs by Krizhevsky *et al* [19], led to the pivotal halving of errors in the 2012 ImageNet visual recognition challenge [18].

According to [79], CNNs comprise of multiple alternating convolutional and pooling layers. At

each convolutional layer, a linear filter is convolved by computing its inner product against its current local receptive field, followed by a non-linear activation function. This localised process is repeated across the input observation data or visual field, producing a filtermap for each convolutional filter respectively.

The linear filters effectively correspond to the weight vector of neurons, which along with the bias, are in essence replicated across the entire visual field. These replicated neurons therefore share the same weight vectors, and biases. This intentional property allows CNNs conceptually to detect distinctive patterns or features, regardless of their position [76]. The training of a CNN, therefore involves deciding initially on the optimum number and size of the filters, such that the patterns and features important for end class discrimination are learnt.

Figure 6.3 illustrates the convolution calculation with a diagram derived from McLaren *et al* [25]. A single nominal linear filter of size five frames by two filterbank coefficients, is convolved with a filterbank spectral image restricted to the frequency axis only (in image processing, convolutions are performed in 2-dimensions to reflect that objects can occur anywhere in an image, which is not usually the case for speech). With a step size of one, this produces a vector of length six. Typically multiple filters are also used, producing multiple corresponding maps as illustrated. It should also be noted that each replicated neuron, with weights corresponding to the linear filter, also includes a bias term. The bias adjusted convolutional output scores are then passed through a non-linear activation function prior to max pooling [79].

Following the convolution layer, max pooling or another form is normally applied, and can be considered a form of down-sampling [18, 78]. Max pooling involves computing the maximum of a locally specified patch, and aids in translational invariance, and reducing complexity at the higher layers [76]. Final training of the CNN is then said to be no different to with a DNN, with backpropagating gradients [18]. Figure 6.3 illustrates max pooling with a group size of three with no overlap, producing a vector of length two from the original size of six.

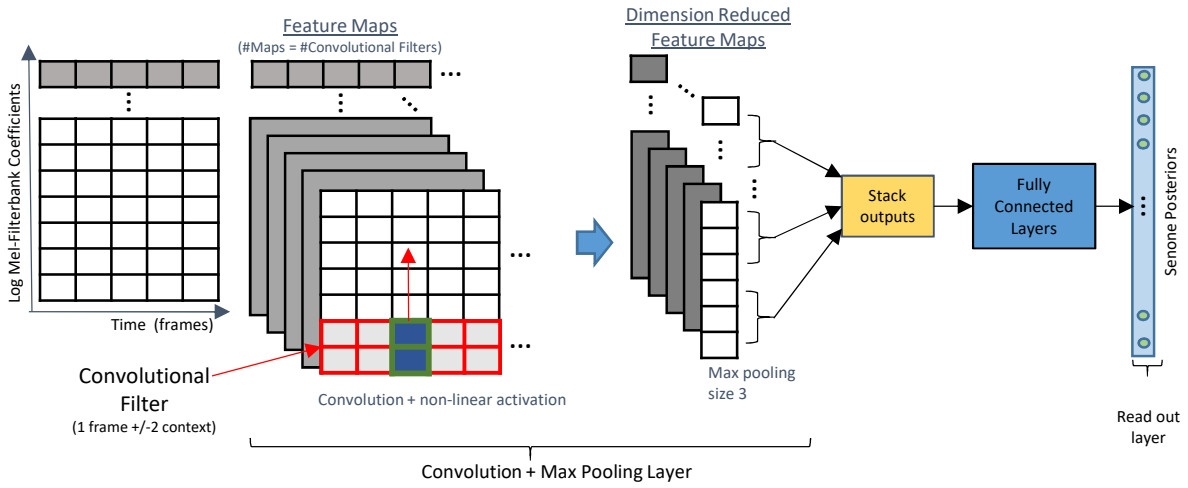


Figure 6.3: Illustration of the CNN-ASR deep network used by McLaren *et al* [25], for their alternate CNN/i-vector framework for ASV in noisy conditions. Compared with their original DNN/i-vector framework in [22], the first layer is substituted for a convolutional layer instead. The remainder of the network is unchanged, consisting of between 5 to 7 fully connected layers. Following the diagram in [25], only one convolutional filter is shown, but in total they use 200 filters, generating 200 corresponding 'filter maps'. The filters are convolved with the filterbank spectral image along the frequency access only. The size of the filter used is also larger in practice than shown, with a context of 15 time frames (equal to the CNN input), and a height normally of 8 filterbank coefficients. They use non-overlapping max pooling, with a pooling size of 3. In the illustrative figure, this produces a 2-dimensional output vector.

McLaren *et al* [25] in their experimentation, first extract 40 log Mel-spaced filterbank coefficients, with a context of 15 frames to produce a stacked vector of length 600 for input into the CNN. In total they used 200 convolutional filters, restricting to a 1-D convolution along the frequency axis as illustrated in Figure 6.3, but using a context of 15 frames thereby matching their CNN input size. They then concatenate the output vectors from the CNN layer into one long vector, and input it into their traditional fully connected DNN layers, of which they use between five to seven layers.

McLaren *et al* [25] state that they choose to restrict convolution to the frequency dimension, based on the earlier findings of a study conducted by Abel-Hamid *et al* [80] for ASR. In their study, they find that convolving along the time dimension degrades ASR performance, conclud-

$Configuration$	$\%Miss@1.5\%FA$	$\%EER$
UBM(PLP)	33.3	9.4
CNN(PLP)	30.2	8.5
CNN(PNCC)	29.5	8.5
UBM(PNCC)	27.5	8.1
CNN(PLP) + CNN(PNCC)	27.5	8.1
UBM(PLP) + UBM(PNCC)	23.9	7.4
CNN(PNCC) + UBM(PNCC)	20.8	6.7
CNN(PLP) + UBM(PNCC)	20.4	6.6

Table 6.2: Percentage miss at 1.5% false alarm (FA) and percentage equal error rate (%EER) scores taken from McLaren *et al* [25] in descending order, derived on the RATS SID 10s-10s (enroll[6x10s]-test) [83] noisy radio re-transmissions, comparing the use of the classic UBM/i-vector extraction [3] to their proposed CNN/i-vector framework, and with fusion. They also make comparisons between using perceptual linear prediction (PLP), and power normalised cepstral coefficient (PNCC) features. The matched language test set used consists of 85K target and 5.8M impostor trials, from 305 unique speakers.

ing that perhaps the implicit word or phonetic shift detracts from performance. Despite this issue, they found the use of a CNN still to perform favourably over their original DNN.

Again like in [22], each output node from the DNN corresponds to each senone defined by the ASR decision tree. Similarly to [22], McLaren *et al* [25] note that a pre-trained HMM ASR with GMM states is needed to generate the initial timing alignments, prior to CNN training.

Table 6.2 shows percentage miss at 1.5% false alarm (FA) rate, and percentage EER scores taken from McLaren *et al* [25], presented in descending order. They compare their proposed CNN/i-vector framework to the classic UBM/i-vector [3], and with fusion. They also compare two types of input features: perceptual linear prediction (PLP) [81], and power normalised cepstral coefficients (PNCC) [82]. McLaren *et al* [25] describe PNCC features as using a power law to design the filterbank, and a power-based normalisation instead of a logarithm. For reference, the features used to train the CNN are independent to that used for ASV [22]. This was discussed in Section 6.1 previous.

Their results are derived on the 10s-10s (enrol-test) trial set from the noisy RATS corpora [83],

where according to McLaren *et al* [25] the 10s train implies six recordings each in duration 10 seconds. The RATS speech data is sourced from noisy radio re-transmissions. The matched language 10s-10s set used to derive their results shown in Table 6.2, consists of 53K re-transmissions from 5899 speakers, and a matched language test set of 85K target (hypothesised) and 5.8M non-target trials from 305 unique speakers. The languages are Levantine Arabic, Dari, Farsi, Pushto and Urdu.

The results in Table 6.2 show that PLP features perform better with the CNN/i-vector framework, at 30.2% miss compared with 33.3% for the UBM(PLP). However the reverse is found for the UBM/i-vector framework, with UBM(PNCC) giving 27.5% miss compared with 29.5% for the CNN(PNCC). The best single configuration result is therefore found using the classical UBM/i-vector framework [3], albeit using PNCC features.

Fusing again is found to lead to significant performance improvements in the percentage miss scores, which correspond to a more realistic operating point compared with the EER. The best configuration found is with CNN(PLP) + UBM(PNCC), scoring 20.4% miss, corresponding approximately to a significant one-third improvement. The results, like in [22] again highlight how the information contain in the (DNN/CNN)/i-vector framework is potentially orthogonal to a degree, with that contained in the UBM/i-vector framework. It would be interesting therefore also, to compare the performance with the DNN/i-vector, to try to understand the potential benefit of the CNN over the DNN.

On a related CNN aspect, research has also taken place recently, with experimenting with time-delay neural networks (TDNNs) [84, 85]. In a TDNN [86], the filter is set to match the number of filterbank coefficients, thereby restricting convolution to the time axis. TDNNs are however structured, such that the lower layers only have a restricted time context, with higher layers processing increasingly wide activation contexts. Therefore unlike in a typical DNN, information is not potentially lost due to averaging effectively across an entire temporal context [84]. The

$Configuration$	$EER\%$	$minDCF10^{-3}$	$minDCF10^{-2}$
UBM(5297)/i-vector	2.00	0.410	0.241
UBM(4096)/i-vector	1.96	0.414	0.227
TDNN(5297)/i-vector	1.09	0.214	0.108

Table 6.3: Percentage EER and minimum cost scores for two operating points for gender dependent models taken from Snyder *et al* [85], on the NIST-SRE 2010 core five extended telephony condition, comparing the classic UBM(5297)/i-vector [3] with 5297 Gaussian components, and their TDNN(5297)/i-vector framework. The minDCF10^{-3} refers to the new NIST-SRE 2010 minimum cost cost operating point, with an a-priori hypothesised speaker probability of 0.001.

context restriction of the lower layers, also potentially resolves the issues found by Abel-Hamid *et al* [80], with the word and phonetic variability along the time dimension.

Peddinti *et al* [84] in 2015 re-investigated the use of TDNNs for ASR, which was then experimented with by Snyder *et al* [85] for ASV. Peddinti *et al* [84] find for ASR, that the use of TDNNs leads to a pivotal improvement on average of 4.3% word error rate, across several large vocabulary continuous speech recognition (LVCSR) tasks. They conclude that the TDNN is able to learn wider time frame contexts better than DNNs. They also find that the structuring of the multiple TDNN layers, such that the higher layers have increasingly wide contexts is also beneficial. Intuitively, this suggests that the lower layers learn to detect effectively localised features, whilst the higher layers learn the wider complex patterns across time.

Similar substantial improvements are found by Snyder *et al* [85], when applying TDNNs for ASV, reporting a 50% EER relative improvement on NIST-SRE 2010 extended condition five (telephony) data. Table 6.3 is taken from [85], and shows a summary of their findings, with the %EER improving from their classical UBM(4096)/i-vector baseline of 1.96%, with 4096 Gaussian component classes, to 1.09% for the TDNN(5297)/i-vector. Their results also show that the use of TDNNs leads to comparable performance improvements, at both minimum cost operating points. The UBM(5297)/i-vector for minDCF^{-3} for example improves from 0.410 to 0.214. NIST in 2010 defined a new minimum cost operating point, with a lower a-prior probability of 0.001 for an hypothesised speaker.

Both Snyder *et al*'s [85] and McLaren *et al* [25] therefore demonstrate the effectiveness of CNNs for ASV even if applied indirectly through the use of ASR. It would be interesting however to compare performances directly between DNNs and CNNs, to understand better the benefits of applying these networks under different joint conditions.

6.2 Direct Training of DNNs for ASV

In the direct DNN-ASV method, the DNN is directly trained with output classes that are specifically speaker related. Choosing to define speaker related classes, effectively directs the deep network to discover potentially new robust representations or features for ASV. This approach can be notionally defined as "End-to-End", in view of that the model parameters are all learnt jointly for the end speaker recognition task.

Somewhat surprisingly, the number of published works investigating this direct method appears limited, when compared to the indirect DNN-ASR path. This is perhaps a consequence of the limited amount of labelled ASV data available per speaker [22]. However of late, research interests in direct methods appear to be increasing, with a number of published works [26,87,88]. Two of probably the most prominent methods, include that of 'd-vectors' by Variani *et al* [23], and very recently 'speaker embeddings' by Snyder *et al* [26]. The theory behind these two methods is examined, with a review then made on the experimental findings found.

6.2.1 D-Vectors via DNNs for Speaker ID

One of if not the first approach to training a DNN directly for ASV, was proposed by Variani *et al* [23] in 2014, where they proposed the use of 'd-vectors'.

Figure 6.4 illustrates how they first train a background DNN for speaker identification, taking in stacked filterbank and energy features, with the output classes defined as individual speakers.

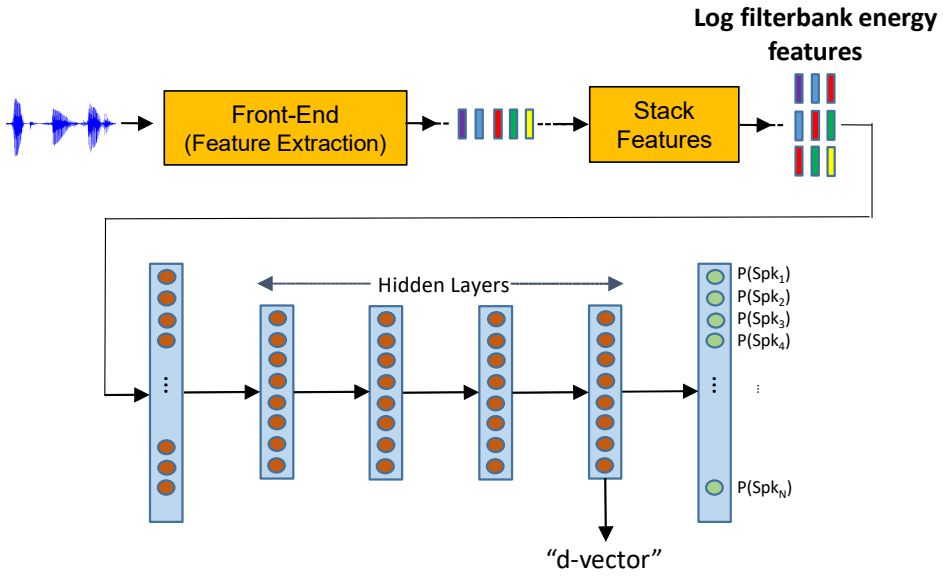


Figure 6.4: Illustration of the background speaker identification DNN used to extract a d-vectors for a speaker, by averaging the output activations from the last hidden layer.

Once the DNN is trained, they then use the outputs of the last hidden layer, to extract their new 'd-vector' speaker discriminative features.

Variani *et al* [23] experiment with the use of d-vectors for their text dependent ASV application, with the phrase "Ok Google". Their hypothesis, is that "d-vectors" are generalisable to represent unseen speakers during ASV classification. During speaker enrolment, the d-vector is averaged across all training utterances. Variani *et al* [23] used cosine scoring to compare the speaker similarity between train and test utterance d-vectors.

The configuration of their speaker identification DNN in [23] included four hidden layers, with 256 nodes per layer. The DNN used was specifically a *maxout* DNN with *dropout*, the intent being to try to minimise potential over-fitting issues with the small training set. Variani *et al* [23] used a pool size of two, with dropout restricted to the last two layers at 50%. They also used rectified linear units (ReLU) as the non-linear activation function on hidden nodes.

Random dropout is proposed in [89] as a form of regularisation, helping to prevent complex

co-adaptations in which network nodes are only helpful in the context of several other nodes, by randomly omitting certain hidden nodes during training. Hinton *et al* [89] also describe how dropout, can be viewed as an efficient form of model averaging using neural networks. During testing all the hidden nodes are used, effectively in a "mean network". However due to there being effectively more weights during testing compared with to training, the weights of the network have to be scaled down accordingly. For example if a dropout rate of 50% is used, then the weights have to be halved.

Maxout DNNs [90] were developed to minimise the model averaging approximation when using dropout. Variani *et al* [23] define maxout DNNs as differing from multilayer perceptrons (MLPs), by dividing the nodes in each hidden layer into non-overlapping groups. Each group then only generates a single activation output via max pooling i.e., the maximum output value is simply taken. By effectively adopting a smarter model averaging process, this can optimise the activation function for each hidden node during training with maxout.

The training criterion is the cross entropy loss, computed using a softmax activation function, defined in [91] as

$$l_{softmax} = -log\frac{exp\left(w_s^T a + b_s\right)}{\sum_{\tilde{s}} exp\left(w_{\tilde{s}}^T a + b_{\tilde{s}}\right)} \tag{6.4}$$

where a denotes the activation vector from the last hidden layer; w and b the learnt weight matrix and bias; and s the hypothesised speaker. The softmax normalisation is computed across all training speakers $\tilde{s}$.

Research into the use of d-vectors has since included work by Heigold *et al* [91], where they formulate a complete 'end-to-end' deep network for ASV, illustrated in Figure 6.5. They effectively train a deep network to try to verify whether or not, a test utterance was produced by an hypothesised speaker or a different speaker.

In a manner similar to Variani *et al* [23], they first enrol a speaker by computing their average d-vector across available training utterances. Verification again includes computing the cosine

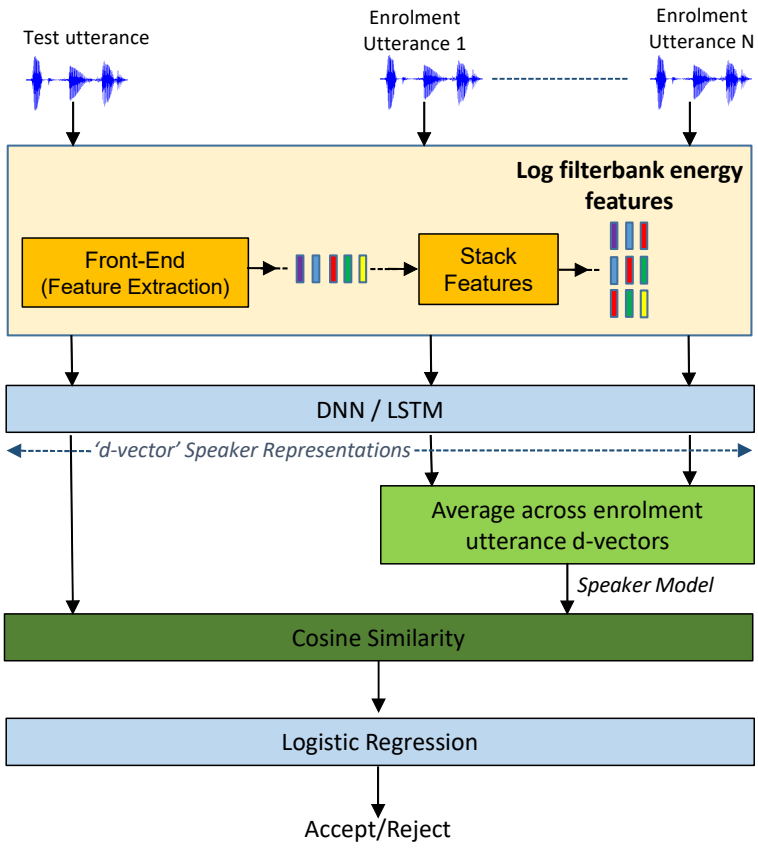


Figure 6.5: Illustration of the 'end-to-end' deep network proposed by Heigold *et al* [91] for ASV, building on the original d-vector work by Variani *et al* [23], with an additional logistic regression layer added to learn the cosine speaker model and test utterance d-vector distance scores, and the use of a time-sequence LSTM RNN in place of a DNN. Heigold *et al* [91] refer to d-vectors as speaker representations, which inspires the very recent work on 'speaker embeddings' by Snyder *et al* [26].

distance between a test utterance d-vector representation, and the average d-vector representation of an hypothesised speaker across all available training utterances. However, unlike with Variani *et al* [23], they add an additional logistic regression classification stage after the cosine similarity.

To train their complete end-to-end ASV network jointly with the DNN, they use the following expanded loss function (l_{e2e}) (where their derivations are expanded based on [92])

$$l_{e2e} = C(w,b) = -\sum_{i} \Big(y^{(i)} log(h_{w,b}(x^{(i)})) \; + \; (1 - y^{(i)}) log(1 - h_{w,b}(x^{(i)})) \Big) \tag{6.5}$$

where y represents the binary-valued labels ($y^{(i)} \epsilon \{false, true\}$) as to whether the hypothesised speaker is truly present or not; for the i'th computed cosine similarity score x, between a pair of d-vectors representing respectively a test utterance and an hypothesised speaker; and $h_{w,b}$ represents the logistic function of the form

$$h_{w,b}(x) = P(y = accept|x) = \frac{1}{1 + exp(-(w^T x) + b)}$$
$$P(y = reject|x) = 1 - P(y = accept|x) = 1 - h_{w,b}(x)$$

where the objective is to learn parameters w and b, so that $P(y = accept|x) = h_{w,b}(x)$ is large when x represents an *accept*, and small otherwise (so that $P(y = reject|x)$ is large).

Heigold *et al* [91] within their framework also investigate the use of two types of networks as opposed to one, namely a DNN similar to that used by Variani *et al* [23], and a LSTM neural network. The LSTM is limited to a single output, and accrues time-sequence information across individual utterance observation frames.

LSTM networks are a special form of recurrent neural networks (RNN), but modified to address the long-term dependency problems associated with RNNs, when modelling long sequences of data. It is therefore perhaps slightly intriguing the amount of benefit, specifically an LSTM network delivers over a conventional RNN, when they are only considering short "Ok Google" phrases.

6.2.2 Speaker Embeddings for End-to-End ASV

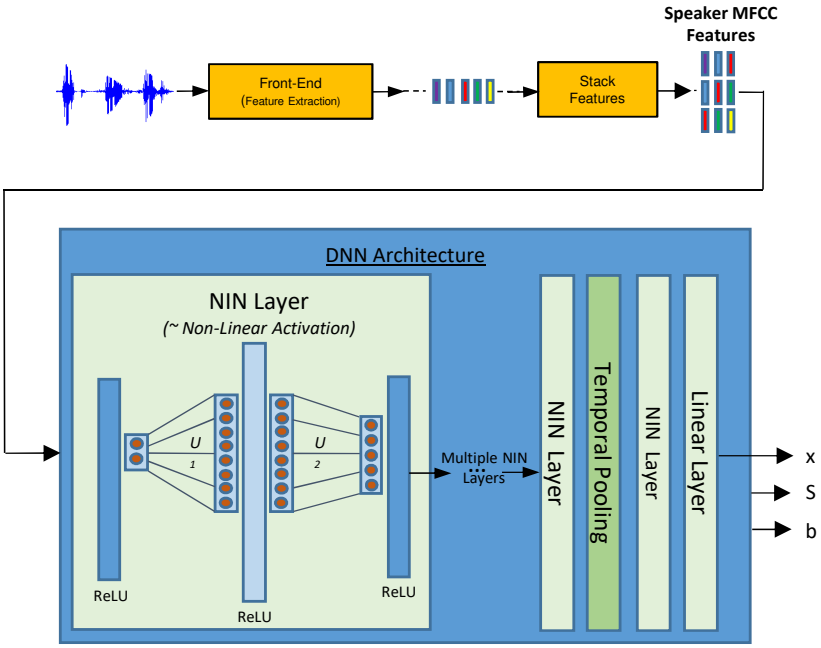
In work by Heigold *et al* [91], they use the term 'speaker representation' when they refer to the use of 'd-vector' type formulations, for speaker models. The concept of representing a speaker in some discriminative space notionally derived using 'end-to-end' deep learning, inspired the very recent work by Snyder *et al* [26] with 'speaker embeddings'. This is perhaps the second most prominent work of late in direct training of DNNs for ASV.

Snyder *et al* [26] comparably train a feed-forward DNN, with an objective function that operates on pairs of embeddings. The intention is to train a network that maximises the same speaker probability when two embeddings originate from the same speaker, and conversely minimises the same probability if the embeddings are from two different speakers.

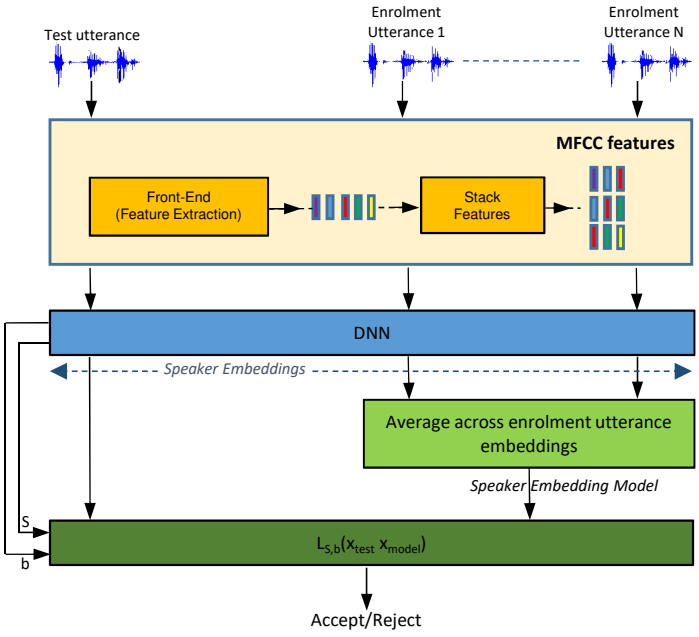
Motivated by the earlier work of Heigold *et al* [91], they develop a DNN framework that they state is capable of handling variable duration input through the use of a temporal pooling layer, and is developed for text-independent verification. Figure 6.6 illustrates their proposed 'end-to-end' ASV utilising 'speaker embeddings', with (a) their DNN architecture to map stacked MFCC features to an embedding vector, and (b) their ASV scoring process.

The DNN architecture shown in Figure 6.6(a), illustrates how their network consists of a number of network-in-network (NIN) [79] non-linear activations layers, and a temporal pooling layer. The final layer is linear, and produces the output embedding vector x, a symmetric weight matrix S, and a bias offset b. Both S and b are reported in [26] as constants, that are independent of the input observation features, but are required for their distance objective function.

The temporal pooling layer is described in [26] as aggregating the output of the preceding hidden layer over time, and computing its average and standard deviation. They then concatenate these statistics together, and append them to the input of a last hidden layer. The final layer is then linear, producing the final speaker embedding.



(a) DNN Architecture



(b) ASV Process

Figure 6.6: Illustration of the end-to-end ASV process using speaker embeddings by Snyder *et al* [26], with (a) the DNN architecture proposed that maps stacked MFCC to a 'speaker embedding' vector, and (b) the ASV scoring process. The objective function $L(x_{test}, x_{model})$ operates on pairs of embeddings, maximising same speaker embeddings, and conversely minimising pairs of embeddings from different speakers.

Ghahremani *et al* [93] describe the motivation for calculating the mean and standard deviation over a moving input window to the network, as expecting to capture long term variability effects in the speaker and channel. In regards therefore to handling variable duration input, the additional first and second order statistics, speculatively perhaps adds a form or regularisation. The DNN notionally can be viewed as a background model [23], of which the accrued mean and standard deviation possibly helps to align the frame level data inputs into the next NIN layer.

Figure 6.6(a) as stated, highlights the multiple NIN layers employed within their DNN. The NIN concept was originally developed by Lin *et al* [79], in pursuance of improving linear assumptions made around the max pooling process used with CNNs. Max pooling layers are used to both reduce the dimensionality, and aid in translational invariance. Lin *et al* [79] argue that the max pooling process is effectively too simplistic, and makes such assumptions that the latent concepts are linearly separable.

Lin *et al* [79] instead recommend substituting both the activation function of the convolutional layer, and the max pooling layer for another non-linear deep network, which they term a 'micro network' within the CNN. The micro-network therefore acts as the non-linear activation, whilst also encompassing the max pooling process. This effectively ties the mapping of the receptive field data through the convolutional filter, the non-linear activation, and max-pooling process, to the output vector into one joint process. Lin *et al* [79] investigate using conventional feed-forward DNNs as their 'micro-networks'. They term the overall structure a 'network in network' (NIN). Snyder *et al* [26] decide to adopt the NIN non-linearity concept for ASV, following its introduction to ASR by [93] for acoustic modelling from the signal domain using CNNs.

Figure 6.6(a) illustrates within their DNN architecture, the structure of a single NIN layer adopted from Ghahremani *et al* [93]. The modified NIN non-linearity presented in [93], is a new many-to-many non-linearity comprising they state of two block diagonal matrices. These repeated blocks are interleaved between layers of rectified linear units (ReLU). The transfor-

mation matrix block U_1, maps an input vector into some high dimensional space, where it is subsequently passed through a ReLU function. The matrix block U_1 is of dimension m x k, where m corresponds to the input size, and k the number of dimensions of the higher dimensional space. Ghahremani *et al* [93] define this higher dimensional space as the "NIN hidden dimension", which presumably at deeper layers within the network corresponds ideally to an effective space to discriminate speakers. If the micro neural network block parameters are shared across the NIN layer, then Ghahremani *et al* [93] asserts that each column of the block U_1 can be interpreted as a 1-d convolutional filter of size and shift equal to m, the size of input to the block U_1.

The second matrix transformation block U_2 then maps back down to a lower dimensional space, where it is followed by another ReLU function. They define the combination of the blocks U_1 and U_2, with the ReLUs, as a "micro neural network block". Their proposed NIN non-linearity thus resembles closely an autoencoder type structure, with many repetitions, but interleaved by ReLU non-linearities.

The intuition behind their DNN architecture, can be perhaps explained further in reference to the earlier works by Chen and Salman [94], who they cite use a similar constructed DNN. Snyder *et al* [26] describe their study, as investigating the training of DNNs on a speaker comparison task, producing frame-level features that capture speaker characteristics. According to Snyder *et al* [26], they then supposedly use these statistics to create single Gaussian component speaker models.

A key remark is made by Chen and Salman in [94], in that they consider their DNN to be essentially comprised of two deep autoencoders. The pair of autoencoders appear to conceptually derive a space at the code layer, with which minimises speaker utterance comparison errors. Chen and Salman [94] also propose that their DNN can be viewed as a regularised Siamese (RS) architecture, in which the data reconstruction of the autoencoders conceptually regularises

interference from non-speaker related information.

The multiple NIN autoencoding type micro-neural networks proposed by Snyder *et al* in [26], are possibly intended to instil similar qualities, in attempting to iteratively derive an effective speaker discriminant space through repeated application, of the autoencoder type micro-neural networks. However interestingly their overall DNN architecture, illustrated in Figure 6.6, can maybe be considered to be an autoencoder type layout without the eventual decoding back to the original input data, which is applied by Chen and Salman in [94] for regularisation. The speaker embedding output layer then possibly can viewed equivalently to the study of Chen and Salman [94], as the coding layer.

Snyder *et al* [26] report that they use a NIN configuration, consisting of 150 micro-neural networks, with an input d_i size of 600 nodes, a hidden layer d_h of 2000, and an output layer d_o of 3000.

Training of their DNN architecture can be explained via their ASV scoring process, as is illustrated in Figure 6.6(b) with the objective function $L_{S,b}(x_{test}, x_{model})$. Snyder *et al*'s [26] intention is to train a network, where the objective function maximises speaker embeddings from the same speaker, and conversely minimises if they are from different.

Taking a pair of embedding vectors x and y, they thus define their objective according to an error (E) probability function as follows

$$E = -\left(\sum_{(x,y)\epsilon P_{same}} ln\left(p(x,y)\right) + K \left[\sum_{(x,y)\epsilon P_{diff}} ln\left(1 - p(x,y)\right) \right] \right) \tag{6.6}$$

where K is a constant introduced in lieu of there usually being many more pairs of different speakers (P_{diff}), compared with the same (P_{same}),

The joint probability $p(x, y)$ they define by the logistics function

$$p(x, y) = \frac{1}{1 + exp^{-L_{S,b}(x,y)}} \tag{6.7}$$

where $L_{S,b}$ represents a form of linear distance similarity score, defined as

$$L_{S,b}(x, y) = x^T y - x^T S x - y^T S y + b \tag{6.8}$$

Intuitively if speaker embeddings x and y are sourced from the same speaker, then $L_{S,b}(x, y)$ will tend to zero. The goal of training is to learn the symmetric weight vector S and bias b, such that S scales optimally components $x^T S x$ and $y^T S y$ with offset adjustment b, with respect to the dot product $x^T y$. Snyder *et al* [26] also points out, that Equation 6.8 can alternatively be viewed as having a PLDA-like quality, implying presumably a factor analysis type characteristic, where S might be construed as a factor loading matrix.

Having defined their scoring scheme for training the DNN, Snyder *et al* [26]'s enrolment process for a speaker is otherwise analogous to [23, 91]. Speaker embedding vectors are simply averaged across all available utterances to produce the final embedding.

6.2.3 Performance of D-Vectors and Speaker Embeddings

This sub-section presents a review of the experimental findings by Variani *et al* [23], Heigold *et al* [91], and Snyder *et al* [26], the theory of which was discussed in the previous section.

Naturally both Variani *et al* [23] and Heigold *et al* [91], who are developing a text-dependent ASV system to operate on the fixed utterance "Ok Google", present findings following a similar experimental methodology. Snyder *et al* [26] however also chooses to focus on short utterances for a private application, presenting results on a large internal US telephony data set.

Corpora and Hyperparameter Settings

Variani *et al* [23] in their d-vector ASV experimentation, report that they used a total of 496 speakers for training their d-vector extraction neural network, with a remaining 150 speakers from their set used for evaluation. All speakers are uttering only the same phrase "Ok Google", in many sessions. During training of the DNN, they state that they used anywhere between 60 to 130 "Ok Google" utterances per speaker. For speaker enrolment, they used between 4 to 20 utterances per speaker. In the construction of their evaluation set, they set one out of every 150 trials as a true target trial, with approximately 12750 trials in total.

Their maxout DNN with dropout comprised of four hidden layers, with 256 nodes per layer, and a pool size of 2 per layer. They restricted the dropout to the last two layers, dropping 50% of activations. The output layer size corresponds to the number of training speakers, namely 496. For input into their DNN, they stacked 40 dimensional log filterbank energy features, over a context of 30 frames to the left and 10 to the right. No information is available on the frame size and increment used, as well as the recording format and collection conditions.

Heigold *et al* [91] in comparison to Variani *et al* [23], experiment on a significantly larger amount of data. Their corpus comprises of two DNN training sets, a small (*train_2M* utterances), and a large (*train_22M* utterances). The small contains a mere 4K speakers with over 500 utterances per speaker, whilst the large *train_22M* contains 80K speakers with over 150 utterances per speaker. Heigold *et al* [91] inform that they then augment this data, by artificially adding in car and cafeteria noise at multiple signal-to-noise ratios (SNRs) simulating far distance characteristics. They evaluate on 3K speakers, with between 1-9 utterances for enrolment per speaker, and 3-5 utterances for testing.

Due to presumably the larger data size in comparison to Variani *et al* [23], they use 504 node DNN layers instead of 256, with 4 layers in total, and again ReLU non-linearities except for the last layer that is linear. The first layer is also locally connected, with a patch size they define as

10 x 10, found in [95] to reduce their model size by 30% relative to their original fully connected DNN, and to similarly improve the relative ASV EER performance by 8%. For input into the DNN, like Variani *et al* [23], they use stacked 40 dimensional log-filterbanks (with they note some basic spectral subtraction), but concatenated over double the number of frames at 80 to match the average duration found for "OK Google" utterances.

Snyder *et al* [26] as stated, report findings on an internally collected corpus for a short duration application, where high accuracy and good calibration of scores across test conditions is required. Their full training dataset used for their DNN (train102K), equates to 102K speakers and more than 5.7K hours of US telephony speech at 8kHz. Their evaluation set comprised of 2419 speakers, with 2915 training utterances for training (equating to 1.21 utterances per speaker, with an average duration of 91s), and 2419 test utterances (with a duration between 1 to 92s). They constructed their test set, such that approximately 80% of the trials were non-target (false trial).

Their DNN comprised of four hidden NIN layers, followed by a temporal pooling layer, another NIN layer, and then a liner output layer. They state that they used 150 micro-neural networks within each NIN, each taking an input size of 600, a hidden layer of 2000 nodes, and an output layer of 3000 nodes. They used ReLUs non-linear activation functions with the NIN layers. For input into the DNN, they used a 20 dimensional MFCC feature vector with a frame-length of 25ms, reportedly mean-normalised over a sliding window of up to 3s.

Snyder *et al* [26] inform that they splice 9 x 20 MFCC frames together to create a 180 dimensional input vector. The size of this input vector however is less than the required minimum input dimension of the NIN, at 600 nodes. No details are provided as to how this is managed. Speculatively, their baseline i-vector system uses also 20 MFCCs, but appends delta and acceleration features to create a 60 dimensional vector. Whilst Snyder *et al* [26] explicitly state that they only splice 9 frames together, perhaps they in fact meant 10, if they accidentally had not

included the central frame. This would coincidentally generate the minimum 600 input node length required.

Review of Experimental Findings

Table 6.7 shows eleven pairs of %EER scores in total, comparing with and without test score normalisation (T-norm), taken from Variani *et al* [23] (V), Heigold *et al* [91] (H), and Snyder [26] (S).

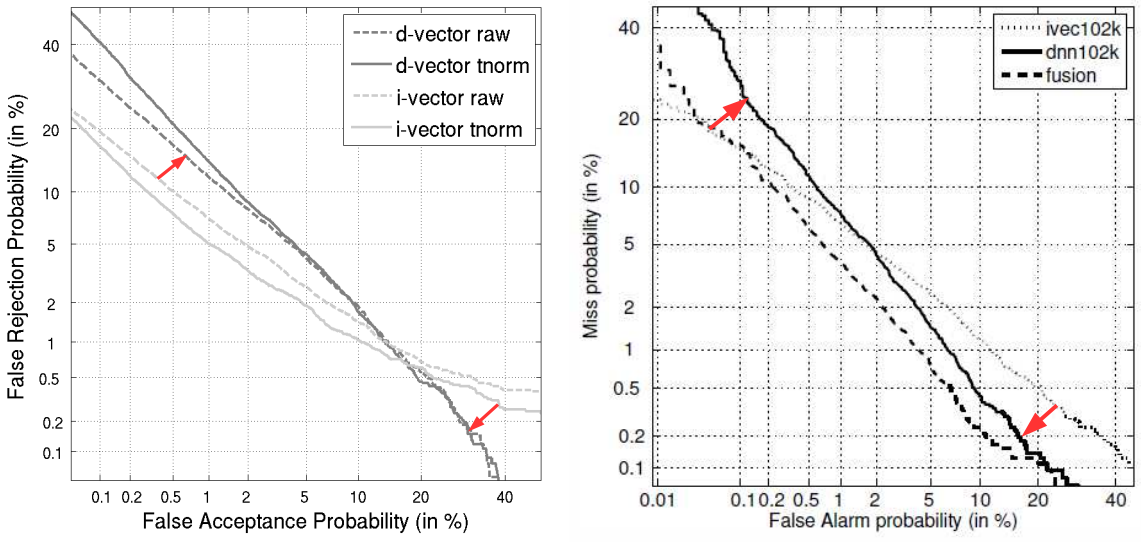
The two results taken from Variani *et al* [23], compare their d-vector ASV with the classic i-vector [3] approach, using cosine scoring, and with twenty "Ok Google" utterances per speaker used for enrolment. The two %EER scores show that they achieve comparable performance using d-vectors at 2.00%, compared with 1.21% for T-normalised i-vector scores. Variani *et al* [23] notably also find that applying T-norm to d-vector derived ASV scores, slightly degrades performance. They therefore choose in [23] to focus their efforts on the original non-normalised scores.

The marginal degradation in performance due to T-norm can be observed by the DET plot taken from [23], shown in Figure 6.8(a). The two pairs of profiles, for with and without T-norm, are derived using just 4 training utterances per speaker, instead of the 20 in Table 6.7. At 0.5% false alarm rate, the DET plot shows that applying T-norm increases the percentage miss from 17% ('d-vector raw') to approximately 20.5% ('d-vector tnorm'). However T-norm is shown to be beneficial for i-vectors, where at the same 0.5% false alarm rate, the percentage miss improves from 10% to approximately 7.5%.

The DET plot shown in Figure 6.8(a) also highlights, that the 'd-vector' ASV approach appears to perform much better than i-vectors at very high false alarm rates. For example at approximately 30% false alarm, both d-vector profiles correspond to a 0.2% miss, compared with the i-vector profile pair at 0.4% miss and greater (highlighted by the red arrows). Variani *et al* [23] in

Configuration	Data		%EER	
	UBM/DNN Training	*Enrol − Test (#utterances)*	*Original*	*T − Norm*
V: d-vector	-	20-1	2.00	-
V: i-vector	-	20-1	-	1.21
S: i-vector+PLDA	102K	full-full	2.4	-
S: i-vector+PLDA	102K	[1-20s]-20s	3.2	-
S: DNN	102K	[1-20s]-20s	2.6	-
S: Fusion	102K	[1-20s]-20s	1.9	-
S: Fusion	102K	[1-20s]-full	1.6	-
H: DNN,softmax	train_2M	[1-9]-[3-5]	3.86	3.32
H: DNN,softmax	train_22M	[1-9]-[3-5]	2.69	2.08
H: DNN,e-to-e	train_22M	[1-9]-[3-5]	2.04	2.14
H: LSTM,e-to-e	train_22M	[1-9]-[3-5]	1.36	-

Figure 6.7: Summary %EER scores with and without T-norm taken from Variani *et al* [23] (V), Heigold *et al* [91] (H), and Snyder *et al* [26] (S), comparing direct DNN training approaches for ASV: V=comparison between d-vector and classic i-vector with T-norm; S=comparisons between classic i-vectors with PLDA scoring, their speaker embedding ASV (DNN), and fusion, whilst varying the enrol and test durations, [1-20s] implies variable between 1 to 20s, and full implies a complete recording; H=d-vector type formulations using either a softmax or a complete end-to-end objective training criterion, substituting the DNN for a LSTM network, and varying the amount of DNN training data from 2M utterances (train_2M) to 22M (train_22M).



(a) Variani *et al* [23]

(b) Snyder *et al* [26]

Figure 6.8: DET ASV performance graphs highlighting potential score calibration issues with the current direct training of DNNs for ASV: (a) taken from Variani *et al* [23], with d-vectors using only 4 "Okay Google" utterances for enrolment; and (b) taken from Snyder *et al* [26] for pooled 10s, 20s and full recording test conditions, with 1-20s enrolment, and their 102K speaker set for training the i-vector UBM and DNN for speaker embedding extraction.

response to this phenomena, perform a separate histogram analysis of the unprocessed d-vector derived scores, concluding that the scores are heavy-tailed distributed instead of normal. They therefore speculate that a more sophisticated score normalisation method may be required for the d-vector based ASV.

In a rather unexpected manner, the potential score calibration issue found by Variani *et al* [23], is also observable in the more recently produced DET plots by Snyder *et al* [26]. An example plot is shown directly alongside in Figure 6.8(b), for pooled test 10s, 20s and full recording condition scores. Their DET plot shows how at the %EER point, whilst performance might be better using their speaker embedding approach, at the typical NIST-SRE [63] minimum cost operating point for large data sets - corresponding to low false alarm rates, performance using d-vectors is worse. The use therefore of the %EER, whilst a more easily interpretable benchmark point, should be tempered with the application operating point of interest.

Concerning the potential issue of score calibration, it perhaps can be hypothesised that the issue might be partially the result of the direct use of the network activation outputs, which will have invariably passsed through a non-linear activation function. In the case of d-vectors, Variani *et al* [23] take the outputs of the last hidden layer of the DNN, which will have passed through a ReLU activation function. Similarly, Snyder *et al* [26] apply a logistics function as part of their ASV scoring process. Further research is needed to understand this peculiarity, but the heavy-tailed score distribution might lend itself also to the length normalisation process applied by Garcia-Romero for PLDA [10].

Examining Snyder *et al*'s [26] results further, Table 6.7 lists five %EER scores. The scores suggest that when there is more enrolment and test data available, the i-vector with PLDA scoring performs better than the DNN approach. For example, when complete train and test recordings are used (full-full), the i-vector+PLDA achieves 2.4% EER. If however, the data is reduced to between 1 to 20s enrolment with 20s test ([1-20s]-20s), then the %EER increases to

3.2%. In comparison, the corresponding DNN speaker embedding approach achieves 2.6% EER. Fusion as expected leads to improvements, at 1.9% with 20s test, and 1.6% using the full test recordings.

In lieu of the promising low fusion score, and the potential issue of calibration, reference is made back to the DET plot in Figure 6.8(b). At the %EER point, there is a similar marked improvement from the DNN profile at approximately 3%, to the fusion profile at 2%EER. The fusion also aids it appears, in reducing the degradation in performance at low false alarm rates, where performance can be seen to tend towards following mostly the i-vector profile.

Analysing and constrasting next %EER results from Heigold *et al* [91], four pairs of scores are included in Table 6.7. The initial pair of 'DNN,softmax' scores, are derived using a softmax training criterion, similar to that used to originally train the d-vector speaker ID DNN proposed by Variani *et al* [23]. Heigold *et al* [91] compare training the DNN using the smaller 'train_2M' background set of 2M utterances, to utilising the much larger 'train_22M' set. The number of enrolment and test utterances is fixed at between 1-9 and 3-5 respectively.

Heigold *et al* [91] find, not unsurprisingly using more data to train their DNN leads to improvement, with for example the EER improving from 3.86% without T-norm to 2.69%. They also find improvement with applying T-norm, with for example the 'train_22M' DNN score improving from 2.69% to 2.08%. The improvement found by Heigold *et al* [91] is therefore in slight contradiction to the result of Variani *et al* [23], who found the opposite when applying T-norm. Further research is therefore needed to understand this, especially in view of the potential score calibration issues with Variani *et al* [23], and Snyder *et al* [26].

Table 6.7 next shows their %EER scores derived when applying their 'end-to-end' loss criterion, and substituting the DNN for an LSTM network. Applying the end-to-end loss criterion, they find improves the original unprocessed ASV scores from 2.69% to 2.04%. However almost no change is found with the T-norm pair, which degrades marginally from 2.08% to 2.14%. In

regard to using a LSTM network in place of a DNN, they find a substantial improvement from 2.04% to 1.36%, but at the expense of approximately ten times more calculations.

In summary the experimental findings of Variani *et al* [23], Heigold *et al* [91], and Snyder *et al* [26] highlight the potential of a directly trained DNN for ASV, with %EER results almost comparable if not better at times to the classic i-vector based ASV system. The issue with score calibration however highlights that further research is still very much needed, but a direct approach appears promising. It perhaps would be beneficial to isolate the interesting short duration problem in the first instance, to focus research into understanding the notionally single problem of directly training a DNN for ASV.

6.3 Preliminary Experiment - Speaker Identification using CNNs

In the previous sections, research exploring the use of deep learning, both indirectly via a pre-trained DNN for ASR, and directly was reviewed. It can be surmised from the review made, that the degree of research into understanding the direct approach, where the classes are speaker related, has until very recently received much less attention. This is perhaps due to the often limited amounts of enrolment data available per speaker [22].

Direct training of DNNs for ASV however notionally directs the DNN to discover new, potentially more robust features or representations from which to discriminate speakers. An apparent emerging term to such effect, is a 'speaker embedding', coined by Snyder *et al* [26]. The use of deep networks for the potential discovery of new robust features or representations, therefore motivates the preliminary experiment presented here, with the use of convolutional neural networks (CNNs) for speaker identification.

More widely the use of convolutional neural networks (CNN) has led to significant advancements in recognition performance, particularly in computer vision with the work of Krizhevsky *et al* [19] in the 2012 ImageNet recognition competition. CNNs were described earlier in Section 6.1.2, where they can be considered as a biologically inspired variation of multi-layer perceptrons (DNNs) [76].

CNNs are intentionally constructed, such that they are sensitive to specific localised patterns, but also invariant to their precise location. Combining then these localised patterns or features in higher order or deeper layers, conceptually allows more complex-higher level semantic level features to be subsequently discovered [74]. Notionally, it can be considered then that Krizhevsky *et al* [19] are effectively using their five CNNs layers, as a data-driven feature extraction, directed by the subsequent fully connected regular DNN layers.

In the preliminary experiment presented, the data-driven CNN framework of Krizhevsky *et*

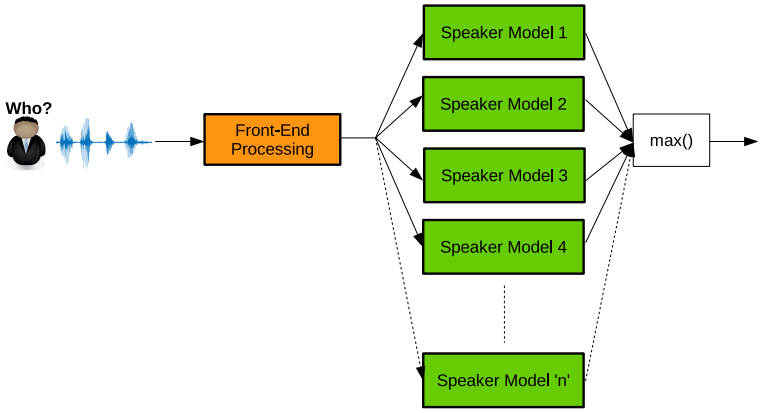


Figure 6.9: Proposed closed speaker identification task.

al [19] for image object recognition, is followed with an analogous speaker identification (ID) task. Figure 6.9 illustrates, where given a speech utterance, the question is who out of 'N' possible speakers produced it. Similar to ASV, the speech sample is first passed through a front-end process to extract features, before then being compared against the 'N' speaker models. A maximum can then be taken to decide who out of the 'N' speakers is most probable.

The goal of the experiment presented, is thus to establish whether or not a CNN can be trained successfully for speaker identification. It is hoped that this work might eventually lead to the development of more robust features or speaker representations, effectively analogous to the current research in 'speaker embeddings' [26].

6.3.1 Deep Network Architecture and Component Settings

Figure 6.10 illustrates the CNN-DNN architecture used, based on Google TensorFlow's MNIST CNN tutorial [96].

The complete network comprises of two CNN layers, followed by a fully connected conventional DNN layer, and a final output (readout in TensorFlow terminology) layer. The first CNN layer is expanded to illustrate the two inner processes of convolution with non-linear activation, and

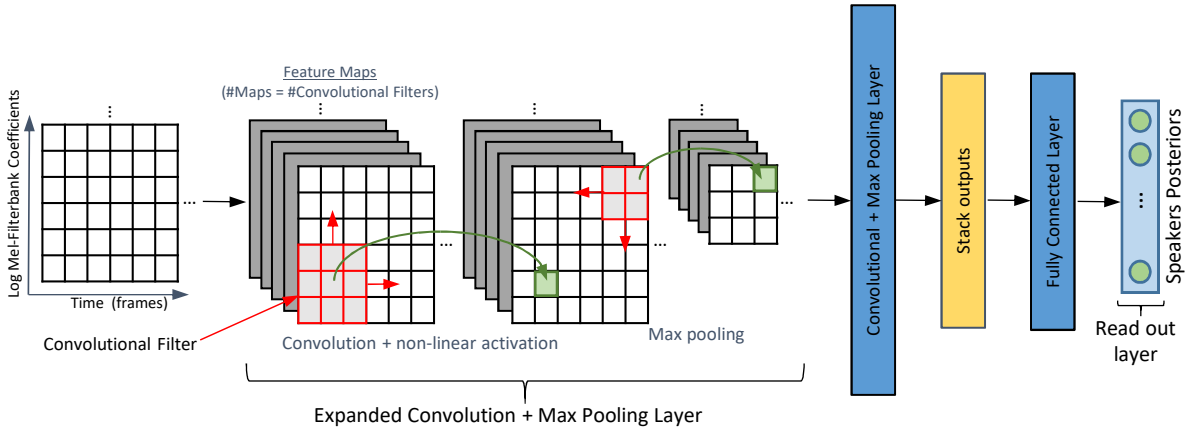


Figure 6.10: Illustration of the CNN-DNN network architecture used for speaker identification. The first CNN layer is expanded for illustrative purposes, comprising of a convolution+non-linear activation and a max pooling process. The single 3 x 3 example convolutional filter shown for display only, is smaller than the filter used during the experiments, which was 5 x 5 for both CNN layers. In total, 32 filters were used in the first CNN layer, and 64 in the second, generating the equivalent number of respective feature maps. A max pooling group size of 2 x 2 was used for both CNN layers.

max pooling. It should be noted that the single example convolution filter shown in Figure 6.10 is for display purpose only, and is smaller than was used in practice. A slightly larger size of 5 x 5 was used for both layers, with a max pool group size of 2 x 2.

The input image into the network comprises of the log Mel filterbank output coefficients, based on the works of [23,84]. The filterbank coefficients were also passed through a linear discriminant analysis (LDA) transform, based on [84]. The LDA transform matrix was estimated using all male training data from the NIST-SRE 2005 'conv4w' training set.

In total 26 filterbank coefficients were calculated, from a 32ms frame size with 16ms increment. In order to aid the twice halving of the dimension size from the max pooling processes, the last two filterbank coefficients were dropped, giving a length of 24. Each image also underwent mean and standard deviation normalisation, before being scaled to a comparable [0,1] scale. The mean and standard deviation statistics, and [0,1] minimum-maximum scaling parameters were derived

Component	*Values*
Front-End	26 mel spaced filters, 32ms window with 16ms increment
Input Filterbank	24 filterbank coeff's x 32 time frames
Conv' Layer 1	5 x 5 x 32
Max' Pooling	2 x 2 with stride 1
Conv' Layer 2	5 x 5 x 64
Max' Pooling	2 x 2 with stride 1
#Fully Connected Nodes	1024
Activation function	ReLU

Table 6.4: DNN-CNN and front-end log-mel filterbank calculation settings. Rectified linear units (ReLU) were used as the non-linear activation function for all three layers.

from across all the enrolment and test data.

Figure 6.10 illustrates how the linear filter is convolved across the filterbank spectral image producing a corresponding feature map. Often many filters are used, producing the respective number of filter maps. In total 32 filters were used in the first layer, and 64 in the second. The filters were stepped across the filterbank image with a stride of 1 coefficient. To also maintain the dimension size, the log spectral filterbank images were padded with zeros.

Max pooling is then applied, which reduces the dimensionality, aiding in computational burden at higher layers and translational invariance. Figure 6.10 illustrates how with a group size of 2 x 2, the filter maps are halved in dimension (the stride size of max pooling group is 1 coefficient). The output filtermaps of the max pooling are subsequently fed into a second convolutional layer with max pooling, the outputs of which are stacked, and processed by a fully connected regular DNN layer. The fully connected layer has 1024 nodes.

Table 6.4 lists the component settings chosen for reference. Rectified linear units (ReLU) were used as the non-linear activation function for all three layers in the network.

Model	Train 1	Train 2	Test
M7029	jiio.sph:B	jhfb.sph:A	jebn:A (test 1)
M7040	jewi.sph:B	jhmt.sph:A	jaxv:A (test 2)
M7845	jico.sph:B	jacj.sph:B	-
M8611	jhpl.sph:B	jekq.sph:B	-

Table 6.5: Initial test models and training/test speech data taken from the NIST-SRE 2005 '3conv4w' training set [58] . The M7845 test recording was not used due to the recording only containing half the amount of speech to the other two recordings. The M8611 test recording was found to contain no speech at all.

6.3.2 Experimental Protocol

Google's open source TensorFlow MNIST tutorial and software development kit (SDK) [96] was used as the basis for implementing the CNN-DNN described. The CNN-DNN was trained to initially just learn 4 speaker models from the NIST-SRE 2005 [58] '3conv4w-1conv4w' male set, before be being subsequently increased up to 84 models using training data from the NIST-SRE 2004 [57] '3conv4w' male set.

The NIST-SRE 2005 '3conv4w-1conv4w' male set [58] consists of 3 recordings per speaker model. For the preliminary experiment presented, four models were chosen, which are listed in Table 6.5. Two of the recordings were used for training, whilst the third was kept for testing. Unfortunately the third recording for the M7845 model contained only half the amount of speech to the M7029 and M7040 recordings, and the third recording for M8611 contained no speech at all. Due to time restrictions, there was not sufficient time to substitute these models, but they do nevertheless aid in testing for false alarm events.

The initial four male speaker models from the NIST-SRE 2005 '3conv4w-1conv4w' training set were subsequently expanded upon, with models from the NIST-SRE 2004 [57] '3conv4w' male set. Initially, 40 speaker models were appended before being increased to 80, resulting in 84 in total.

To minimise further complexity, the filterbank images were kept fixed at a size of 24 x 32 coefficients, corresponding to the number of filterbank coefficients, and time frames respectively. The duration of all enrolment and test recordings was fixed at 3200 voice activity detected (VAD) frames, corresponding to 51s (31999*16ms increment + 32ms frame duration) of speech per recording. In respect that each filterbank image is fixed at 32 time frames, this produces 100 filterbank images per audio recording. Prior to training, the images were first randomised to help stabilise the training of the network.

A fixed total number of 32K iterations (referred to as 'steps' in TensorFlow) was set for training the network. At each iteration (step), the first 50 images were drawn (the batch size) from the enrolment collection of filterbank images, and used to train the network. The images were not replaced back into the collection. When the collection of training filterbank images was empty (corresponding to 1 epoch), the training images were re-randomised and populated back into the collection. The training cycle was then repeated, until 32K iterations (steps) was reached.

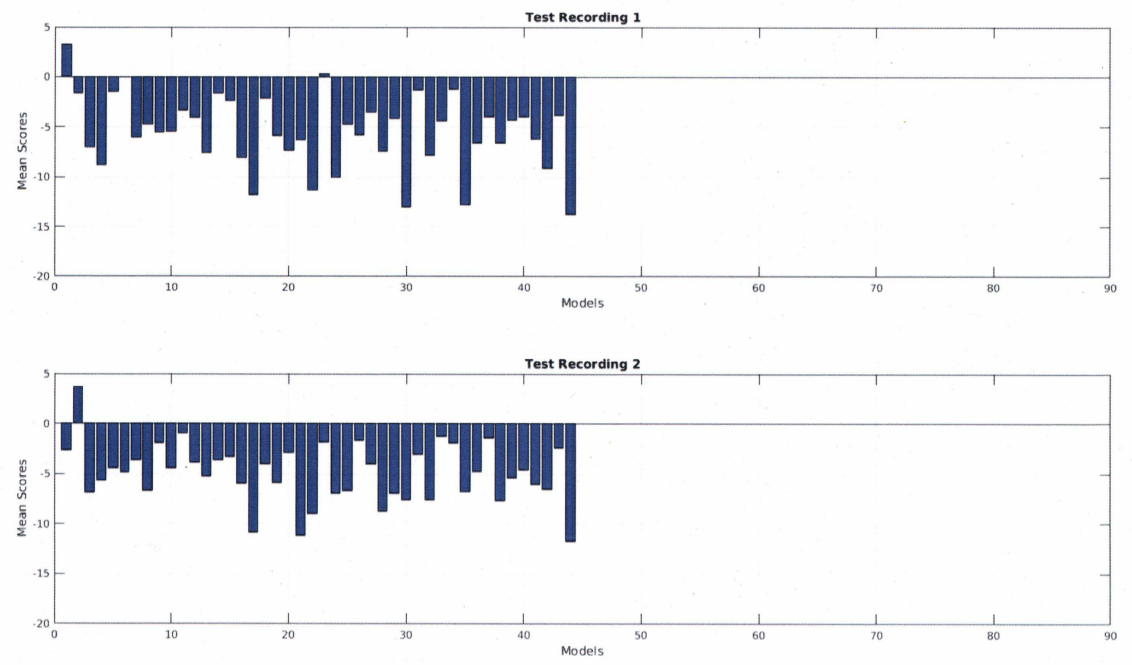
With 44 speaker models, 87 training audio recordings were used, equating to 8700 training images. The number of epochs (complete training repetitions across all the 8700 training images) is therefore (50 batches x 32000 steps) / 8700 images = 184. For 84 speaker models, 166 audio recordings were used, which equates to 96 epochs. The high number of epochs or repetitions over the complete training data, perhaps may lead to future over-tuning issues; however, the motivation here was simply to establish whether or not a CNN network could be trained successfully for speaker identification.

The cross entropy loss function was used as the training criterion, with the output speaker posteriors computed using a softmax.

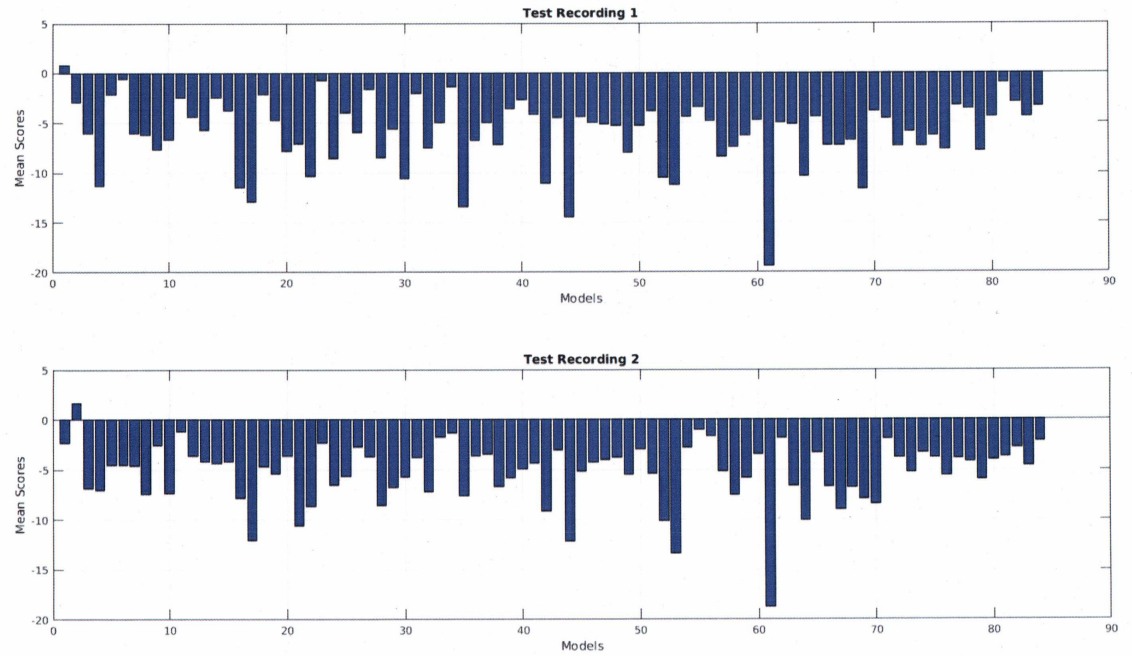
6.3.3 Results and Analysis

Figure 6.11 shows the output speaker posterior scores from the network for the two test recording (jebn:A, jaxv:A) respectively, with a network of 44 models (a) and 88 models (b) respectively. The posterior scores correspond to the mean taken across all the test utterance time frames.

The two pairs of plots show that the CNN-DNN network is able to correctly identify the two speaker models, namely M7029 (model 1) and M7040 (model 2) when processed against the two test utterances respectively. The use of 84 speaker models is perhaps at the limit for the size of network specified, evident by the lower mean posterior scores obtained. Further experimentation is needed, but the preliminary objective to determine whether or not a CNN network can be trained successfully for speaker identification, at least for a very small trial appears true.



(a) 44 models



(b) 84 models

Figure 6.11: Activation scores averaged across the two respective test recordings (jebn:A, jaxv:A), for the network with 44 models (a), and 88 models (b). The two pairs of plots highlight how the network correctly identifies speakers M7029 (model 1), and M7040 (model 2) correctly.

6.4 Deep Learning Conclusions

In this chapter, a study into deep learning approaches within ASV was made, motivated by the potential to discover new robust speaker representations and features.

A literature review was first presented of research into deep learning to date in ASV, which highlighted two prime approaches: indirectly via a pre-learnt DNN for ASR; and the direct training of the DNN, where the output classes are speaker related. The review highlighted how to date, the indirect approach appears to have received more research attention than the direct form. This was concluded as being most likely the result of the often limited amounts of enrolment data available per speaker [22]. However of late, it would appear that the direct training of DNNs is beginning to see an increase in interest, with such works including [26, 87, 88, 91].

The indirect approach using a DNN pre-trained for ASR was originally proposed by Lei *et al* [22]. In their framework, they propose that the classes of the UBM are defined by phonetic senone states, instead of corresponding to the unsupervised GMM component indices. The senone posterior state alignments are calculated using the DNN trained for ASR.

Lei *et al* [22] define senones as the leaves of the ASR decision tree, which therefore correspond to the underlying tied triphone states of the ASR HMM. By defining the UBM classes as such, they argue allows the ASV process to be aligned on the phonetic content present in utterances. They refer to their alternate framework as 'DNN/i-vector' in [56].

Two practical ASV methodologies appear to have emerged from the indirect approach, with the first as described. The second involves bottleneck features extracted directly from the network, from a layer with a reduced number of nodes. The performance of both methods was assessed in great detail by McLaren *et al* in [56]. They show on NIST-SRE 2012 [63] data, that using bottleneck features or the DNN/i-vector framework on telephony English conditions, gave on

average a 30% relative improvement over their UBM(MFCC)/i-vector baseline. They further show that fusion between the acoustic MFCC derived i-vectors, and the indirect phonetic ASR schemes, leads to even greater gains with a relative improvement in the order of 45%. The significant gains found with fusion, leads Mclaren *et al* [56] to conclude that the information from conventional acoustic MFCC trained i-vectors is orthogonal, to the information supplied by the indirect phonetic ASR-DNN approach.

McLaren *et al* [56] find similar patterns of improvement with 'microphone' derived speech, but highlight that DNN is perhaps more susceptible to unwanted channel variabilities. Somewhat also strikingly, their cost scores appear generally appear lower for the microphone speech, compared with their telephony results. Very little detail is provided about the recordings, other than that a higher bit depth of 16 bits is used with microphone speech, compared with 8 bits for telephony.

Further such research into indirect ASR schemes, included the substitution of the DNN for a CNN for robustness in noisy conditions. McLaren *et al* [25] present findings on the large radio re-transmission RATS [83] evaluation corpus. Comparing again to a classic i-vector baseline, they find that in this case the use of the CNN/i-vector framework alone, does not lead to significant gains over a more conventionally trained i-vector based system on acoustic type features. However the fusion of both leads to significant performance improvements, at around the more conventional NIST-SRE [63] minimum cost operating point, corresponding to a large speech data processing system. They find at 1.5% false alarm rate, approximately a 10% improvement in miss performance, on an evaluation set comprising of 5.8M impostor and 5.8K target trials.

The use of CNNs it appears was also explored by Snyder *et al* [85], in the form of time delay neural networks (TDNN). Their use of TDNNs was inspired by the major gains found by Peddinti *et al* [84] in ASR, with on average a 4.3% improvement in word error rate across several large vocabulary tasks. Snyder *et al* [85] find similar relative improvements, with in the order of 50%

EER.

A review was then made of research exploring the direct training of the DNN for ASV. The early works of Variani *et al* [23] on d-vectors, and the very recent work by Snyder *et al* [26], are probably the two most prominent works in this context. Both methods essentially can be viewed, as attempting to derive an effective space to represent and discriminate speakers within, derived through the direct training of DNNs. To this effect, Snyder *et al* [26] coins the term 'speaker embedding', referring to the mapping of a speech utterance into a speaker embedding vector.

Research into the direct training of an 'end-to-end' deep learnt ASV system appears promising, if perhaps at a relatively early stage of research. Snyder *et al* [26] find that under short duration conditions, their 'end-to-end' trained DNN performs better than their i-vector+PLDA based ASV system, at 2.6% EER compared with 3.2%. This equates to a 20% improvement. They find that fusion improves this to 1.6% EER, a 50% relative improvement. Heigold *et al* [91], following the work of Variani *et al* [23], also investigate the use of LSTM to capture time-sequence information, and reporting similar performance gains.

However the DET plots from both Snyder *et al* [26], and Variani *et al* [23] both exhibit potential score calibration issues. At the usual NIST-SRE [63] minimum cost operating point, corresponding to a low false alarm rate, the miss performance degrades significantly compared with the use of i-vectors. Conversely, at high false alarm rates, the direct trained DNN ASV systems perform better than the i-vector based schemes. The DET profiles in effect, appear to be tilted clockwise relative to the i-vector profiles. Despite this potential calibration issue, research into direct training of DNNs for ASV appears promising. It perhaps would also be interesting to compare performances over longer durations, attempting to focus efforts thereby on the successful training of the DNN.

Last a preliminary experiment was presented, on directly training a CNN for speaker identifica-

tion (ID). The use of CNNs as a data-driven feature extractor, was motivated by the significant breakthrough in computer vision by [19]. The experiment demonstrated that it is possible to train a small CNN for speaker identification, discriminating two speakers from out of a possible 82 confusable speakers, for a very small trial. Much further research is needed, but it is hoped that this might lead to further understanding or discovery of new robust speaker representations.

Chapter 7

Conclusions and Recommendations

This thesis opened by stating how performance on English telephony conditions, can be considered in many respects to be at, or if not very close to the upper achievable limit. However despite this exceptional performance achieved, ASV systems still remain somewhat susceptible to sources of unseen variability, and especially in challenging degraded conditions [12]. It was hypothesised, that if ASV systems are to become truly robust to unseen or unwanted variabilities, then more robust features beyond cepstra are likely needed. Much of the research over the last two decades, has been focused instead on developing speaker models and classification techniques capable of learning large amounts of data.

With this underlying motivation in mind, an extensive review was first made of the developments from the mid-'90s with the early work on GMM-UBMs by Reynolds *et al* [1,16], up to the eventual development of i-vectors by Dehak *et al* [3], and the adoption of PLDA scoring [41,42] in around 2010. A thorough explanation was made in particular of i-vectors, including the adoption of the underlying UBM centralised MAP adaptation concept from Reynolds [16], for the compensation of the often limited amounts of speaker enrolment data, and the derivation of the EM training process for the total variability matrix.

In reviewing the subsequent work by Kenny on HT-PLDA [41] scoring of i-vectors, attention was drawn to his observations made, concerning the linear statistical independence assumption made between speaker and channel variability. Kenny *et al* interestingly in [41] argues that this assumption is potentially both flawed, and still not well understood. He hypothesises that the channel variability is speaker dependent, based on scatter plots of the first two i-

vector components produced by Tang *et al* [53], which are highly directional with respect to the speakers. This he argues, indicates that there is not a consistent axis of session variability between speakers.

Whilst Kenny's observations are potentially over-stated, this does partially corroborate the opening observations made, that there still lacks the discovery or understanding of a truly robust set of speaker features or representation, beyond that of cepstra. However if a true physical understanding of the relationship between speaker and channel variabilities is to be achieved, then perhaps experimental speech corpora need to be made under much more tightly controlled conditions, and labelled accordingly. This invariably would be more costly and time consuming, and would also come at the risk of the results found not being generalisable to wider conditions.

Following the review, the thesis then presented a series of performance tuning experiments on GMM-UBM and i-vector based ASV systems. Specific implementation details are sometimes lacking from published works, which can make it difficult to repeat leading findings, and to gauge the limits of these approaches. The experiments were conducted using a fixed reference telephony test set (NIST-SRE 2005 '3conv4w-1conv4w' [58]), to allow for comparison throughout.

The experiments highlighted in particular how effectively adding more background data, without due regard to the amount of potential variability it contains, can lead if not careful to a degradation in performance. This susceptibility was particularly evident with early the GMM-UBM ASV, where adding Switchboard II data [14,59] to the UBM was found to degrade performance.

An analysis was made of the cumulative LLR score distributions, which showed two noticeable patterns. The first was a very noticeable shift in scores with respect to adding more Switchboard II data to the UBM, indicating that score normalisation should be applied (Z-Norm and or T-Norm [29]). The second, were subtle score variations, which it was concluded was likely attributed to the sensitivity of the GMM-UBM ASV to unwanted variabilities. The training of

the UBM for example, does not make any allowance for unwanted speaker and channel variations, applying effectively a 'blind' maximum likelihood EM fit of the Gaussian distributions.

Somewhat surprisingly, the training of the total variability matrix (T-matrix), required for extraction of i-vectors from utterances, was also found to be slightly susceptible to unwanted variabilities. Often whole evaluation datasets are used to train the T-matrix for robustness [3,12], with notionally little regard to the amount of benefit each set adds to performance. It was found that adding the Fisher English-Part 2 [97] degraded performance marginally, from 4.61% EER to 4.80% EER with 2048 Gaussian components. From a brief analysis, it was speculated that this might have been due to the increased number of speakers in the Fisher English-Part 2 at 562, compared with 418 in the Fisher English-Part 1 [98], coupled with also that the total number of recordings remains approximately the same.

A fairly extensive study was then made into the use of deep learning with ASV, which can be viewed as the data-driven discovery of new robust features or representations for classification. The chapter began by presenting a literature review of deep learning work to date in ASV, before then concluding with a preliminary experiment on training a CNN directly for speaker identification. The literature review highlighted effectively two main approaches adopted within ASV, the first by indirectly using a pre-trained DNN for ASR, and the second being the direct training with speaker related output DNN classes.

The review also highlighted how the direct approach until of recent, has received less attention. It was concluded that this was perhaps due to the often limited amounts of speaker enrolment data available, making it more challenging [99]. However it would appear, that very recently the direct training approach is receiving an increased amount of interest, with works including the likes of [23, 26, 87, 88, 91].

The indirect ASR-DNN scheme is described by Lei *et al* [22], where they propose defining the UBM classes by phonetic senone states intead, which correspond to the underlying tied-triphone

states of an ASR-HMM. They recommend accurate calculation of the senone posteriors, by the use of a DNN trained for ASR. They conceptually describe their 'DNN/i-vector' framework as guiding the ASV process, such that speakers utterances are compared with respect to the phonetic content present.

Findings by McLaren *et al* [56] and Snyder *et al* [85] indicate gains by as much as 50% EER on NIST-SRE US English data can be achieved. McLaren *et al* in [25] also finds similar significant gains in noisy conditions, using CNNs instead of DNNs and at low false alarm rates, where they achieve a 10% improvement in miss performance on a test data set comprising of more than 5.8M trials.

The indirect ASR-DNN approach therefore would seem to be well established, provided however there is sufficient data to train the ASR in the required ASV operating domain. In view of understanding further speaker and channel variability, and the fusion of both acoustic and phonetic information, it perhaps might be interesting to see if it possible to develop a more forensic type speaker recognition system in the future.

Direct DNN training schemes seem to comprise of two main works, that of originally by Variani *et al* [23] with 'd-vectors', and very recently by Snyder *et al* [26] with 'speaker embeddings'. Both methods are effectively attempting to discover through the use of deep learning, a new robust space from which to represent and discriminate speakers. Performance appears promising if at quite an early stage, with Snyder *et al* [26] finding that they achieve a slightly better percentage EER with shorter durations compared with using i-vectors and PLDA scoring. However analysis of their DET plots indicates a potential issue with score calibration.

Last, a preliminary experiment was presented, investigating whether or not a CNN could be directly trained successfully for a very small speaker identification task. A successful result was achieved. The use of CNNs was motivated by the major performance gains found in image object recognition by Krizhevsky *et al* [19], where they effectively use CNNs as data-driven

feature extraction components in their network. Much further experimentation is needed, but it is hoped that this might lead to the new more robust speaker representations, or in the very least a better understanding of the allusive relationship between speaker and channel variabilities highlighted.

Bibliography

[1] Douglas A Reynolds and Richard Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Trans. on Speech and Audio Processing*, 3(1):72–83, 1995.

[2] Patrick Kenny, Gilles Boulianne, and Pierre Dumouchel. Eigenvoice modeling with sparse training data. *Speech and Audio Processing, IEEE Transactions on*, 13(3):345–354, 2005.

[3] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(4):788–798, 2011.

[4] NIST. Nist speaker recognition evaluations. `http://www.itl.nist.gov/iad/mig/speaker-recognition`. Accessed: 2016-11-01.

[5] Luciana Ferrer, Elizabeth Shriberg, Sachin S Kajarekar, Andreas Stolcke, Kemal Sonmez, Anand Venkataraman, and Harry Bratt. The contribution of cepstral and stylistic features to sri's 2005 nist speaker recognition evaluation system. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 1, pages I–I. IEEE, 2006.

[6] NIST. 2012 nist speaker recognition evaluation results. `http://www.nist.gov/itl/iad/mig/sre12-results`. Accessed: 2016-11-20.

[7] NIST. Nist 2016 speaker recognition evaluation. `http://www.itl.nist.gov/iad/mig/speaker-recognition-evaluation-2016`. Access: 2016-11-20.

[8] Yosef Solewicz, Andy Hatch, and Other ISCI team members. Sris nist 2006 speaker recognition evaluation system. In *NIST SRE Workshop*. NIST, 2006.

[9] NIST. 2008 nist speaker recognition evaluation results. `http://www.nist.gov/itl/iad/mig/sre08_results`. Accessed: 2016-11-20.

[10] Daniel Garcia-Romero and Carol Y Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *Proc. of Interspeech*, pages 249–252, 2011.

[11] Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Trans. on ASSP*, 29:254–272, 1981.

[12] D. Garcia-Romero. *Robust speaker recognition based on latent variable models*. PhD thesis, Department of Electrical and Computer Engineering, University of Mayland, 2012.

[13] Massimiliano Todisco, Delgado Héctor, and Nicholas Evans. A new feature for automatic speaker verification anti-spoofing constant q cepstral coefficients. In *Proc. of Speaker Odyssey*, pages 283–289. ISCA, 2016.

[14] David Graff, Alexandra Canavan, and George Zipperlen. Switchboard-2 phase i ldc98s75. DVD, 1998.

[15] Tomi Kinnunen and Haizhou Li. An overview of text-independent speaker recognition: from features to supervectors. *Speech communication*, 52(1):12–40, 2010.

[16] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1):19–41, 2000.

[17] Benoît GB Fauve, Driss Matrouf, Nicolas Scheffer, J-F Bonastre, and John SD Mason. State-of-the-art performance in text-independent speaker verification through open-source software. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(7):1960–1968, 2007.

[18] Y LeClun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, E. Imagenet classification with deep convolutional neural networks. In *Adv. in Neural Information Processing Systems*, pages 1097–1105, 2012.

[20] Li Deng, Geoffrey Hintern, and Brian Kingsbury. New types of deep neural networks for speech recognition and related applications: an overview. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8599–8603, 2013.

[21] Thong Luong, Cho Kyunghyun, and D. Manning, Christopher. Neural machine translation. In *Tutorial for Association of Computational Linguistics (ACL) Conference*. ACL, 2016.

[22] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell McLaren. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *Proc. of International Conference on Acoustic, Speech and Signal Processing*, pages 1695–1699. IEEE, 2014.

[23] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 4052–4056. IEEE, 2014.

[24] Yun Lei, Luciana Ferrer, Mitchell McLaren, and Nicolas Scheffer. A deep neural network speaker verification system targeting microphone speech. In *Proc. of Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[25] Mitchell McLaren, Yun Lei, Nicolas Scheffer, and Luciana Ferrer. Application of convolutional neural networks to speaker recognition in noisy conditions. In *Proc. of Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[26] David Snyder, Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel, and Sanjeev Khudanpur. Deep neural network-based speaker embeddings for end-to-end speaker verification. In *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pages 165–170. IEEE, 2016.

[27] Barclays. Banking on the power of speech. `https://wealth.barclays.com/en_gb/home/international-banking/insight-research/manage-your-money/banking-on-the-power-of-speech.html`. Accessed: 2016-11-25.

[28] First Direct. Phone banking - what is voice id security? `http://www1.firstdirect.com/1/2/banking/ways-to-bank/telephone-banking#voice-id-security`. Accessed: 2016-11-25.

[29] Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meignier, Teva Merlin, Javier Ortega-García, Dijana Petrovska-Delacrétaz, and Douglas A Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Applied Signal Processing*, 2004:430–451, 2004.

[30] Michael J Carey, Eluned S Parris, and J Bridle. A speaker verification system using alpha-nets. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 397–400. IEEE, 1991.

[31] Roland Auckenthaler, Eluned S Parris, and Michael J Carey. Improving a gmm speaker verification system by phonetic weighting. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 1, pages 313–316. IEEE, 1999.

[32] Eluned S Parris and Michael J Carey. Discriminative phonemes for speaker identification. In *Third International Conference on Spoken Language Processing*, 1994.

[33] Steven Davis and Paul Mermelstein. Comparison of parametric representations for mono-

syllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[34] Hiroshi Shimodaira and Steve Renals. Speech signal analysis. *University of Edinburgh: Automatic Speech Recognition - ASR Lectures 2&3*, 2017.

[35] Douglas Reynolds. Correspondence: Experimental evaluation of features for robust speaker identification. *IEEE Trans. on Speech and Audio Processing*, 2(4):639–643, 1994.

[36] Mike Brookes. Voicebox: Speech processing toolbox for matlab. `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`. Accessed: 2016-12-01.

[37] J-L Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE transactions on speech and audio processing*, 2(2):291–298, 1994.

[38] Microsoft. Msr identity toolbox (without binaries). `http://research.microsoft.com/en-us/downloads/a6262fec-03a7-4060-a08c-0b0d037a3f5b/`. Accessed: 2015-03-31.

[39] K P Li and P E Porter. Normalizations and selection of speech segments for speaker recognition scoring. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 1, pages 595–598, 1988.

[40] R Auckenthaler, M Carey, and H Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1), 2000.

[41] Patrick Kenny. Bayesian speaker verification with heavy-tailed priors. In *Proc. of Odyssey*, page 14, 2010.

[42] Simon JD Prince and James H Elder. Probabilistic linear discriminant analysis for inferences about identity. In *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

[43] Patrick Kenny, Vishwa Gupta, Themos Stafylakis, P Ouellet, and J Alam. Deep neural networks for extracting baum-welch statistics for speaker recognition. In *Proc. Odyssey*, pages 293–298, 2014.

[44] Luciana Ferrer, Yun Lei, Mitchell McLaren, and Nicolas Scheffer. Study of senone-based deep neural network approaches for spoken language recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(1):105–116, 2016.

[45] Howard Lei. Joint factor analysis (jfa) and i-vector tutorial. *ICSI. Web. 02 Oct*, 2011.

[46] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Trans. on ASLP*, 15(4):1435–1447, 2007.

[47] Alex Solomonoff, Carl Quillen, and William M Campbell. Channel compensation for svm speaker recognition. In *Odyssey*, volume 4, pages 219–226, 2004.

[48] Alex Solomonoff, William M Campbell, and Ian Boardman. Advances in channel compensation for svm speaker recognition. In *Proc. of IEEE ICASSP*, pages 629–632, 2005.

[49] David Andrzejewski. Expectation maximization. `http://pages.cs.wisc.edu/~andrzeje/research/em.pdf`, 2010. Accessed: 2017-06-04.

[50] Lester Mackey, Jordan Bryan, and Dangna Li. Lecture 3 – april 7th. STATS 306B: Unsupervised Learning, 2014. Downloaded: 2017-06-11.

[51] Andrew Ng. The em algorithm. CS229 Lecture Notes, 2016. Downloaded: 2017-06-11.

[52] Mark JF Gales. Cluster adaptive training of hidden markov models. *IEEE transactions on speech and audio processing*, 8(4):417–428, 2000.

[53] Hao Tang, Stephen Chu, Mark Hasegawa-Johnson, and Thomas Huang. Partially supervised speaker clustering. *IEEE transactions on pattern analysis and machine intelligence*, 34(5):959–971, 2012.

[54] Siwei Lyu and Eero P Simoncelli. Nonlinear extraction of independent components of natural images using radial gaussianization. *Neural computation*, 21(6):1485–1519, 2009.

[55] NIST. 2010 nist speaker recognition evaluation. `http://www.itl.nist.gov/iad/mig//tests/sre/2010/NIST_SRE10_evalplan.r6.pdf`. Accessed: 2015-03-27.

[56] Mitchell McLaren, Yun Lei, and Luciana Ferrer. Advances in deep neural network approaches to speaker recognition. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4814–4818. IEEE, 2015.

[57] NIST. The nist year 2004 speaker recognition evaluation plan. No longer available on NIST website. Included therefore at the end of this thesis. Attempted access: 2017-08-19. Corpora available from LDC (LDC2006S44).

[58] NIST. The nist year 2005 speaker recognition evaluation plan. `https://catalog.ldc.upenn.edu/docs/LDC2011S04/sre-05_evalplan-v5.pdf`. Accessed: 2017-09-11. Corpora available from LDC (LDC2011S01, LDC2011S04).

[59] D Graff, K Walker, and A Canavan. Switchboard-2 phase ii ldc99s79. DVD, 1999.

[60] Jean-François Bonastre, Nicolas Scheffer, Corinne Fredouille, and Driss Matrouf. Nist04 speaker recognition evaluation campaign: new lia speaker detection platform based on alize toolkit. *NIST SRE*, 4, 2004.

[61] Ondrej Novotnỳ, Pavel Matejka, Oldrich Plchot, Ondrej Glembek, Lukás Burget, and Jan Cernockỳ. Analysis of speaker recognition systems in realistic scenarios of the sitw 2016 challenge. In *Proc. of Interspeech*, pages 828–832, 2016.

[62] Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson. The 2016 speakers in the wild speaker recognition evaluation. In *Proc. of Interspeech*, pages 823–827, 2016.

[63] NIST. 2012 nist speaker recognition evaluation. `http://www.nist.gov/itl/iad/mig/sre12.cfm`. Accessed: 2015-02-20.

[64] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[65] Fred Richardson, Doug Reynolds, and Najim Dehak. Deep neural network approaches to speaker and language recognition. *IEEE Signal Processing Letters*, 22(10):1671–1675, 2015.

[66] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[67] David Snyder, Daniel Garcia-Romero, and Daniel Povey. Time delay deep neural network-based universal background models for speaker recognition. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 92–97. IEEE, 2015.

[68] Takanori Yamada, Longbiao Wang, and Atsuhiko Kai. Improvement of distant-talking speaker identification using bottleneck features of dnn. In *Proc. of Interspeech*, pages 3661–3664, 2013.

[69] Mehryar Mohri. Speech recognition, lecture 10: pronunciation models. `http://www.cs.nyu.edu/~mohri/asr12/lecture_10.pdf`. Accessed: 2017-08-30.

[70] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. The htk book. *Cambridge university engineering department*, 3:175, 2002.

[71] Fred Richardson, Brian Nemsick, and Douglas Reynolds. Channel compensation for speaker recognition using map adapted plda and denoising dnns. In *Proc. of Odyssey*, pages 225–229, 2016.

[72] Y Lei, L Ferrer, M McLaren, and N Scheffer. Comparative study on the use of senone-based deep neural networks for speaker recognition. *Submitted to IEEE Trans. ASLP*, 2014.

[73] Yan Song, Bing Jiang, YeBo Bao, Si Wei, and Li-Rong Dai. I-vector representation based on bottleneck features for language identification. *Electronics Letters*, 49(24):1569–1570, 2013.

[74] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[75] Yun Lei, Luciana Ferrer, Aaron Lawson, Mitchell McLaren, and Nicolas Scheffer. Application of convolutional neural networks to language identification in noisy conditions. *Proc. Odyssey-14, Joensuu, Finland*, 2014.

[76] deeplearning.net. Convolutional neural networks (lenet). `http://deeplearning.net/tutorial/lenet.html`. Accessed: 2017-09-07.

[77] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243, 1968.

[78] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[79] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[80] Ossama Abdel-Hamid, Li Deng, and Dong Yu. Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Interspeech*, pages 3366–3370, 2013.

[81] Hynek Hermansky. Perceptual linear predictive (plp) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

[82] Chanwoo Kim and Richard M Stern. Power-normalized cepstral coefficients (pncc) for robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4101–4104. IEEE, 2012.

[83] David Graff, Kevin Walker, Stephanie Strassel, Xiaoyi Ma, Karen Jones, and Ann Sawyer. The rats collection: supporting hlt research with degraded audio data. `https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/lrec2014-rats-collection.pdf`. Accessed: 2017-09-08.

[84] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proceedings of Interspeech*, pages 2440–2444. ISCA, 2015.

[85] David Snyder, Daniel Garcia-Romero, and Daniel Povey. Time delay deep neural network-based universal background models for speaker recognition. In *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 92–97. IEEE, 2015.

[86] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, 37(3):328–339, 1989.

[87] Anna Silnova, Lukáš Burget, and Jan Černocky. Alternative approaches to neural network based speaker verification. In *Proc. of Interspeech*, pages 1572–1575, 2017.

[88] Rama Doddipatla, Norbert Braunschweiler, and Ranniery Maia. Speaker adaptation in dnn-based speech synthesis using d-vectors. In *Proc. of Interspeech*, pages 3404–3408, 2017.

[89] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[90] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

[91] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 5115–5119. IEEE, 2016.

[92] UFLDL. Ufldl tutorial - logic regression. `http://ufldl.stanford.edu/tutorial/supervised/LogisticRegression/`. Accessed: 2017-09-05.

[93] Pegah Ghahremani, Vimal Manohar, Daniel Povey, and Sanjeev Khudanpur. Acoustic modelling from the signal domain using cnns. In *INTERSPEECH*, pages 3434–3438, 2016.

[94] Ke Chen and Ahmad Salman. Learning speaker-specific characteristics with a deep neural architecture. *IEEE Transactions on Neural Networks*, 22(11):1744–1756, 2011.

[95] Yu-hsin Chen, Ignacio Lopez-Moreno, Tara N Sainath, Mirkó Visontai, Raziel Alvarez, and Carolina Parada. Locally-connected and convolutional neural networks for small footprint speaker recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[96] Google. Tensorflow: Deep mnist for experts. `https://www.tensorflow.org/tutorials/mnist/pros`. Access: 2016-12-22.

[97] Christopher Cieri, David Graff, Owen Kimball, Dave Miller, and Kevin Walker. Fisher english training speech part 2 speech ldc2005s13. DVD, 2005.

[98] Christopher Cieri, David Graff Graff, Owen Kimball, Dave Miller, and Kevin Walker. Fisher english training speech part 1 speech ldc2004s13. DVD, 2004.

[99] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

The NIST Year 2004 Speaker Recognition Evaluation Plan

1 INTRODUCTION

The year 2004 speaker recognition evaluation is part of an ongoing series of yearly evaluations conducted by NIST. These evaluations provide an important contribution to the direction of research efforts and the calibration of technical capabilities. They are intended to be of interest to all researchers working on the general problem of text independent speaker recognition. To this end the evaluation is designed to be simple, to focus on core technology issues, to be fully supported, and to be accessible to those wishing to participate.

The 2004 evaluation will use new conversational speech data collected in the Mixer Project using the Linguistic Data Consortium's new "Fishboard" platform.[1] This data will be mostly conversational telephone speech in English as in previous evaluations, but it is expected to include some speech in languages other than English and may include some microphone data. The evaluation will include twenty-eight different speaker detection tests defined by the duration and type of both the training and the test segments of the individual trials of which these tests are composed. For each such test, an unsupervised adaptation mode will be offered in addition to the basic test.

The evaluation will be conducted in the spring. The data will be made available to participants in late March, with results due to be submitted to NIST about three and a half weeks later. A follow-up workshop for evaluation participants to discuss research findings will be held early in June. Specific dates are listed in the Schedule (section 11).

Participation in the evaluation is invited for all sites that find the tasks and the evaluation of interest. For more information, and to register to participate in the evaluation, please contact Dr. Alvin Martin at NIST.[2]

2 TECHNICAL OBJECTIVE

This evaluation focuses on speaker detection, posed primarily in the context of conversational telephone speech. The evaluation is designed to foster research progress, with the goals of:

- Exploring promising new ideas in speaker recognition.
- Developing advanced technology incorporating these ideas.
- Measuring the performance of this technology.

2.1 Task Definition

The year 2004 speaker recognition evaluation is limited to the broadly defined task of **speaker detection**. This has been NIST's basic speaker recognition task over the past eight years. The task is to determine whether a specified speaker is speaking during a given speech segment.[3]

2.2 Task Conditions

Previous evaluations have included both a limited data condition and an extended data condition. Limited data meant that the training and test segment data for each trial consisted of two minutes or less of concatenated segments of speech data, with silence intervals removed, while extended data meant that each of these consisted of an entire conversation side or, for training, multiple conversation sides. It has been decided this year to remove the specific distinction between limited and extended data tests, and to no longer do silence removal, but to offer multiple testing conditions involving the amount and type of data available for both the training and the test segments.

Thus the speaker detection task for 2004 includes tests involving seven distinct training conditions and four distinct (test) segment conditions. There will thus be 28 different combinations of training/segment conditions. A test (sequence of trials) will be offered for each of these combinations. One of these (see section 2.2.3) is designated the core test. Participants must do the core test and may choose to do any subset of the remaining tests. Results must be submitted for all trials included in each test for which any results are submitted. For each test, there will also be an optional unsupervised adaptation condition. A site may do the adaptation condition for a particular test only if it also does the particular test without adaptation.

2.2.1 Training Conditions

The training segments in the 2004 evaluation will be continuous conversational excerpts. Unlike in past years, there will be no prior removal of intervals of silence. For some training conditions the NIST energy-based automatic speech detector will be used to estimate the duration of actual speech in the chosen excerpts.

The seven training conditions to be included involve target speakers defined by the following training data:

1. An excerpt from a single channel conversation side estimated to contain approximately 10 seconds of speech
2. An excerpt from a single channel conversation side estimated to contain approximately 30 seconds of speech
3. A single channel conversation side, of approximately five minutes total duration[4]

---

[1] See http://www.upenn.edu/mixer/

[2] To contact Dr. Martin, send him email at alvin.martin@nist.gov, or call him at 301/975-3169. Each site must return a signed registration form to complete the registration process: http://www.nist.gov/speech/tests/spk/2004/register.pdf

[3] In previous evaluation plans, the speaker detection task was divided into a "one-speaker" and a "two-speaker" task. However, this distinction relates to the task conditions rather than the task definition. Therefore in this evaluation plan the one- and two-speaker conditions, both for training and for test segments, are included under task conditions in section 2.2.

[4] Each conversation side will consist of the last five minutes of a six-minute conversation. This will eliminate from the evaluation data the less-topical introductory dialogue, which is more likely to contain identifying information about the speakers.

4. Three single channel conversation sides involving the same speaker
5. Eight single channel conversation sides involving the same speaker
6. Sixteen single channel conversation sides involving the same speaker
7. Three summed-channel conversations, formed by sample-by-sample summing of the two sides of actual conversations, each including a common speaker (the target of interest) and a second speaker not participating in the other two conversations

Word transcripts derived from an automatic speech recognition (ASR) system[5] will be provided for all English training segments of each condition. These transcripts will, of course, be errorful, perhaps with word error rates in the 20-30% range.

2.2.2 Test Segment Conditions

The test segments in the 2004 evaluation will be continuous conversational excerpts. Unlike in past years, there will be no prior removal of intervals of silence. For some test segment conditions the NIST automatic speech detector will be used to estimate the duration of actual speech in the chosen excerpts.

The four test segment conditions to be included are the following:

1. An excerpt from a single channel conversation side estimated to contain approximately 10 seconds of speech
2. An excerpt from a single channel conversation side estimated to contain approximately 30 seconds of speech
3. A single channel conversation side, of approximately five minutes total duration[4]
4. A summed channel conversation, formed by sample-by-sample summing of the two sides of an actual conversation

Errorful ASR word transcripts derived from an ASR system will be provided for all English-language test segments of each condition.

2.2.3 Training/Segment Condition Combinations

The matrix of training and segment condition combinations is shown in Table 1. A test will be offered for each combination. Each test consists of a sequence of trials, where each trial consists of a target speaker, defined by the training data provided, and a test segment. The system must decide whether speech of the target speaker occurs in the test segment. Both an actual decision ('T' or 'F') and a likelihood score, indicating system confidence in the correctness of this decision, must be submitted for each trial.

The shaded box in Table 1 corresponds to the condition of a single conversation side as training and a single conversation side as test segment. The test for this condition will be defined as the **core test** for the 2004 evaluation. All participants are required to submit results for this core test. Each participant may choose to also submit results for all, some, or none of the other 27 test conditions. For each test for which results are submitted, they must be submitted for all trials included in the test.

[5] All conversations will be processed at BBN using a system derived from their RT-03 conversational telephone speech STT evaluation system.

Table 1: Matrix of training and test segment conditions. The shaded entry is the required core test condition.

		Test Segment Condition			
		10 sec	30 sec	1 side	1 conv
Training Condition	10 sec	X	X	X	X
	30 sec	X	X	X	X
	1 side	X	X	X	X
	3 sides	X	X	X	X
	8 sides	X	X	X	X
	16 sides	X	X	X	X
	3 convs	X	X	X	X

2.2.4 Unsupervised Adaptation Mode

In previous evaluations, adaptive strategies were not allowed and each trial was restricted to use data from a single test segment and a single (static) model. This year, an unsupervised adaptation mode will be supported, allowing models to be updated based on test segments processed in previous trials.

As in previous evaluations, for each trial systems may not in general use information about other evaluation target speakers or other test segments. In unsupervised adaptation mode, however, the trials for each target speaker model must be processed in order, and after each trial the model may optionally be updated based on the test segments in the preceding trials. How this is accomplished should be discussed in the system descriptions (see section 10).

Thus for each test, participants will have the option of doing the test in unsupervised adaptation mode. Unsupervised adaptation results may only be submitted for tests for which (standard) non-adaptive results are also submitted, and the performance results with and without such adaptation will be compared.

3 PERFORMANCE MEASURE

There will be a single basic cost model for measuring speaker detection performance, to be used for all speaker detection tests. This detection cost function is defined as a weighted sum of miss and false alarm error probabilities:

$$C_{Det} = C_{Miss} \times P_{Miss|Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1-P_{Target})$$

The parameters of this cost function are the relative costs of detection errors, C_{Miss} and $C_{FalseAlarm}$, and the *a priori* probability of the specified target speaker, P_{Target}. The parameter values in **Table** 2 will be used as the primary evaluation of speaker recognition performance for all speaker detection tests.

Table 2: Speaker Detection Cost Model Parameters for the primary evaluation decision strategy

C_{Miss}	$C_{FalseAlarm}$	P_{Target}
10	1	0.01

3.1 Normalization

One of the advantages of using a cost model is that it can be easily applied to different applications simply by changing the model parameters. On the other hand, a potential disadvantage of using cost as a performance measure is that it gives values that often lack intuitive meaning. To improve the intuitive value of the cost defined to be the best cost that could be obtained without processing the input data (i.e., by always making the same decision, namely either to accept or to reject the segment speaker as being the target speaker, whichever gives the lowest cost):

$$C_{Default} = \min\begin{Bmatrix} C_{Miss} \times P_{Target}, \\ C_{FalseAlarm} \times (1 - P_{Target}) \end{Bmatrix}$$

and

$$C_{Norm} = C_{Det} / C_{Default}$$

4 EVALUATION CONDITIONS

Speaker detection performance will be evaluated in terms of the detection cost function. For each test, the cost function will be computed over the sequence of trials provided and over subsets of these trials of particular evaluation interest. Each trial must be independently judged as "true" (the model speaker speaks in the test segment) or "false" (the model speaker does not speak in the test segment), and the correctness of these decisions will be tallied.[6]

In addition to the actual detection decision, a decision (likelihood) score will also be required for each test hypothesis. Higher scores will be taken to indicate greater confidence that "true" is the correct decision and lesser confidence that "false" is the correct decision. This decision score will be used to produce detection error tradeoff curves, in order to see how misses may be traded off against false alarms.[7]

4.1 Training Data

As discussed in section 2.2.1, there will be seven training conditions. NIST will be interested in examining how performance varies among these conditions for fixed test segment conditions.

Most of the training data will be in English, but some training conversations involving bi-lingual speakers may be collected in Arabic, Mandarin, Russian, and Spanish. Thus it will then be possible to examine how performance is affected by whether or not the training language matches the language, generally English, of the test data. For the training conditions involving multiple conversations, the effect of having a mix of languages in the training may also be examined. The language used in all training data files will be indicated in the file header and available for use.

All training data is expected to be collected over telephone channels.

The sex of each target speaker will be provided to systems.

For all training conditions, errorful ASR transcriptions of all English language data will be provided along with the audio data. Systems may utilize the data provided as they wish. The acoustic data may be used alone, the transcriptions may be used alone, or all data may be used in combination.[8]

4.1.1 Single Channel Excerpts

As discussed in section 2.2.1, there will be training conditions consisting of excerpts with approximately 10-seconds and approximately 30-seconds of estimated speech duration. These estimated durations will vary so that the excerpts may include only whole turns whenever possible, but they will be constrained to lie in the ranges of 8-12 seconds for "10-second" excerpts, and 25-35 seconds for "30-second" excerpts.

4.1.2 Single Channel Conversation Sides

As discussed in section 2.2.1, there will be training conditions consisting of one, three, eight, or sixteen single conversation sides of a given speaker. These sides will consist of approximately five minutes from an original six minute conversation side, with an initial segment of around one minute excised. The excision point will be chosen so as not to include a partial speech turn. Areas of silence within the five minutes of conversation chosen will not be excised.

4.1.3 Summed Channel Conversations

As discussed in section 2.2.1, the final training condition will consist of three whole conversations, minus initial segments of about a minute each. In contrast with the other training conditions, however, the two sides of each conversation, in which both the target speaker of interest and another speaker participate, will be summed together. Thus the challenge is to distinguish speech by the intended target speaker from speech by other participating speakers. To make this challenge feasible, the training conversations will be chosen so that each non-target speaker participates in only one conversation, while the target speaker participates in all three.

The difficulty of finding the target speaker's speech in the training data is affected by whether the other speaker in a training conversation is of the same or of the opposite sex as the target. Systems will not be provided with this information, but may use automatic gender detection techniques if they wish. Performance results will be examined as a function of how many of the three training conversations contain same-sex other speakers.

Note that an interesting contrast will exist between this training condition and that consisting of three single conversation sides.

[6] This means that an explicit speaker detection decision is required for each trial. Explicit decisions are required because the task of determining appropriate decision thresholds is a necessary part of any speaker detection system and is a challenging research problem in and of itself.

[7] Decision scores for all of the trials in a given test will be pooled before plotting detection error tradeoff curves. Thus it is necessary to normalize scores across speakers to achieve satisfactory detection performance.

[8] Note, however, that there will be some non-English training data, for which no meaningful ASR transcripts will be available.

4.2 Test data

As discussed in section 2.2.2, there will be four test segment conditions. NIST will be interested in examining how performance varies among these conditions for fixed training conditions.

For a limited number of speakers some test conversations may be collected using non-telephone channels. Several microphone types will be included in this collection. Thus it will be possible to examine how performance is affected by whether or not test data is recorded over a telephone channel, and by the type of microphone used in non-telephone test data. The non-telephone data will include some or all of the following microphone types:

- Ear-bud/lapel mike
- Miniboom mike
- Courtroom mike
- Conference room mike
- Distant mike
- Near-field mike
- PC stand mike
- Microcassette mike

Information on the microphone type used in each non-telephone test segment data will be available to recognition systems.

With rare exceptions, all test data speech is expected to be in English.

For all test segments conditions, errorful ASR transcriptions of the (English language) data will be provided along with the audio data. Systems may utilize the data provided as they wish. The acoustic data may be used alone, the ASR transcriptions may be used alone, or all data may be used in combination.

4.2.1 Single Channel Excerpts

As discussed in section 2.2.2, there will be test segment conditions consisting of excerpts with approximately 10-seconds and approximately 30-seconds of estimated speech duration These estimated durations will vary so that the excerpts may include only whole turns whenever possible, but they will be constrained to lie in the ranges of 8-12 seconds for "10-second" excerpts, and 25-35 seconds for "30-second" excerpts.

4.2.2 Single Channel Conversation Sides

As discussed in section 2.2.2, there will be a test segment condition consisting of a single conversation side of a given speaker. Each such side will consist of approximately five minutes from an original six minute conversation side, with an initial segment of around one minute excised. The excision point will be chosen so as not to include a partial speech turn. Areas of silence within the five minutes of conversation chosen will not be excised.

4.2.3 Summed Channel Conversations

As discussed in section 2.2.2, the there will be a test segment condition consisting of a single whole conversation, minus an initial segment of about a minute. In contrast with the other test segment conditions, however, the two sides of this conversation will be summed together, and both the target speaker and that speaker's conversation partner will be represented in each conversation.

The difficulty of determining whether the target speaker speaks in the test conversation is affected by the sexes of the speakers in the test conversation. For no trials will both speakers be of opposite sex from the target. Systems will not be told whether the two test speakers are of the same or opposite sex, but may use automatic gender detection techniques if they wish. Performance results will be examined with respect to whether one or both target conversation speakers are of the same sex as the target.

Note that an interesting contrast will exist between this condition and that consisting of a single conversation side.

4.3 Factors Affecting Performance

All trials will be same-sex trials. This means that the sex of the test segment speaker, or of at least one test segment speaker when the test segment is a summed channel conversation, will be the same as that of the target speaker model. Performance will be reported separately for males and females and pooled across sex.

All trials involving telephone test segments will be different number trials. This means that the telephone numbers, and presumably the telephone handsets, used in the training and the test data segments will be different from each other.

Past NIST evaluations have shown that the type of telephone handset and the type of telephone transmission channel used can have a great effect on speaker recognition performance. Factors of these types will be examined in this evaluation.

Telephone callers in the Mixer collection (see section 6) are asked to classify the transmission channel as one of the following types:

- Cellular
- Cordless
- Regular (ie., land-line)

Telephone callers in the Mixer collection are asked to classify the instrument used as one of the following types:

- Speaker-phone
- Head-mounted
- Ear-bud
- Regular (ie., hand-held)

Performance will be examined as a function of the telephone transmission channel type and of the telephone instrument type in both the training and the test segment data. The effects of different types of cellular transmission encoding may also be considered.

4.4 Unsupervised Adaptation

As discussed in section 2.2.4, an unsupervised adaptation mode will be supported for each test. Performance with and without such adaptation will be compared for participants attempting tests with unsupervised adaptation.

4.5 Common Evaluation Condition

In each evaluation NIST specifies a common evaluation condition.[9] The performance results on trials satisfying this condition are

[9] In past NIST evaluations this was referred to as the "primary" condition. The term "common evaluation condition" is more

treated as the basic official evaluation outcome. The common evaluation condition in the 2004 evaluation will be regarded as all trials meeting each of the following specifications:

- Part of the core test as defined in section 2.2.3
- All training and test speech in English
- All training and test speech involve a telephone channel
- Male or female target – pooled across sex
- Hand-held telephone instruments used in all training and test speech
- All training and test segment data involves either land-line or cellular (not cordless) telephone transmission channels

4.6 Comparison With Previous Evaluations

In each evaluation it is of interest to compare performance results, particularly of the best performing systems, with those of previous evaluations. This is complicated by the fact that the evaluation conditions change in each successive evaluation.

The 2004 evaluation is no exception in this regard. The concatenation of speech segments with areas of silence removed, as practiced in the past, will not be done this year. There will be multiple durations utilized in the various training and test conditions, and none will be exactly as in the past. And whereas past evaluations have largely focused either primarily on landline data or primarily of cellular data, this evaluation will involve a mixture of both.

Nonetheless, NIST will examine how the results for test conditions most similar to those used in past compare with the past results. The test condition most similar to the "limited data" condition of the recent evaluations will be that involving training on a single conversation side and testing on 30-second excerpts. The condition most similar to the most widely examined "extended data" results of the past will be that involving training on eight conversation sides and testing on a single conversation sides. Participating sites, particularly those that participated in previous evaluations, may wish to consider including one of these tests in the results they submit for this evaluation. NIST will examine, after the fact, results on the subsets of trials of these tests that most resemble the conditions of past evaluation tests to facilitate the most meaningful comparison of performance results achieved over time in the course of the evaluations.

5 DEVELOPMENT DATA

The evaluation data for 2003 evaluation will serve as the development data for this year's evaluation, and will be covered by the LDC license agreement noted in section 6. Please refer to last year's evaluation plan for details.[10]

Note that no development data that is specific to the changed format and collection methods of the 2004 evaluation data (described in section 6) is being provided. Participating sites may use other speech corpora to which they have access for development. Such corpora should be described in the system descriptions. The original Switchboard-1 Corpus may be used, but participating sites are cautioned, particularly with respect to the development of background speaker models, that an effort is being made to recruit a limited number of the speakers in that corpus to participate in the new data collections from which this year's evaluation data will be selected.

6 EVALUATION DATA

The training and test segment data will be all newly collected by the Linguistic Data Consortium (LDC). The Mixer Project invited participating speakers to take part in numerous six-minute conversations on specified topics with people they did not know. The Fishboard platform allowed an automaton to initiate calls to selected pairs of speakers for most of the conversations, while individual speakers initiated some calls themselves, with the automaton contacting other speakers for them to converse with. Speakers initiating calls were encouraged to use unique telephone numbers (and thus generally unique telephone handsets) for their initiated calls.

The conversational data for this evaluation, to be distributed to participants by NIST on CD-ROM's, has not been publicly released. The LDC will provide a license agreement, which non-member participating sites must sign, governing the use of this data for the evaluation. The ASR transcript data, and any other auxiliary data which may also be supplied, will be made available by NIST in electronic form to all registered participants.

All conversations will have been processed through echo canceling software before being used to create the evaluation training and test segments.

All training and test segments will be stored as 8-bit mu-law continuous speech signals in separate SPHERE files. The SPHERE header of each such file will contain some auxiliary information as well as the standard SPHERE header fields. This auxiliary information will include the language of the conversation, whether or not the data was recorded over a telephone line, and the microphone type for non-telephone data. Most segments will be in English and recorded over a telephone line. The header will not contain information on the type of telephone transmission channel or the type of telephone instrument involved.

6.1 Single Channel Excerpts

The 10-second and 30-second excerpts to be used as training or as test segments will be continuous segments from single conversation sides that are estimated to contain approximately 10 or 30 seconds of actual speech.

The number of single channel excerpt training segments both for the 10-second and for the 30-second training conditions is expected to be around 600. The number of single channel excerpt test segments for each of the two durations is expected to be around 2000.

6.2 Single Conversation Sides

The single conversation sides to be used as training data or as test segments will all be approximately five minutes in total signal duration.

The number of single conversation training sides is expected not to exceed 6400. The number of these to be used to create speaker models based on a single conversation side is expected to be around

---

appropriate in the sense that this condition is used to officially rank system performance, and is not necessarily the condition that is most important to the evaluation.

[10] The year 2003 speaker recognition evaluation plan may be accessed from http://www.nist.gov/speech/tests/spk/2003/doc/

600. The numbers of models specified by 3, 8, or 16 sides are each expected to be around 400 or fewer.

The number of single conversation side test segments is expected to be around 2000.

6.3 Summed Channel Conversations

The summed-channel conversations to be used as training data or as test segments will all be approximately five minutes in total signal duration

The number of summed channel training conversations is expected to be around 1200. These will be used to specify around 400 target speaker models. The number of summed-channel conversation test segments is expected to be around 2000.

6.4 Trials to be included

The trials for each of the 28 speaker detection tests offered will be specified in separate index files. These will be text files in which each record specifies the model and a test segment for a particular trial. The number of trials in each test is expected not to exceed 25,000.

7 EVALUATION RULES

In order to participate in the 2004 speaker recognition evaluation, a site must complete, in its entirety, the core test condition (without unsupervised adaptation) as specified in section 2.2.3.[11] Any other test conditions included must be completed in their entirety.

All participants must observe the following evaluation rules and restrictions:

- Each decision is to be based only upon the specified test segment and target speaker model. Use of information about other test segments (except as permitted for the unsupervised adaptation mode condition) and/or other target speakers is **not** allowed.[12] For example:
  - Normalization over multiple test segments is **not** allowed, except as permitted for the unsupervised adaptation mode condition.
  - Normalization over multiple target speakers is **not** allowed.
  - Use of evaluation data for impostor modeling is **not** allowed.
- If an unsupervised adaptation condition is included, the test segments must be processed in the order specified.
- The use of manually produced transcripts or other information for training is **not** allowed.
- Knowledge of the sex of the *target* speaker (implied by data set directory structure as indicated below) **is** allowed. Note that there will be no cross-sex trials.
- Knowledge of the language used in all segments, which will be provided, is allowed.
- Knowledge of whether or not a segment involves telephone channel transmission, and of the non-telephone microphone type used, which will be provided, is allowed.
- Knowledge of the telephone transmission channel type and of the telephone instrument type used in all segments is not allowed, except as determined by automatic means.
- Listening to the evaluation data, or any other experimental interaction with the data, is **not** allowed before all test results have been submitted. This applies to training data as well as test segments.
- Knowledge of any information available in the SPHERE header **is** allowed.

8 EVALUATION DATA SET ORGANIZATION

The organization of the evaluation data will be:

- A top level directory used as a unique label for the disk: "**sp04-NN**" where NN is a digit pair identifying the disk
- Under which there will be four sub-directories: "**train**", "**test**", "**trials**", and "**doc**"

8.1 train Subdirectory

The "**train**" directory contains three subdirectories:

- **data**: Contains all the SPHERE formatted speech data used for training in each of the seven training conditions.
- **female**: Contains seven training files that defines the *female* models for each of the seven training conditions. (The format of these files is defined below.)
- **male**: Contains seven training files that defines the *male* models for each of the seven training conditions. (The format of these files is defined below.)

The seven training files for both male and female models have the same structure. There is one record per line, and each record contains two fields. The first field is the model identifier and the second field is a comma separated list of speech files (located in the "**data**" directory) that are to be used to train the model.

The seven training files in each gender directory are named:

- "***10sec.trn***" for the 10 second training condition, an example record looks like: "`3232 mrpv.sph`"
- "***30sec.trn***" for the 30 second training condition, an example record looks like: "`5241 mrpw.sph`"
- "***1side.trn***" for the 1 side training condition, an example record looks like: "`4240 mrpz.sph`"
- "***3sides.trn***" for the 3 sides training condition, an example record for this training condition looks like: "`7211 mrpz.sph,hrtz.sph,nost.sph`"
- "***8sides.trn***" for the 8 sides training condition.
- "***16sides.trn***" for the 16 sides training condition.
- "***3convs.trn***" for the 3 conversations (summed sides) training condition, an example record looks like: "`3310 nrfs.sph,irts.sph,poow.sph`"

[11] Participants are encouraged to do as many tests as possible. However, it is absolutely imperative that results for all of the trials in a test be submitted in order for that test to be considered valid and for the results to be accepted.

[12] This means that the technology is viewed as being "application-ready". Thus a system must be able to perform speaker detection simply by being trained on a specific target speaker and then performing the detection task on whatever speech segment is presented, without the (artificial) knowledge of other test data.

8.2 test Subdirectory

The "**test**" directory contains one subdirectory:

- **data**: This directory contains all the SPHERE formatted speech test data to be used for each of the four test segment conditions. The file names will be arbitrary ones of four characters along with a ".sph" extension.

8.3 trials Subdirectory

The "**trials**" directory contains twenty-eight index files, one for each of the possible combinations of the seven training conditions and four test segment types. These index files define the various evaluation tests. The naming convention for these index files will be "*TrainCondition-TestCondition*.ndx" where *TrainCondition*, refers to the training condition and whose models are defined in the corresponding training file. Possible values for *TrainCondition* are: 10sec, 30sec, 1side, 3sides, 8sides, 16sides, and 3convs. "*TestCondition*" refers to the test segment condition. Possible values for *TestCondition* are: 10sec, 30sec, 1side, and 1conv.

Each record in a *TrainCondition-TestCondition.ndx* file contains exactly three fields and defines a single trial. The first field is the model identifier. The second field identifies the gender of the model, either "*m*" or "*f*". The third field is the test segment under evaluation, located in the **test/data** directory. This test segment name will not include the .sph extension. An example for the train on 3-sides and test on 1side index file "3sides-1side.ndx" looks like: "`7211 m nrbw`".

The records in these 28 files are ordered numerically by model identifier, and within each model's tests, alphabetically by the test segments. Each index file orders the trials as they are to be processed when unsupervised adaptation is used

8.4 doc Subdirectory

This will contain text files that document the evaluation and the organization of the evaluation data. This evaluation plan document will be included.

9 SUBMISSION OF RESULTS

Sites participating in one or more of the speaker detection evaluation tests must report results for each test in its entirety. These results for each test condition (1 of the 28 test index files) must be provided to NIST in a single file using a standard ASCII format, with one record for each trial decision. The file name should be intuitively mnemonic and should be constructed as "SSS_N", where

- SSS identifies the site, and
- N identifies the system.

9.1 Format for Results

Each file record must document its decision with the target model identification, test segment identification, and decision information. Each record must contain eight fields, separated by white space and in the following order:

1. The training type of the test – **10sec**, **30sec**, **1side**, **3sides**, **8sides**, **16sides**, or **3convs**
2. Adaptation mode. "**n**" for no adaptation and "**u**" for unsupervised adaptation.
3. The segment type of the test – **10sec**, **30sec**, **1side**, or **1conv**
4. The sex of the target speaker – **m** or **f**
5. The target model identifier
6. The test segment (minus the ".sph" extention).
7. The decision – **t** or **f** (whether or not the target speaker is judged to match the speaker in the test segment)
8. The likelihood score (where the more positive this score, the more likely the target and segment speakers are judged to match)

9.2 Means of Submission

Submissions may be made via email or via ftp. The appropriate addresses for submissions will be supplied to participants receiving evaluation data.

10 SYSTEM DESCRIPTION

A brief description of the system(s) (the algorithms) used to produce the results must be submitted along with the results, for each system evaluated. It is permissible for a single site to submit multiple systems for evaluation for a particular test. In this case, however, the submitting site must identify one system as the "primary" system for the test prior to performing the evaluation.

Sites must report the CPU execution time that was required to process the evaluation data, as if the test were run on a single CPU. This should be reported separately for creating models from the training data and for processing the test segments, and may be reported either as absolute processing time or as a multiple of real-time for the data processed. The additional time required for unsupervised adaptation should be reported where relevant. Sites must also describe the CPU and the amount of memory used.

11 SCHEDULE

The deadline for signing up to participate in the evaluation is March 14, 2004.

The evaluation data set CD-ROM's will be distributed by NIST on March 29, 2004.

The deadline for submission of evaluation results to NIST is April 22, 2004.

Evaluation results will be released to each site by NIST on April 29, 2004.

The deadline for site workshop presentations to be supplied to NIST in electronic form for inclusion in the workshop CD-ROM is May 27, 2004.

Registration and room reservations for the workshop must be received by (a date to be determined).

The follow-up workshop will be held on June 3-4, 2004 at the Hotel Beatriz in Toledo, Spain in conjunction with the 2004: A Speaker Odyssey workshop on speaker and language recognition. Those participating in the evaluation are expected to present and discuss their findings at this NIST portion of the workshop.

12 GLOSSARY

Trial – The individual evaluation unit involving a test segment and a hypothesized speaker.

Target (true speaker) trial – A trial in which the actual speaker of the test segment *is in fact* the target (hypothesized) speaker of the test segment.

Non-target (impostor) trial – A trial in which the actual speaker of the test segment *is in fact not* the target (hypothesized) speaker of the test segment.

Target (model) speaker – The hypothesized speaker of a test segment, one for whom a model has been created from training data.

Non-target (impostor) speaker – A hypothesized speaker of a test segment who is in fact not the actual speaker.

Segment speaker – The actual speaker in a test segment.

Test – A collection of trials constituting an evaluation component.

Turn – The interval during a conversation during when one participant speaks while the other remains silent.

The NIST Year 2005 Speaker Recognition Evaluation Plan

1 INTRODUCTION

The year 2005 speaker recognition evaluation is part of an ongoing series of yearly evaluations conducted by NIST. These evaluations are an important contribution to the direction of research efforts and the calibration of technical capabilities. They are intended to be of interest to all researchers working on the general problem of text independent speaker recognition. To this end the evaluation is designed to be simple, to focus on core technology issues, to be fully supported, and to be accessible to those wishing to participate.

The 2005 evaluation will use conversational telephone speech data collected for the Mixer Corpus by the Linguistic Data Consortium using the "Fishboard" platform plus some "multi-channel" data collected simultaneously from a number of auxiliary microphones. The data will be mostly English speech, but it may include some speech in four additional languages.

The evaluation will include twenty different speaker detection tests defined by the duration and type of the training and test data. For each such test, an unsupervised adaptation mode will be offered in addition to the basic test.

The evaluation will be conducted in April 2005. A follow-up workshop for evaluation participants to discuss research findings will be held in June. Specific dates are listed in the Schedule (section 11).

Participation in the evaluation is invited for all sites that find the tasks and the evaluation of interest. Participating sites must follow the evaluation rules set forth in this plan and must be represented at the evaluation workshop. For more information, and to register to participate in the evaluation, please contact Dr. Alvin Martin at NIST.[1]

2 TECHNICAL OBJECTIVE

This evaluation focuses on speaker detection in the context of conversational telephone speech. The evaluation is designed to foster research progress, with the goals of:

- Exploring promising new ideas in speaker recognition.
- Developing advanced technology incorporating these ideas.
- Measuring the performance of this technology.

2.1 Task Definition

The year 2005 speaker recognition evaluation is limited to the broadly defined task of **speaker detection**. This has been NIST's basic speaker recognition task over the past nine years. The task is to determine whether a specified speaker is speaking during a given segment of conversational speech.

2.2 Task Conditions

The speaker detection task for 2005 is divided into 20 distinct and separate tests. Each of these tests involves one of five training conditions and one of four test conditions. One of these tests (see section 2.2.3) is designated the core test. Participants must do the core test and may choose to do any one or more of the other 19 tests. Results must be submitted for ***all*** trials included in each test for which any results are submitted. For each test, there will also be an optional unsupervised adaptation condition. Sites choosing the adaptation option for a test must also perform the test without adaptation to provide a baseline contrast.

2.2.1 Training Conditions

The training segments in the 2005 evaluation will be continuous conversational excerpts. Unlike in some previous years, but as in 2004, there will be no prior removal of intervals of silence. Also, for the first time, both sides of all two-channel conversations will be provided (to aid systems in echo cancellation, dialog analysis, etc.). For all two-channel segments, the channel containing the putative target speaker to be recognized will be identified.

The five training conditions to be included involve target speakers defined by the following training data:

1. A two-channel (4-wire) excerpt from a conversation estimated to contain approximately 10 seconds of speech of the target on its designated side (The NIST energy-based automatic speech detector will be used to estimate the duration of actual speech in the chosen excerpts.)
2. One two-channel (4-wire) conversation, of approximately five minutes total duration[2], with the target speaker channel designated.
3. Three two-channel (4-wire) conversations involving the target speaker on their designated sides
4. Eight two-channel (4-wire) conversations involving the target speaker on their designated sides
5. Three summed-channel (2-wire) conversations, formed by sample-by-sample summing of their two sides. Each of these conversations will include both the target speaker and another speaker. These three non-target speakers will all be distinct.

English language word transcripts, produced using an automatic speech recognition (ASR) system, will be provided for all training segments of each condition. These transcripts will, of course, be errorful, with word error rates typically in the range of 15-30%.

[1] To contact Dr. Martin, send him email at alvin.martin@nist.gov, or call him at 301/975-3169. Each site must return a signed registration form to complete the registration process: http://www.nist.gov/speech/tests/spk/2005/register.pdf

[2] Each conversation side will consist of the last five minutes of a six-minute conversation. This will eliminate from the evaluation data the less-topical introductory dialogue, which is more likely to contain language that identifies the speakers.

2.2.2 Test Segment Conditions

The test segments in the 2005 evaluation will be continuous conversational excerpts. Unlike in some previous years, but as in 2004, there will be no prior removal of intervals of silence. Also, for the first time, both sides of all two-channel conversations will be provided (to aid systems in echo cancellation, dialog analysis, etc.). For all two-channel segments, the channel containing the putative target speaker to be recognized will be identified.

The four test segment conditions to be included are the following:

1. A two-channel (4-wire) excerpt from a conversation estimated to contain approximately 10 seconds of speech of the putative target speaker on its designated side (The NIST energy-based automatic speech detector will be used to estimate the duration of actual speech in the chosen excerpts.)
2. A two-channel (4-wire) conversation, of approximately five minutes total duration, with the putative target speaker channel designated.
3. A summed-channel (2-wire) conversation formed by sample-by-sample summing of its two sides
4. A two-channel (4-wire) conversation, with the usual telephone speech replaced by auxiliary microphone data in the putative target speaker channel. This auxiliary microphone data will be supplied in 8 kHz 8-bit μ-law form.

English language word transcripts, produced using an automatic speech recognition (ASR) system, will be provided for all test segments of each condition.

2.2.3 Training/Segment Condition Combinations

The matrix of training and test segment condition combinations is shown in **Table 1**. A test will be offered for each combination. Each test consists of a sequence of trials, where each trial consists of a target speaker, defined by the training data provided, and a test segment. The system must decide whether speech of the target speaker occurs in the test segment. The shaded box labeled "required" in **Table 1** is the **core test** for the 2005 evaluation. All participants are required to submit results for this test. Each participant may also choose to submit results for all, some, or none of the other 19 test conditions. For each test for which results are submitted, results for **all** trials must be included.

2.2.4 Unsupervised Adaptation Mode

The unsupervised adaptation mode allows systems to update themselves based on previous trial segments for the target model involved (up to and including the current trial segment). This is in contrast to the non-adaptive mode in which the system is static and the target (and background) speaker models are a function only of the target speaker training data. (The speaker models of course also benefit from speech data used and knowledge acquired during system development.)

In the unsupervised adaptation mode it is required that the trials for each target model be performed in the order given in the test index file (see section 8.3). The trials for each model will be grouped together, and the test segments for each of these target models will be listed in chronological order. Within the testing for each target model, the target (and background) models may be updated by the system after each trial using the test segment data processed thus far for that target model. However, the adaptation must be discarded and the system reset to its initial unadapted state whenever a new model is encountered in the test index file.

Table 1: Matrix of training and test segment conditions. The shaded entry is the required core test condition.

		Test Segment Condition			
		10 sec 2-chan	**1 conv 2-chan**	**1 conv summed-chan**	**1 conv aux mic**
Training Condition	**10 seconds 2-channel**	optional	optional	optional	optional
	1 conversation 2-channel	optional	**required**	optional	optional
	3 conversation 2-channel	optional	optional	optional	optional
	8 conversation 2-channel	optional	optional	optional	optional
	3 conversation summed-channel	optional	optional	optional	optional

For each test performed in unsupervised adaptation mode results must also be submitted for that test in non-adaptive mode in order to provide a contrast between adaptive and non-adaptive performance. The unsupervised adaptation techniques used should be discussed in the system description (see section 10).

3 PERFORMANCE MEASURE

There will be a single basic cost model for measuring speaker detection performance, to be used for all speaker detection tests. This detection cost function is defined as a weighted sum of miss and false alarm error probabilities:

$$C_{\text{Det}} = C_{\text{Miss}} \times P_{\text{Miss|Target}} \times P_{\text{Target}} + C_{\text{FalseAlarm}} \times P_{\text{FalseAlarm|NonTarget}} \times (1-P_{\text{Target}})$$

The parameters of this cost function are the relative costs of detection errors, C_{Miss} and $C_{\text{FalseAlarm}}$, and the *a priori* probability of the specified target speaker, P_{Target}. The parameter values in **Table 2** will be used as the primary evaluation of speaker recognition performance for all speaker detection tests.

Table 2: Speaker Detection Cost Model Parameters for the primary evaluation decision strategy

C_{Miss}	$C_{\text{FalseAlarm}}$	P_{Target}
10	1	0.01

To improve the intuitive meaning of C_{Det}, it will be normalized by dividing it by the best cost that could be obtained without processing the input data (i.e., by either always accepting or always rejecting the segment speaker as matching the target speaker, whichever gives the lower cost):

$$C_{\text{Default}} = \min\begin{Bmatrix} C_{\text{Miss}} \times P_{\text{Target}}, \\ C_{\text{FalseAlarm}} \times (1-P_{\text{Target}}) \end{Bmatrix}$$

and

$$C_{\text{Norm}} = C_{\text{Det}} / C_{\text{Default}}$$

4 EVALUATION CONDITIONS

Speaker detection performance will be evaluated in terms of the detection cost function. For each test, the cost function will be computed over the sequence of trials provided and over subsets of these trials of particular evaluation interest. Each trial must be independently judged as "true" (the model speaker speaks in the test segment) or "false" (the model speaker does not speak in the test segment), and the correctness of these decisions will be tallied.[3]

In addition to the actual detection decision, a confidence score will also be required for each test hypothesis. This confidence score is the system's estimate of the probability that the test segment contains speech from the target speaker. This confidence score will be used to produce detection error tradeoff curves, in order to see how misses may be traded off against false alarms.[4]

4.1 Training Data

As discussed in section 2.2.1, there will be five training conditions. NIST is interested in examining how performance varies among these conditions for fixed test segment conditions.

Most of the training data will be in English, but some training conversations involving bi-lingual speakers may be collected in Arabic, Mandarin, Russian, and Spanish. Thus it will then be possible to examine how performance is affected by whether or not the training language matches the language of the test data. For the training conditions involving multiple conversations, the effect of having a mix of languages in the training may also be examined. The language used in all training data files will be indicated in the file header and available for use.

All training data will have been collected over telephone channels.

The sex of each target speaker will be provided to systems in the test index file (see section 8.3).

For all training conditions, English language ASR transcriptions of all data will be provided along with the audio data. Systems may utilize this data as they wish. The acoustic data may be used alone, the transcriptions may be used alone, or all data may be used in combination.[5]

4.1.1 Excerpts

As discussed in section 2.2.1, one of the training conditions is an excerpt of a conversation containing approximately 10 seconds of estimated speech duration in the channel of interest. The actual duration of target speech will vary (so that the excerpts include only whole turns whenever possible) but the target speech duration will be constrained to lie in the range of 8-12 seconds.

4.1.2 Two-channel Conversations

As discussed in section 2.2.1, there will be training conditions consisting of one, three, and eight two-channel conversations of a given speaker. These will consist of approximately five minutes from an original six minute conversation, with an initial segment of about one minute excised. The excision point will be chosen so as not to include a partial speech turn.

4.1.3 Summed-channel Conversations

As discussed in section 2.2.1, one of the training conditions will consist of three summed-channel conversations, minus initial segments of about a minute each. Here the two sides of each conversation, in which both the target speaker and another speaker participate, will be summed together. Thus the challenge is to distinguish speech by the intended target speaker from speech by other participating speakers. To make this challenge feasible, the training conversations will be chosen so that each non-target speaker participates in only one conversation, while the target speaker participates in all three.

The difficulty of finding the target speaker's speech in the training data is affected by whether the other speaker in a training conversation is of the same or of the opposite sex as the target. Systems will not be provided with this information, but may use automatic gender detection techniques if they wish. Performance results will be examined as a function of how many of the three training conversations contain same-sex other speakers.

Note that an interesting contrast will exist between this training condition and that consisting of three two-channel conversations.

4.2 Test data

As discussed in section 2.2.2, there will be four test segment conditions. NIST is interested in examining how performance varies among these conditions for fixed training conditions.

Most of the test data will be in English, but some may be in Arabic, Mandarin, Russian, or Spanish. The language used in all test data files will be indicated in the file header and available for use.

For all test conditions, English language ASR transcriptions of the data will be provided along with the audio data. Systems may utilize this data as they wish. The acoustic data may be used alone, the ASR transcriptions may be used alone, or all data may be used in combination.[5]

4.2.1 Excerpts

As discussed in section 2.2.2, one of the test conditions is an excerpt of a conversation containing approximately 10 seconds of estimated speech duration in the channel of interest. The actual duration of target speech will vary (so that the excerpts include only whole turns whenever possible) but the target speech duration will be constrained to lie in the range of 8-12 seconds.

4.2.2 Two-channel Conversations

As discussed in section 2.2.2, one of the test conditions is a single two-channel conversation. (The channel of interest will be designated in the test index file – see section 8.3.) Each conversation will consist of approximately five minutes from an original six minute conversation, with an initial segment of about one minute excised. The excision point will be chosen so as not to include a partial speech turn.

---

[3] This means that an explicit speaker detection decision is required for each trial. Explicit decisions are required because the task of determining appropriate decision thresholds is a necessary part of any speaker detection system and is a challenging research problem in and of itself.

[4] Confidence scores for all of the trials in a given test will be pooled before plotting detection error tradeoff curves. Thus it is necessary to normalize these scores across speakers to achieve satisfactory detection performance.

[5] Note, however, that the ASR transcripts will all be generated by an English language recognizer, regardless of the actual language being spoken.

4.2.3 Summed-channel Conversations

As discussed in section 2.2.2, one of the test conditions is a single summed-channel conversation, minus an initial segment of about a minute. Here the two sides of the conversation will be summed together, and only one of the two speakers included may match a target speaker specified in a trial.

The difficulty of determining whether the target speaker speaks in the test conversation is affected by the sexes of the speakers in the test conversation. Systems will not be told whether the two test speakers are of the same or opposite sex, but automatic gender detection techniques may be used. Performance results will be examined with respect to whether one or both of the test speakers are of the same sex as the target. (For all trials there will be at least one speaker who is of the same sex as the target speaker.)

Note that an interesting contrast will exist between this condition and that consisting of a single two-channel conversation.

4.2.4 Auxiliary Microphone Conversations

As discussed in section 2.2.2, one of the test conditions is a two-channel conversation in which the channel of interest is an auxiliary microphone channel. The other channel will contain normal telephone data. As with the normal two-channel conversation test condition, about five minutes from an original six-minute conversation will be provided. The microphone data will be provided in single byte 8-bit μ-law form that matches the telephone data provided.

Several types of auxiliary microphones will be included in this data. Thus it will be possible to examine how performance is affected by whether or not test data is recorded over a telephone channel, and by the type of microphone used in non-telephone test data. The non-telephone data will include some or all of the following microphone types:

- Ear-bud/lapel mike
- Mini-boom mike
- Courtroom mike
- Conference room mike
- Distant mike
- Near-field mike
- PC stand mike
- Micro-cassette mike

Information on the microphone type used in each non-telephone test segment data will not be available to recognition systems.

These auxiliary microphone conversations will all be in English. ASR transcriptions will be provided as they are for all other calls. Note, however, that the ASR transcript will be produced using telephone data input rather than the auxiliary microphone signal.

4.3 Factors Affecting Performance

All trials will be *same-sex* trials. This means that the sex of the test segment speaker in the channel of interest (two-channel), or of at least one test segment speaker (summed-channel), will be the same as that of the target speaker model. Performance will be reported separately for males and females and also for both sexes pooled.

All trials involving telephone test segments will be *different-number* trials. This means that the telephone numbers, and presumably the telephone handsets, used in the training and the test data segments will be different from each other.

Past NIST evaluations have shown that the type of telephone handset and the type of telephone transmission channel used can have a great effect on speaker recognition performance. Factors of these types will be examined in this evaluation.

Telephone callers in the Mixer collection (see section 6) are asked to classify the transmission channel as one of the following types:

- Cellular
- Cordless
- Regular (i.e., land-line)

Telephone callers in the Mixer collection are also asked to classify the instrument used as one of the following types:

- Speaker-phone
- Head-mounted
- Ear-bud
- Regular (i.e., hand-held)

Performance will be examined as a function of the telephone transmission channel type and of the telephone instrument type in both the training and the test segment data.

4.4 Unsupervised Adaptation

As discussed in section 2.2.4, an unsupervised adaptation mode will be supported for each test. Performance with and without such adaptation will be compared for participants attempting tests with unsupervised adaptation.

4.5 Common Evaluation Condition

In each evaluation NIST specifies a common evaluation condition (a subset of trials in the core test that satisfy additional constraints) in order to better foster technical interactions and technology comparisons among sites. The performance results on these trials are treated as the basic official evaluation outcome. The common evaluation condition for the 2005 evaluation will be the subset of the trials in the core test that satisfy the following two conditions:

- The test segment and all of the training data for the target model are in English.
- The test segment and all of the training data are from regular (hand-held) telephone instruments.

4.6 Comparison with Previous Evaluations

In each evaluation it is of interest to compare performance results, particularly of the best performing systems, with those of previous evaluations. This is generally complicated by the fact that the evaluation conditions change in each successive evaluation. For the 2005 evaluation the summed-channel test conditions are essentially identical to ones used in 2004. The two-channel test conditions have changed in that both data channels are being provided this year, unlike last. It will be of interest if any participants offer contrastive systems showing the effect of having the extra data channel. And the auxiliary microphone test condition is, as noted previously, new for 2005. Nevertheless, for fifteen of the 2005 tests it will be possible to make a fairly direct comparison with a comparable 2004 test. Comparisons will also be made with the results of earlier evaluations for conditions most similar to those in this evaluation.

While the test conditions will match those of 2004, the test data will be different. In particular, the 2005 target speakers will all be different from those of the 2004 evaluation. The question always arises of to what extent are the performance differences due to

random differences in the test data sets. For example, are the target speakers in the current evaluation easier, or harder, on the average to recognize? To address this question, sites participating in the 2005 evaluation that also participated in 2004 are strongly encouraged to submit to NIST results for their (unmodified) 2004 systems run on the 2005 data for the same test conditions as in 2004. Such results will not count against the limit of three submissions per test condition (see section 7). Sites are also encouraged to "mothball" their 2005 systems for use in similar comparisons in future evaluations.

5 DEVELOPMENT DATA

The evaluation data for the 2004 evaluation will serve as the primary development data for this year's evaluation. This data is covered by the LDC license agreement noted in section 6. Please refer to last year's evaluation plan for details.[6]

A limited amount of auxiliary microphone data from the cross-channel Mixer collection will be provided to sites requesting it and indicating an interest in participating in any of the evaluation tests that involve this type of data.

Participating sites may use other speech corpora to which they have access for development. Such corpora should be described in the site's system description. The original Switchboard-1 corpus may be used, but note that an effort is being made to recruit a limited number of the speakers from that corpus to participate in the new data collection from which this year's evaluation data will be selected.

6 EVALUATION DATA

Both the target speaker training data and the test segment data will be all newly collected by the Linguistic Data Consortium (LDC) as part of the Mixer project. This project invited participating speakers to take part in numerous six-minute conversations on specified topics with strangers. The Fishboard platform used to collect this data automatically initiated calls to selected pairs of speakers for most of the conversations, while participating speakers also initiated some calls themselves, with the collection system contacting other speakers for them to converse with. Speakers were encouraged to use different telephone instruments for their initiated calls.

The conversational data for this evaluation, to be distributed to evaluation participants by NIST on DVD's, has not been publicly released. The LDC will provide a license agreement, which non-member participating sites must sign, governing the use of this data for the evaluation. The ASR transcript data, and any other auxiliary data which may be supplied, will be made available by NIST in electronic form to all registered participants.

All conversations will have been processed through echo canceling software before being used to create the evaluation training and test segments.

All training and test segments will be stored as 8-bit μ-law speech signals in separate SPHERE[7] files. The SPHERE header of each such file will contain some auxiliary information as well as the standard SPHERE header fields. This auxiliary information will include the language of the conversation and whether or not the data was recorded over a telephone line.

Most segments will be in English and recorded over a telephone line. The header will not contain information on the type of telephone transmission channel or the type of telephone instrument involved. Nor will the microphone type be identified for the auxiliary microphone test, as noted in section 4.2.4.

6.1 Excerpts

The 10-second two-channel excerpts to be used as training data or as test segments will be continuous segments from single conversations that are estimated to contain approximately 10 seconds of actual speech in the channel of interest. When both channels are channels of interest for different trials, then each will contain approximately 10 seconds of actual speech.

The number of training segments is expected *not to exceed 2000*. The number of test segments is expected *not to exceed 4000*.

6.2 Two-channel Conversations

The two-channel conversations to be used as training data or as test segments will be approximately five minutes in duration.

The number of conversations to be used for training is expected *not to exceed 10,000*. The number of these to be used to create speaker models based on a single conversation is expected *not to exceed 2000*. The numbers of models specified by 3 or 8 conversations are each expected *not to exceed 1200*.

The number of test segments is expected not to exceed *4000*.

6.3 Summed-channel Conversations

The summed-channel conversations to be used as training data or as test segments will be approximately five minutes in duration

The number of summed channel training conversations is expected *not to exceed 2400*. These will be used to specify *no more than 800* target speaker models. The number of summed-channel conversation test segments is expected *not to exceed 4000*.

6.4 Auxiliary Microphone Conversations

These two-channel conversations to be used as test segments will be approximately five minutes in duration.

The number of test segments is expected *not to exceed 2000*.

6.5 Number of Trials

The trials for each of the 20 speaker detection tests offered will be specified in separate index files. These will be text files in which each record specifies the model and a test segment for a particular trial. The number of trials in each test is expected not to exceed *50,000*.

7 EVALUATION RULES

In order to participate in the 2005 speaker recognition evaluation a site must submit complete results for the core test condition

[6] www.nist.gov/speech/tests/spk/2004/SRE-04_evalplan-v1a.pdf

[7] ftp://jaguar.ncsl.nist.gov/pub/sphere_2.6a.tar.Z

(without unsupervised adaptation) as specified in section 2.2.3.[8] Results for other tests are optional but strongly encouraged.

All participants must observe the following evaluation rules and restrictions in their processing of the evaluation data:

- Each decision is to be based only upon the specified test segment and target speaker model. Use of information about other test segments (except as permitted for the unsupervised adaptation mode condition) and/or other target speakers is **not** allowed.[9] For example:
  - Normalization over multiple test segments is **not** allowed, except as permitted for the unsupervised adaptation mode condition.
  - Normalization over multiple target speakers is **not** allowed.
  - Use of evaluation data for impostor modeling is **not** allowed, except as permitted for the unsupervised adaptation mode condition.
- If an unsupervised adaptation condition is included, the test segments for each model must be processed in the order specified.
- The use of manually produced transcripts or other human-produced information for training is **not** allowed.
- Knowledge of the sex of the *target* speaker (implied by data set directory structure as indicated below) **is** allowed. Note that no cross-sex trials are planned, but that summed-channel segments may involve either same sex or opposite sex speakers.
- Knowledge of the language used in all segments, which will be provided, is allowed.
- Knowledge of whether or not a segment involves telephone channel transmission is allowed.
- Knowledge of the telephone transmission channel type and of the telephone instrument type used in all segments is not allowed, except as determined by automatic means.
- Listening to the evaluation data, or any other human interaction with the data, is **not** allowed before all test results have been submitted. This applies to training data as well as test segments.
- Knowledge of any information available in the SPHERE header **is** allowed.

The following general rules about evaluation participation procedures will also apply for all participating sites:

- Access to past presentations – Each new participant that has signed up for, and thus committed itself to take part in, the upcoming evaluation and workshop will be able to receive, upon request, the CD of presentations that were presented at the preceding workshop.
- Limitation on submissions – Each participating site may submit results for up to three different systems per evaluation condition for official scoring by NIST. Results for systems using unsupervised adaptation and results for 2004 systems run on 2005 data will not count against this limit. Note that the answer keys will be distributed to sites by NIST shortly after the submission deadline. Thus each site may score for itself as many additional systems and/or parameter settings as desired.
- Attendance at workshop – Each evaluation participant is required to have one or more representatives at the evaluation workshop who must present there a meaningful description of its system(s). Evaluation participants failing to do so may be excluded from future evaluation participation.
- Dissemination of results
  - Participants may publish and otherwise disseminate their own results.
  - Participants may publish and otherwise disseminate anonymous charts, produced by NIST, of all system results for a condition.
  - Participants may not publish or otherwise disseminate the names or results of other participants without the explicit written permission of each such participant. Participants violating this rule may be excluded from future evaluations.

[8] It is imperative that results be complete for every test submission. A test submission is complete if and only if it includes a result for every trial in the test.

[9] This means that the technology is viewed as being "application-ready". Thus a system must be able to perform speaker detection simply by being trained on the training data for a specific target speaker and then performing the detection task on whatever speech segment is presented, without the (artificial) knowledge of other test data.

8 EVALUATION DATA SET ORGANIZATION

The organization of the evaluation data will be:

- A top level directory used as a unique label for the disk: "**sp05-NN**" where NN is a digit pair identifying the disk
- Under which there will be four sub-directories: "**train**", "**test**", "**trials**", and "**doc**"

8.1 train Subdirectory

The "**train**" directory contains three subdirectories:

- **data**: Contains the SPHERE formatted speech data used for training in each of the seven training conditions.
- **female**: Contains five training files that define the *female* models for each of the seven training conditions. (The format of these files is defined below.)
- **male**: Contains five training files that define the *male* models for each of the seven training conditions. (The format of these files is defined below.)

The five training files for both male and female models have similar structures. Each has one record per line, and each record contains two fields. The first field is the model identifier. The second includes a comma separated list of speech files (located in the "**data**" directory) that are to be used to train the model. For the 2-channel training conditions, each list item also specifies whether the target speaker's speech is on the "A" or the "B" channel of the speech file.

The five training files in each gender directory are named:

- "***10sec4w.trn***" for the 10 second two-channel training condition, an example record looks like:

```
3232 mrpv.sph:B
```

- "***1conv4w.trn***" for the 1 conversation two-channel training condition, an example record looks like:

```
4240 mrpz.sph:A
```

- "***3conv4w.trn***" for the 3 conversation two-channel training condition, an example record for this training condition looks like:

```
7211 mrpz.sph:B,hrtz.sph:A,nost.sph:B
```

- "***8conv4w.trn***" for the 8 conversation training condition.
- "***3conv2w.trn***" for the 3 conversation summed-channel training condition, an example record looks like:

```
3310 nrfs.sph,irts.sph,poow.sph
```

8.2 test Subdirectory

The "**test**" directory contains one subdirectory:

- **data**: This directory contains all the SPHERE formatted speech test data to be used for each of the four test segment conditions. The file names will be arbitrary ones of four characters along with a ".sph" extension.

8.3 trials Subdirectory

The "**trials**" directory contains twenty index files, one for each of the possible combinations of the five training conditions and four test segment types. These index files define the various evaluation tests. The naming convention for these index files will be "*TrainCondition-TestCondition*.ndx" where *TrainCondition*, refers to the training condition and whose models are defined in the corresponding training file. Possible values for *TrainCondition* are: 10sec4w, 1conv4w, 3conv4w, 8conv4w, and 3conv2w. "*TestCondition*" refers to the test segment condition. Possible values for *TestCondition* are: 10sec4w, 1conv4w, 1conv2w, and 1convmic.

Each record in a *TrainCondition-TestCondition.ndx* file contains four fields and defines a single trial. The first field is the model identifier. The second field identifies the gender of the model, either "*m*" or "*f*". The third field is the test segment under evaluation, located in the **test/data** directory. This test segment name will not include the .sph extension. The fourth field specifies the channel of the test segment speech of interest, either "A" or "B". (This will always be "A" for the summed channel test.) An example for the train on three conversations two-channel and test on one conversation two-channel index file "3conv4w-1conv2w.ndx" looks like: "`7211 m nrbw B`".

The records in these 20 files are ordered numerically by model identifier, and within each model's tests, chronologically by the recording dates of the test segments. Thus each index file specifies the processing order of the trials for each model. (This order of processing is mandatory when unsupervised adaptation is used.)

8.4 doc Subdirectory

This will contain text files that document the evaluation and the organization of the evaluation data. This evaluation plan document will be included.

9 SUBMISSION OF RESULTS

Sites participating in one or more of the speaker detection evaluation tests must report results for each test in its entirety. These results for each test condition (1 of the 20 test index files) must be provided to NIST in a single file using a standard ASCII format, with one record for each trial decision. The file name should be intuitively mnemonic and should be constructed as "SSS_N", where

- SSS identifies the site, and
- N identifies the system.

9.1 Format for Results

Each file record must document its decision with the target model identification, test segment identification, and decision information. Each record must contain eight fields, separated by white space and in the following order:

1. The training type of the test – **10sec4w**, **1conv4w**, **3conv4w**, **8conv4w**, or **3convs2w**
2. Adaptation mode. "**n**" for no adaptation and "**u**" for unsupervised adaptation.
3. The segment type of the test – **10sec4w**, **1conv4w**, **1conv2w**, or **1convmic**
4. The sex of the target speaker – **m** or **f**
5. The target model identifier
6. The test segment identifier
7. The decision – **t** or **f** (whether or not the target speaker is judged to match the speaker in the test segment)
8. The confidence score (an estimate of the probability that the test segment contains speech from the target speaker)

9.2 Means of Submission

Submissions may be made via email or via ftp. The appropriate addresses for submissions will be supplied to participants receiving evaluation data.

10 SYSTEM DESCRIPTION

A brief description of the system(s) (the algorithms) used to produce the results must be submitted along with the results, for each system evaluated. A single site may submit the results for up to three separate systems for evaluation for each particular test, not counting test results using unsupervised adaptation and not counting results for 2004 systems run on 2005 data. If results for more than one system are submitted for a test, however, the site must identify one system as the "primary" system for the test prior to performing the evaluation. Sites are welcome to present descriptions of and results for additional systems at the evaluation workshop.

For each system for which results are submitted, sites must report the CPU execution time that was required to process the evaluation data, as if the test were run on a single CPU. This should be reported separately for creating models from the training data and for processing the test segments, and may be reported either as absolute processing time or as a multiple of real-time for the data processed. The additional time required for unsupervised adaptation should be reported where relevant. Sites must also describe the CPU and the amount of memory used.

11 SCHEDULE

The deadline for signing up to participate in the evaluation is March 1, 2005.

The evaluation data set DVD's will be distributed by NIST on April 4, 2005.

The deadline for submission of evaluation results to NIST is April 28, 2005.

Evaluation results will be released to each site by NIST on May 5, 2005.

The deadline for site workshop presentations to be supplied to NIST in electronic form for inclusion in the workshop CD-ROM is May 27, 2005.

Registration and room reservations for the workshop must be received by (a date to be determined).

The follow-up workshop will be held on or about June 6-8, 2005 at a location to be designated in the eastern United States. All sites participating in the evaluation must have representatives in attendance to discuss their systems and results.

12 GLOSSARY

Test – A collection of trials constituting an evaluation component.

Trial – The individual evaluation unit involving a test segment and a hypothesized speaker.

Target (model) speaker – The hypothesized speaker of a test segment, one for whom a model has been created from training data.

Non-target (impostor) speaker – A hypothesized speaker of a test segment who is in fact not the actual speaker.

Segment speaker – The actual speaker in a test segment.

Target (true speaker) trial – A trial in which the actual speaker of the test segment *is in fact* the target (hypothesized) speaker of the test segment.

Non-target (impostor) trial – A trial in which the actual speaker of the test segment *is in fact not* the target (hypothesized) speaker of the test segment.

Turn – The interval in a conversation during which one participant speaks while the other remains silent.

The NIST Year 2006 Speaker Recognition Evaluation Plan

1 INTRODUCTION

The year 2006 speaker recognition evaluation is part of an ongoing series of yearly evaluations conducted by NIST. These evaluations are an important contribution to the direction of research efforts and the calibration of technical capabilities. They are intended to be of interest to all researchers working on the general problem of text independent speaker recognition. To this end the evaluation is designed to be simple, to focus on core technology issues, to be fully supported, and to be accessible to those wishing to participate.

The 2006 evaluation will reuse from the 2005 evaluation some of the conversational telephone speech data collected for the Mixer Corpus by the Linguistic Data Consortium using the "Fishboard" platform, and will use some additional unexposed data from this collection and some similar data collected more recently. Some unexposed or newly collected "multi-channel" data collected simultaneously from a number of auxiliary microphones will also be included. The data will be mostly English speech, but will include some speech in four additional languages.

The evaluation will include 15 different speaker detection tests defined by the duration and type of the training and test data. For each such test, an unsupervised adaptation mode will be offered in addition to the basic test.

The evaluation will be conducted in April and May of 2006. A follow-up workshop for evaluation participants to discuss research findings will be held late in June in San Juan, Puerto Rico, preceding the Odyssey 2006 Workshop there. Specific dates are listed in the Schedule (section 11).

Participation in the evaluation is invited for all sites that find the tasks and the evaluation of interest. Participating sites must follow the evaluation rules set forth in this plan and must be represented at the evaluation workshop. For more information, and to register to participate in the evaluation, please contact NIST.[1]

2 TECHNICAL OBJECTIVE

This evaluation focuses on speaker detection in the context of conversational telephone speech. The evaluation is designed to foster research progress, with the goals of:

- Exploring promising new ideas in speaker recognition.
- Developing advanced technology incorporating these ideas.
- Measuring the performance of this technology.

2.1 Task Definition

The year 2006 speaker recognition evaluation is limited to the broadly defined task of **speaker detection**. This has been NIST's basic speaker recognition task over the past ten years. The task is to determine whether a specified speaker is speaking during a given segment of conversational speech.

2.2 Task Conditions

The speaker detection task for 2006 is divided into 15 distinct and separate tests. Each of these tests involves one of five training conditions and one of four test conditions. One of these tests (see section 2.2.3) is designated as the core test. Participants must do the core test and may choose to do any one or more of the other tests. Results must be submitted for ***all*** trials included in each test for which any results are submitted. For each test, there will also be an optional unsupervised adaptation condition. Sites choosing the adaptation option for a test must also perform the test without adaptation to provide a baseline contrast.

2.2.1 Training Conditions

The training segments in the 2006 evaluation will be continuous conversational excerpts. As in the previous two years, there will be no prior removal of intervals of silence. Also, as in 2005, both sides of all two-channel conversations will be provided (to aid systems in echo cancellation, dialog analysis, etc.). For all two-channel segments, the channel containing the putative target speaker to be recognized will be identified.

The five training conditions to be included involve target speakers defined by the following training data:

1. A two-channel (4-wire) excerpt from a conversation estimated to contain approximately 10 seconds of speech of the target on its designated side (The NIST energy-based automatic speech detector will be used to estimate the duration of actual speech in the chosen excerpts.)
2. One two-channel (4-wire) conversation, of approximately five minutes total duration[2], with the target speaker channel designated.
3. Three two-channel (4-wire) conversations involving the target speaker on their designated sides
4. Eight two-channel (4-wire) conversations involving the target speaker on their designated sides
5. Three summed-channel (2-wire) conversations, formed by sample-by-sample summing of their two sides. Each of these conversations will include both the target speaker and another speaker. These three non-target speakers will all be distinct.

English language word transcripts, produced using an automatic speech recognition (ASR) system, will be provided for all training segments of each condition. These transcripts will, of course, be errorful, with word error rates typically in the range of 15-30%.

---

[1] Send email to speaker_poc@nist.gov, or call 301/975-3169. Each site must complete the registration process by signing and returning the registration form, which is available online at: . http://www.nist.gov/speech/tests/spk/sre-06_registration.pdf

[2] Each conversation side will consist of the last five minutes of a six-minute conversation. This will eliminate from the evaluation data the less-topical introductory dialogue, which is more likely to contain language that identifies the speakers.

2.2.2 Test Segment Conditions

The test segments in the 2006 evaluation will be continuous conversational excerpts. As in the past two years, there will be no prior removal of intervals of silence. Also, as in 2005, both sides of all two-channel conversations will be provided (to aid systems in echo cancellation, dialog analysis, etc.). For all two-channel segments, the channel containing the putative target speaker to be recognized will be identified.

The four test segment conditions to be included are the following:

1. A two-channel (4-wire) excerpt from a conversation estimated to contain approximately 10 seconds of speech of the putative target speaker on its designated side (The NIST energy-based automatic speech detector will be used to estimate the duration of actual speech in the chosen excerpts.)
2. A two-channel (4-wire) conversation, of approximately five minutes total duration, with the putative target speaker channel designated.
3. A summed-channel (2-wire) conversation formed by sample-by-sample summing of its two sides
4. A two-channel (4-wire) conversation, with the usual telephone speech replaced by auxiliary microphone data in the putative target speaker channel. This auxiliary microphone data will be supplied in 8 kHz 8-bit μ-law form.

English language word transcripts, produced using an automatic speech recognition (ASR) system, will be provided for all test segments of each condition.

2.2.3 Training/Segment Condition Combinations

The matrix of training and test segment condition combinations is shown in **Table 1**. Note that only 15 (out of 20) condition combinations will be included in this year's evaluation. Each test consists of a sequence of trials, where each trial consists of a target speaker, defined by the training data provided, and a test segment. The system must decide whether speech of the target speaker occurs in the test segment. The shaded box labeled "required" in **Table 1** is the **core test** for the 2006 evaluation. All participants are required to submit results for this test. Each participant may also choose to submit results for all, some, or none of the other 14 test conditions. For each test for which results are submitted, results for **all** trials must be included.

2.2.4 Unsupervised Adaptation Mode

The unsupervised adaptation mode allows systems to update themselves based on previous trial segments for the target model involved (up to and including the current trial segment). This is in contrast to the non-adaptive mode in which the system is static and the target (and background) speaker models are a function only of the target speaker training data. (The speaker models of course also benefit from speech data used and knowledge acquired during system development.)

In the unsupervised adaptation mode it is required that the trials for each target model be performed in the order given in the test index file (see section 8.3). The trials for each model will be grouped together, and the test segments for each of these target models will be listed in chronological order. Within the testing for each target model, the target (and background) models may be updated by the system after each trial using the test segment data processed thus far for that target model. However, the adaptation must be discarded and the system reset to its initial unadapted state whenever a new model is encountered in the test index file.

Table 1: Matrix of training and test segment conditions. The shaded entry is the required core test condition.

		Test Segment Condition			
		10 sec 2-chan	**1 conv 2-chan**	**1 conv summed-chan**	**1 conv aux mic**
Training Condition	**10 seconds 2-channel**	optional			
	1 conversation 2-channel	optional	**required**	optional	optional
	3 conversation 2-channel	optional	optional	optional	optional
	8 conversation 2-channel	optional	optional	optional	optional
	3 conversation summed-channel		optional	optional	

For each test performed in unsupervised adaptation mode results must also be submitted for that test in non-adaptive mode in order to provide a contrast between adaptive and non-adaptive performance. The unsupervised adaptation techniques used should be discussed in the system description (see section 10).

3 PERFORMANCE MEASURE

There will be a single basic cost model for measuring speaker detection performance, to be used for all speaker detection tests. For each test, a detection cost function will be computed over the sequence of trials provided. Each trial must be independently judged as "true" (the model speaker speaks in the test segment) or "false" (the model speaker does not speak in the test segment), and the correctness of these decisions will be tallied.[3]

This detection cost function is defined as a weighted sum of miss and false alarm error probabilities:

$$C_{Det} = C_{Miss} \times P_{Miss|Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1-P_{Target})$$

The parameters of this cost function are the relative costs of detection errors, C_{Miss} and $C_{FalseAlarm}$, and the *a priori* probability of the specified target speaker, P_{Target}. The parameter values in **Table 2** will be used as the primary evaluation of speaker recognition performance for all speaker detection tests.

[3] This means that an explicit speaker detection decision is required for each trial. Explicit decisions are required because the task of determining appropriate decision thresholds is a necessary part of any speaker detection system and is a challenging research problem in and of itself.

Table 2: Speaker Detection Cost Model Parameters for the primary evaluation decision strategy

C_{Miss}	$C_{FalseAlarm}$	P_{Target}
10	1	0.01

To improve the intuitive meaning of C_{Det}, it will be normalized by dividing it by the best cost that could be obtained without processing the input data (i.e., by either always accepting or always rejecting the segment speaker as matching the target speaker, whichever gives the lower cost):

$$C_{Default} = \min\begin{Bmatrix} C_{Miss} \times P_{Target}, \\ C_{FalseAlarm} \times (1 - P_{Target}) \end{Bmatrix}$$

and

$$C_{Norm} = C_{Det} / C_{Default}$$

In addition to the actual detection decision, a confidence score will also be required for each test hypothesis. This confidence score should reflect the system's estimate of the probability that the test segment contains speech from the target speaker. Higher confidence scores should indicate greater estimated probability that the target speaker's speech is present in the segment. The confidence scores will be used to produce *Detection Error Tradeoff (DET)* curves, in order to see how misses may be traded off against false alarms. Since these curves will pool all trials in each test for all target speakers, it is necessary to normalize the confidence scores across all target speakers.

The ordering of the confidence scores is all that matters for computing the detection cost function, which corresponds to a particular application defined by the parameters specified in section 3, and for plotting DET curves. But these scores are more informative, and can be used to serve any application, if they represent actual probability estimates. It is suggested that participants provide as scores estimated likelihood ratio values, which do not depend on the application parameters. In terms of the conditional probabilities for the observed data of a given trial relative to the alternative target and non-target hypotheses the likelihood ratio *(LR)* is given by:

LR = prob (data | target hyp.) / prob (data | non-target hyp.)

Sites are asked to specify if their scores may be interpreted as likelihood ratio estimates. If so, floating point format should probably be used for scores to avoid any truncation of small values to zero.

A further type of scoring and graphical presentation will be performed on submissions whose scores are declared to represent likelihood ratios. A log likelihood ratio (*llr*) based cost function, which is not application specific and may be given an information theoretic interpretation, is defined as follows:

$$C_{llr} = 1 / (2 * \log 2) * (\sum \log(1+1/s)/N_{TT}) + (\sum \log(1+s))/N_{NT})$$

where the first summation is over all target trials, the second is over all non-target trials, N_{TT} and N_{NT} are the total numbers of target and non-target trials, respectively, and s represents a trial's likelihood ratio score.[4] i

Graphs based on this cost function, somewhat analogous to DET curves, will also be included. These may serve to indicate the ranges of possible applications for which a system is or is not well calibrated.[5]

4 EVALUATION CONDITIONS

Performance will be measured, graphically presented, and analyzed, as discussed in section 3, over all the trials of each of the 15 tests specified in section 2, and over subsets of these trials of particular evaluation interest. Comparisons will be made of performance variation across the different training conditions and the different test segment conditions which define these tests. The effects of factors such as language, telephone transmission type, and microphone type, will be examined. The possible performance benefit of unsupervised adaptation will be considered. As in previous years, a common evaluation condition (a subset of the core test) will be defined. And comparisons will be made between this year's evaluation results and those of recent past years.

4.1 Training Data

As discussed in section 2.2.1, there will be five training conditions. NIST is interested in examining how performance varies among these conditions for fixed test segment conditions.

Most of the training data will be in English, but some training conversations involving bi-lingual speakers may be collected in Arabic, Mandarin, Russian, and Spanish. Thus it will then be possible to examine how performance is affected by whether or not the training language matches the language of the test data. For the training conditions involving multiple conversations, the effect of having a mix of languages in the training may also be examined. The language used in all training data files will be indicated in the file header and available for use.

All training data will have been collected over telephone channels.

The sex of each target speaker will be provided to systems in the test index file (see section 8.3).

For all training conditions, English language ASR transcriptions of all data will be provided along with the audio data. Systems may utilize this data as they wish. The acoustic data may be used alone, the transcriptions may be used alone, or all data may be used in combination.[6]

4.1.1 Excerpts

As discussed in section 2.2.1, one of the training conditions is an excerpt of a conversation containing approximately 10 seconds of

[4] This reasons for choosing this cost function, and its possible interpretations, are described in detail in the paper "Application-independent evaluation of speaker detection" in Computer Speech & Language, volume 20, issues 2-3, April-July 2006, pages 230-275, by Niko Brummer and Johan du Preez.

[5] See the discussion of *Applied Probability of Error (APE)* curves in the reference cited in the preceding footnote.

[6] Note, however, that the ASR transcripts will all be generated by an English language recognizer, regardless of the actual language being spoken.

estimated speech duration in the channel of interest. The actual duration of target speech will vary (so that the excerpts include only whole turns whenever possible) but the target speech duration will be constrained to lie in the range of 8-12 seconds.

4.1.2 Two-channel Conversations

As discussed in section 2.2.1, there will be training conditions consisting of one, three, and eight two-channel conversations of a given speaker. These will consist of approximately five minutes from an original six minute conversation, with an initial segment of about one minute excised. The excision point will be chosen so as not to include a partial speech turn.

4.1.3 Summed-channel Conversations

As discussed in section 2.2.1, one of the training conditions will consist of three summed-channel conversations, minus initial segments of about a minute each. Here the two sides of each conversation, in which both the target speaker and another speaker participate, will be summed together. Thus the challenge is to distinguish speech by the intended target speaker from speech by other participating speakers. To make this challenge feasible, the training conversations will be chosen so that each non-target speaker participates in only one conversation, while the target speaker participates in all three.

The difficulty of finding the target speaker's speech in the training data is affected by whether the other speaker in a training conversation is of the same or of the opposite sex as the target. Systems will not be provided with this information, but may use automatic gender detection techniques if they wish. Performance results will be examined as a function of how many of the three training conversations contain same-sex other speakers.

Note that an interesting contrast will exist between this training condition and that consisting of three two-channel conversations.

4.2 Test data

As discussed in section 2.2.2, there will be four test segment conditions. NIST is interested in examining how performance varies among these conditions for fixed training conditions.

Most of the test data will be in English, but some may be in Arabic, Mandarin, Russian, or Spanish. The language used in all test data files will be indicated in the file header and available for use.

For all test conditions, English language ASR transcriptions of the data will be provided along with the audio data. Systems may utilize this data as they wish. The acoustic data may be used alone, the ASR transcriptions may be used alone, or all data may be used in combination.[6]

4.2.1 Excerpts

As discussed in section 2.2.2, one of the test conditions is an excerpt of a conversation containing approximately 10 seconds of estimated speech duration in the channel of interest. The actual duration of target speech will vary (so that the excerpts include only whole turns whenever possible) but the target speech duration will be constrained to lie in the range of 8-12 seconds.

4.2.2 Two-channel Conversations

As discussed in section 2.2.2, one of the test conditions is a single two-channel conversation. (The channel of interest will be designated in the test index file – see section 8.3.) Each conversation will consist of approximately five minutes from an original six minute conversation, with an initial segment of about one minute excised. The excision point will be chosen so as not to include a partial speech turn.

4.2.3 Summed-channel Conversations

As discussed in section 2.2.2, one of the test conditions is a single summed-channel conversation, minus an initial segment of about a minute. Here the two sides of the conversation will be summed together, and only one of the two speakers included may match a target speaker specified in a trial.

The difficulty of determining whether the target speaker speaks in the test conversation is affected by the sexes of the speakers in the test conversation. Systems will not be told whether the two test speakers are of the same or opposite sex, but automatic gender detection techniques may be used. Performance results will be examined with respect to whether one or both of the test speakers are of the same sex as the target. (For all trials there will be at least one speaker who is of the same sex as the target speaker.)

Note that an interesting contrast will exist between this condition and that consisting of a single two-channel conversation.

4.2.4 Auxiliary Microphone Conversations

As discussed in section 2.2.2, one of the test conditions is a two-channel conversation in which the channel of interest is an auxiliary microphone channel. The other channel will contain normal telephone data. As with the normal two-channel conversation test condition, about five minutes from an original six-minute conversation will be provided. The microphone data will be provided in single byte 8-bit μ-law form that matches the telephone data provided.

Several types of auxiliary microphones will be included in this data. Thus it will be possible to examine how performance is affected by whether or not test data is recorded over a telephone channel, and by the type of microphone used in non-telephone test data. The non-telephone data will include some or all of the following microphone types:

- Ear-bud/lapel mike
- Mini-boom mike
- Courtroom mike
- Conference room mike
- Distant mike
- Near-field mike
- PC stand mike
- Micro-cassette mike

Information on the microphone type used in each non-telephone test segment data will not be available to recognition systems.

These auxiliary microphone conversations will all be in English. ASR transcriptions will be provided as they are for all other calls. Note, however, that the ASR transcript will be produced using telephone data input rather than the auxiliary microphone signal.

4.3 Factors Affecting Performance

All trials will be *same-sex* trials. This means that the sex of the test segment speaker in the channel of interest (two-channel), or of at least one test segment speaker (summed-channel), will be the same as that of the target speaker model. Performance will be reported separately for males and females and also for both sexes pooled.

All trials involving telephone test segments will be *different-number* trials. This means that the telephone numbers, and

presumably the telephone handsets, used in the training and the test data segments will be different from each other. (For some telephone conversational data collected at the sites collecting the auxiliary microphone data, information other than phone numbers may be used to establish that different handsets are used.)

Past NIST evaluations have shown that the type of telephone handset and the type of telephone transmission channel used can have a great effect on speaker recognition performance. Factors of these types will be examined in this evaluation to the extent that information of this type is available.

Telephone callers are generally asked to classify the transmission channel as one of the following types:

- Cellular
- Cordless
- Regular (i.e., land-line)

Telephone callers are generally also asked to classify the instrument used as one of the following types:

- Speaker-phone
- Head-mounted
- Ear-bud
- Regular (i.e., hand-held)

Performance will be examined, to the extent the information is available and the data sizes are sufficient, as a function of the telephone transmission channel type and of the telephone instrument type in both the training and the test segment data.

4.4 Unsupervised Adaptation

As discussed in section 2.2.4, an unsupervised adaptation mode will be supported for each test. Performance with and without such adaptation will be compared for participants attempting tests with unsupervised adaptation.

4.5 Common Evaluation Condition

In each evaluation NIST specifies a common evaluation condition (a subset of trials in the core test that satisfy additional constraints) in order to better foster technical interactions and technology comparisons among sites. The performance results on these trials are treated as the basic official evaluation outcome. The common evaluation condition for the 2006 evaluation will be the subset of the trials in the core test that satisfy the following condition:

- The test segment and all of the training data for the target model are in English.

Note that all transmission and instrument types will be included in the common evaluation condition this year.

4.6 Comparison with Previous Evaluations

In each evaluation it is of interest to compare performance results, particularly of the best performing systems, with those of previous evaluations. This is generally complicated by the fact that the evaluation conditions change in each successive evaluation. For the 2006 evaluation the test conditions are essentially identical to ones used in 2005, and most are similar to ones used in 2004. Thus it will be possible to make fairly direct comparisons between 2006 and 2005 and even 2004 tests. Comparisons may also be made with the results of earlier evaluations for conditions most similar to those in this evaluation.

While the test conditions will match those used previously, the test data will be partially different. The 2006 target speakers will all be different from those of the 2004 evaluation, but will include many of the same speakers as in 2005. The question always arises of to what extent are the performance differences due to random differences in the test data sets. For example, are the new target speakers in the current evaluation easier, or harder, on the average to recognize? To help address this question, sites participating in the 2006 evaluation that also participated in 2004 or 2005 are strongly encouraged to submit to NIST results for their (unmodified) 2004 or 2005 systems run on the 2006 data for the same test conditions as previously. Such results will not count against the limit of three submissions per test condition (see section 7). Sites are also encouraged to "mothball" their 2006 systems for use in similar comparisons in future evaluations.

5 Development Data

The evaluation data for the 2004 evaluation will serve as the primary development data for this year's evaluation. This data is covered by the LDC license agreement noted in section 6. Please refer to the 2004 evaluation plan for details.[7]

All of the cross-channel microphone speech data used in the 2005 evaluation, and all of the telephone data involving the speakers of this microphone data, will also be available as development data for the 2006 evaluation. NIST will be making this data available as a package, and it will be covered by the LDC license agreement as well.

Participating sites may use other speech corpora to which they have access for development. Such corpora should be described in the site's system description. The original Switchboard-1 corpus may be used, but note that an effort is being made to recruit a limited number of the speakers from that corpus to participate in the new data collection from which this year's evaluation data will be selected.

6 Evaluation Data

Both the target speaker training data and the test segment data will have been collected by the Linguistic Data Consortium (LDC) as part of the Mixer project or in more recent similar collections. The Mixer project invited participating speakers to take part in numerous six-minute conversations on specified topics with strangers. The Fishboard platform used to collect this data automatically initiated calls to selected pairs of speakers for most of the conversations, while participating speakers also initiated some calls themselves, with the collection system contacting other speakers for them to converse with. Speakers were encouraged to use different telephone instruments for their initiated calls.

The conversational data for this evaluation will be distributed to evaluation participants by NIST on a firewire drive. The LDC provides a license agreement[8], which non-member participating sites must sign, governing the use of this data for the evaluation. The ASR transcript data, and any other auxiliary data which may be supplied, will be made available by NIST in electronic form to all registered participants.

All conversations will have been processed through echo canceling software before being used to create the evaluation training and test segments.

---

[7] www.nist.gov/speech/tests/spk/2004/SRE-04_evalplan-v1a.pdf

[8] Available online at http://www.nist.gov/speech/tests/spk/2006/

All training and test segments will be stored as 8-bit μ-law speech signals in separate SPHERE[9] files. The SPHERE header of each such file will contain some auxiliary information as well as the standard SPHERE header fields. This auxiliary information will include the language of the conversation and whether or not the data was recorded over a telephone line.

Most segments will be in English and recorded over a telephone line. The header will not contain information on the type of telephone transmission channel or the type of telephone instrument involved. Nor will the microphone type be identified for the auxiliary microphone test, as noted in section 4.2.4.

6.1 Excerpts

The 10-second two-channel excerpts to be used as training data or as test segments will be continuous segments from single conversations that are estimated to contain approximately 10 seconds of actual speech in the channel of interest. When both channels are channels of interest for different trials, then each will contain approximately 10 seconds of actual speech.

The number of training segments is expected *not to exceed 2000*. The number of test segments is expected *not to exceed 4000*.

6.2 Two-channel Conversations

The two-channel conversations to be used as training data or as test segments will be approximately five minutes in duration.

The number of conversations to be used for training is expected *not to exceed 10,000*. The number of speaker models based on a single conversation, and the numbers of models specified by 3 or by 8 conversations are each expected *not to exceed 2000*.

The number of test segments is expected not to exceed *4000*.

6.3 Summed-channel Conversations

The summed-channel conversations to be used as training data or as test segments will be approximately five minutes in duration

The number of summed channel training conversations is expected *not to exceed 2400*. These will be used to specify *no more than 800* target speaker models. The number of summed-channel conversation test segments is expected *not to exceed 4000*.

6.4 Auxiliary Microphone Conversations

These two-channel conversations to be used as test segments will be approximately five minutes in duration.

The number of test segments is expected *not to exceed 2000*.

6.5 Number of Trials

The trials for each of the speaker detection tests offered will be specified in separate index files. These will be text files in which each record specifies the model and a test segment for a particular trial. The number of trials in each test is expected not to exceed *75,000*.

7 Evaluation Rules

In order to participate in the 2006 speaker recognition evaluation a site must submit complete results for the core test condition (without unsupervised adaptation) as specified in section 2.2.3.[10] Results for other tests are optional but strongly encouraged.

All participants must observe the following evaluation rules and restrictions in their processing of the evaluation data:

- Each decision is to be based only upon the specified test segment and target speaker model. Use of information about other test segments (except as permitted for the unsupervised adaptation mode condition) and/or other target speakers is **not** allowed.[11] For example:
  - Normalization over multiple test segments is **not** allowed, except as permitted for the unsupervised adaptation mode condition.
  - Normalization over multiple target speakers is **not** allowed.
  - Use of evaluation data for impostor modeling is **not** allowed, except as permitted for the unsupervised adaptation mode condition.
- If an unsupervised adaptation condition is included, the test segments for each model must be processed in the order specified.
- The use of manually produced transcripts or other human-produced information for training is **not** allowed.
- Knowledge of the sex of the *target* speaker (implied by data set directory structure as indicated below) **is** allowed. Note that no cross-sex trials are planned, but that summed-channel segments may involve either same sex or opposite sex speakers.
- Knowledge of the language used in all segments, which will be provided, is allowed.
- Knowledge of whether or not a segment involves telephone channel transmission is allowed.
- Knowledge of the telephone transmission channel type and of the telephone instrument type used in all segments is not allowed, except as determined by automatic means.
- Listening to the evaluation data, or any other human interaction with the data, is **not** allowed before all test results have been submitted. This applies to training data as well as test segments.
- Knowledge of any information available in the SPHERE header **is** allowed.

The following general rules about evaluation participation procedures will also apply for all participating sites:

- Access to past presentations – Each new participant that has signed up for, and thus committed itself to take part in, the upcoming evaluation and workshop will be able to receive, upon request, the CD of presentations that were presented at the preceding workshop.
- Limitation on submissions – Each participating site may submit results for up to three different systems per evaluation

[9] ftp://jaguar.ncsl.nist.gov/pub/sphere_2.6a.tar.Z

[10] It is imperative that results be complete for every test submission. A test submission is complete if and only if it includes a result for every trial in the test.

[11] This means that the technology is viewed as being "application-ready". Thus a system must be able to perform speaker detection simply by being trained on the training data for a specific target speaker and then performing the detection task on whatever speech segment is presented, without the (artificial) knowledge of other test data.

condition for official scoring by NIST. Results for systems using unsupervised adaptation and results for earlier year systems run on 2006 data will not count against this limit. Note that the answer keys will be distributed to sites by NIST shortly after the submission deadline. Thus each site may score for itself as many additional systems and/or parameter settings as desired.

- Attendance at workshop – Each evaluation participant is required to have one or more representatives at the evaluation workshop who must present there a meaningful description of its system(s). Evaluation participants failing to do so may be excluded from future evaluation participation.
- Dissemination of results
  - Participants may publish and otherwise disseminate their own results.
  - Participants may publish and otherwise disseminate anonymous charts, produced by NIST, of all system results for a condition.
  - Participants may not publish or otherwise disseminate the names or results of other participants without the explicit written permission of each such participant. Participants violating this rule may be excluded from future evaluations.

8 EVALUATION DATA SET ORGANIZATION

The organization of the evaluation data will be:

- A top level directory used as a unique label for the disk: "**sp06-NN**" where NN is a digit pair identifying the disk
- Under which there will be four sub-directories: "**train**", "**test**", "**trials**", and "**doc**"

8.1 train Subdirectory

The "**train**" directory contains three subdirectories:

- **data**: Contains the SPHERE formatted speech data used for training in each of the seven training conditions.
- **female**: Contains five training files that define the *female* models for each of the seven training conditions. (The format of these files is defined below.)
- **male**: Contains five training files that define the *male* models for each of the seven training conditions. (The format of these files is defined below.)

The five training files for both male and female models have similar structures. Each has one record per line, and each record contains two fields. The first field is the model identifier. The second includes a comma separated list of speech files (located in the "**data**" directory) that are to be used to train the model. For the 2-channel training conditions, each list item also specifies whether the target speaker's speech is on the "A" or the "B" channel of the speech file.

The five training files in each gender directory are named:

- "***10sec4w.trn***" for the 10 second two-channel training condition, an example record looks like:

  `3232 mrpv.sph:B`
- "***1conv4w.trn***" for the 1 conversation two-channel training condition, an example record looks like:

  `4240 mrpz.sph:A`
- "***3conv4w.trn***" for the 3 conversation two-channel training condition, an example record for this training condition looks like:

  `7211 mrpz.sph:B,hrtz.sph:A,nost.sph:B`
- "***8conv4w.trn***" for the 8 conversation training condition.
- "***3conv2w.trn***" for the 3 conversation summed-channel training condition, an example record looks like:

  `3310 nrfs.sph,irts.sph,poow.sph`

8.2 test Subdirectory

The "**test**" directory contains one subdirectory:

- **data**: This directory contains all the SPHERE formatted speech test data to be used for each of the four test segment conditions. The file names will be arbitrary ones of four characters along with a ".sph" extension.

8.3 trials Subdirectory

The "**trials**" directory contains twenty index files, one for each of the possible combinations of the five training conditions and four test segment types. These index files define the various evaluation tests. The naming convention for these index files will be "*TrainCondition-TestCondition*.ndx" where *TrainCondition*, refers to the training condition and whose models are defined in the corresponding training file. Possible values for *TrainCondition* are: 10sec4w, 1conv4w, 3conv4w, 8conv4w, and 3conv2w. "*TestCondition*" refers to the test segment condition. Possible values for *TestCondition* are: 10sec4w, 1conv4w, 1conv2w, and 1convmic.

Each record in a *TrainCondition-TestCondition.ndx* file contains four fields and defines a single trial. The first field is the model identifier. The second field identifies the gender of the model, either "*m*" or "*f*". The third field is the test segment under evaluation, located in the **test/data** directory. This test segment name will not include the .sph extension. The fourth field specifies the channel of the test segment speech of interest, either "A" or "B". (This will always be "A" for the summed channel test.) An example for the train on three conversations two-channel and test on one conversation two-channel index file "3conv4w-1conv2w.ndx" looks like: "`7211 m nrbw B`".

The records in these 20 files are ordered numerically by model identifier, and within each model's tests, chronologically by the recording dates of the test segments. Thus each index file specifies the processing order of the trials for each model. (This order of processing is mandatory when unsupervised adaptation is used.)

8.4 doc Subdirectory

This will contain text files that document the evaluation and the organization of the evaluation data. This evaluation plan document will be included.

9 SUBMISSION OF RESULTS

Sites participating in one or more of the speaker detection evaluation tests must report results for each test in its entirety. These results for each test condition (1 of the xx test index files) must be provided to NIST in a single file using a standard ASCII format, with one record for each trial decision. The file name should be intuitively mnemonic and should be constructed as "SSS_N", where

- SSS identifies the site, and
- N identifies the system.

9.1 Format for Results

Each file record must document its decision with the target model identification, test segment identification, and decision information. Each record must contain nine fields, separated by white space and in the following order:

1. The training type of the test – **10sec4w**, **1conv4w**, **3conv4w**, **8conv4w**, or **3convs2w**
2. Adaptation mode. "**n**" for no adaptation and "**u**" for unsupervised adaptation.
3. The segment type of the test – **10sec4w**, **1conv4w**, **1conv2w**, or **1convmic**
4. The sex of the target speaker – **m** or **f**
5. The target model identifier
6. The test segment identifier
7. The test segment channel of interest, either "a" or "b"
8. The decision – **t** or **f** (whether or not the target speaker is judged to match the speaker in the test segment)
9. The confidence score (where larger scores indicate greater likelihood that the test segment contains speech from the target speaker)

9.2 Means of Submission

Submissions may be made via email or via ftp. The appropriate addresses for submissions will be supplied to participants receiving evaluation data. Sites should also indicate if it is the case that the confidence scores in a submission are to be interpreted as likelihood ratios.

10 System Description

A brief description of the system(s) (the algorithms) used to produce the results must be submitted along with the results, for each system evaluated. A single site may submit the results for up to three separate systems for evaluation for each particular test, not counting test results using unsupervised adaptation and not counting results for 2004 or 2005 systems run on the 2006 data. If results for more than one system are submitted for a test, however, the site must identify one system as the "primary" system for the test prior to performing the evaluation. Sites are welcome to present descriptions of and results for additional systems at the evaluation workshop.

For each system for which results are submitted, sites must report the CPU execution time that was required to process the evaluation data, as if the test were run on a single CPU. This should be reported separately for creating models from the training data and for processing the test segments, and may be reported either as absolute processing time or as a multiple of real-time for the data processed. The additional time required for unsupervised adaptation should be reported where relevant. Sites must also describe the CPU and the amount of memory used.

11 Schedule

The deadline for signing up to participate in the evaluation is March 27, 2006.

The evaluation data set will be distributed by NIST so as to arrive at participating sites on April 24, 2006.

The deadline for submission of evaluation results to NIST is May 14, 2006 at 11:59 PM, Washington time.

Initial evaluation results will be released to each site by NIST on May 22, 2006.

The deadline for site workshop presentations to be supplied to NIST in electronic form for inclusion in the workshop CD-ROM is (a date to be determined).

Registration and room reservations for the workshop must be received by (a date to be determined).

The follow-up workshop will be held June 25-27, 2006 at the Ritz Carlton Hotel in San Juan, Puerto Rico in conjunction with the IEEE Odyssey 2006 Speaker and Language Recognition Workshop. All sites participating in the evaluation must have one or more representatives in attendance to discuss their systems and results.

12 Glossary

Test – A collection of trials constituting an evaluation component.

Trial – The individual evaluation unit involving a test segment and a hypothesized speaker.

Target (model) speaker – The hypothesized speaker of a test segment, one for whom a model has been created from training data.

Non-target (impostor) speaker – A hypothesized speaker of a test segment who is in fact not the actual speaker.

Segment speaker – The actual speaker in a test segment.

Target (true speaker) trial – A trial in which the actual speaker of the test segment *is in fact* the target (hypothesized) speaker of the test segment.

Non-target (impostor) trial – A trial in which the actual speaker of the test segment *is in fact not* the target (hypothesized) speaker of the test segment.

Turn – The interval in a conversation during which one participant speaks while the other remains silent.

The NIST Year 2008 Speaker Recognition Evaluation Plan

1 INTRODUCTION

The year 2008 speaker recognition evaluation is part of an ongoing series of evaluations conducted by NIST. These evaluations are an important contribution to the direction of research efforts and the calibration of technical capabilities. They are intended to be of interest to all researchers working on the general problem of text independent speaker recognition. To this end the evaluation is designed to be simple, to focus on core technology issues, to be fully supported, and to be accessible to those wishing to participate.

The 2008 evaluation will be distinguished from the most recent evaluations, in particular those in 2005 and 2006, by including in the training and test conditions for the core (required) test not only conversational telephone speech data but also conversational speech data (of comparable duration) recorded over a microphone channel involving an interview scenario, and additionally, for the test condition, conversational telephone speech recorded over a microphone channel. Systems will know whether each segment comes from a telephone or a microphone channel, and whether it involves the interview scenario or an ordinary telephone conversation, but will be required to process trials involving all segments of each type. Submitted results will be scored after the fact to determine performance levels for telephone data, for microphone data of different conversational styles and microphone types, and for differing combinations of training and test data.

The optional tests this year will include, in addition to the training and test conditions of recent evaluations, a condition involving longer duration segments of interview data recorded over microphone channels.

The 2008 evaluation will not reuse data from previous evaluations, but some of the target speakers of the 2006 evaluation may reappear in the 2008 evaluation data. Target speakers from evaluations prior to 2006 will not be used in the 2008 evaluation data.

As in recent evaluations, some of the speakers in the telephone conversational data will be bilingual and their evaluation data may include speech in a language other than English as well as speech in English. The microphone recorded interview data will all be in English.

The evaluation will include 13 different speaker detection tests defined by the duration and type of the training and test data. For each such test, an unsupervised adaptation mode will be offered in addition to the basic test.

The evaluation will be conducted in April and May of 2008. A follow-up workshop for evaluation participants to discuss research findings will be held in June in Montreal, Quebec, Canada. Specific dates are listed in the Schedule (section 11).

Participation in the evaluation is invited for all sites that find the tasks and the evaluation of interest. Participating sites must follow the evaluation rules set forth in this plan and must be represented at the evaluation workshop. For more information, and to register to participate in the evaluation, please contact NIST.[1]

2 TECHNICAL OBJECTIVE

This evaluation focuses on speaker detection in the context of conversational speech. The evaluation is designed to foster research progress, with the goals of:

- Exploring promising new ideas in speaker recognition.
- Developing advanced technology incorporating these ideas.
- Measuring the performance of this technology.

2.1 Task Definition

The year 2008 speaker recognition evaluation is limited to the broadly defined task of **speaker detection**. This has been NIST's basic speaker recognition task over the past twelve years. The task is to determine whether a specified speaker is speaking during a given segment of conversational speech.

2.2 Task Conditions

The speaker detection task for 2008 is divided into 13 distinct and separate tests. Each of these tests involves one of six training conditions and one of four test conditions. One of these tests (see section 2.2.3) is designated as the core test. Participants must do the core test and may choose to do any one or more of the other tests. Results must be submitted for ***all*** trials included in each test for which any results are submitted. For each test, there will also be an optional unsupervised adaptation condition. Sites choosing the adaptation option for a test must also perform the test without adaptation to provide a baseline contrast.

2.2.1 Training Conditions

The training segments in the 2008 evaluation will be continuous conversational excerpts. As in recent evaluations, there will be no prior removal of intervals of silence. Also, except for summed channel telephone conversations and long interview segments as described below, two separate conversation channels will be provided (to aid systems in echo cancellation, dialog analysis, etc.). For all such two-channel segments, the primary channel containing the target speaker to be recognized will be identified.

The six training conditions to be included involve target speakers defined by the following training data:

1. ***10-sec***: A two-channel excerpt from a telephone conversation estimated to contain approximately 10 seconds of speech of the target on its designated side. (An energy-based automatic speech detector will be used to estimate the duration of actual speech in the chosen excerpts.)

[1] Send email to speaker_poc@nist.gov, or call 301/975-3605. Each site must complete the registration process by signing and returning the registration form, which is available online at: . http://www.nist.gov/speech/tests/sre/2008/sre08_registration.pdf

2. ***short2***: One two-channel telephone conversational excerpt, of approximately five minutes total duration[2], with the target speaker channel designated *or* a microphone recorded conversational segment of approximately three minutes total duration involving the target speaker and an interviewer. For the interview segments most of the speech will generally be spoken by the target speaker, and for consistency across the condition, a second zeroed out channel will be included.

3. ***3conv***: Three two-channel telephone conversational excerpts involving the target speaker on their designated sides.

4. ***8conv***: Eight two-channel telephone conversation excerpts involving the target speaker on their designated sides.

5. ***long***: A single channel microphone recorded conversational segment of eight minutes or longer duration involving the target speaker and an interviewer. Most of the speech will generally be spoken by the target speaker.

6. ***3summed***: Three summed-channel telephone conversational excerpts, formed by sample-by-sample summing of their two sides. Each of these conversations will include both the target speaker and another speaker. These three non-target speakers will all be distinct.

English language word transcripts, produced using an automatic speech recognition (ASR) system, will be provided for all training segments of each condition. These transcripts will, of course, be errorful, with English word error rates typically in the range of 15-30%. Note, however, that the ASR system will always be run on two separated channels, and run only once for those segments that have been simultaneously recorded over multiple channels, and the ASR transcripts provided may sometimes be superior to what current systems could provide for the actual channel involved. This is viewed as reasonable since ASR systems are expected to improve over time, and this evaluation is not intended to test ASR capabilities.

For the interview segments of the second and fifth conditions described above, the estimated intervals where the target speaker is speaking, as determined by an energy based segmenter utilizing the audio signals from lavalier microphones worn by each of the two speakers, will be provided. Systems may limit their processing to these intervals, or they may choose to process the full segments and do their own speaker separation processing.

2.2.2 Test Segment Conditions

The test segments in the 2008 evaluation will be continuous conversational excerpts. As in recent evaluations, there will be no prior removal of intervals of silence. Also, except for summed channel telephone conversations and long interview segments as described below, two separate conversation channels will be provided (to aid systems in echo cancellation, dialog analysis, etc.). For all such two-channel segments, the primary channel containing the putative target speaker to be recognized will be identified.

The four test segment conditions to be included are the following:

1. ***10-sec***: A two-channel excerpt from a telephone conversation estimated to contain approximately 10 seconds of speech of the putative target speaker on its designated side (An energy-based automatic speech detector will be used to estimate the duration of actual speech in the chosen excerpts.)

2. ***short3***: A two-channel telephone conversational excerpt, of approximately five minutes total duration, with the putative target speaker channel designated *or* a similar such telephone conversation but with the putative target channel being a (simultaneously recorded) microphone channel *or* a microphone recorded conversational segment of approximately three minutes total duration involving the putative target speaker and an interviewer. For the interview segments, most of the speech will generally be spoken by the target speaker, and for consistency across the condition, a second zeroed out channel will be included.

3. ***long***: A single channel microphone recorded conversational segment of eight minutes or longer duration involving the putative target speaker and an interviewer. Most of the speech will generally be spoken by the target speaker.

4. ***summed***: A summed-channel telephone conversation formed by sample-by-sample summing of its two sides

English language word transcripts, produced using an automatic speech recognition (ASR) system as described in section 2.2.2, will be provided for all test segments of each condition.

For the interview segments of the second and third conditions described above, the estimated intervals where the target speaker is speaking, as described in section 2.2.2, will be provided.

2.2.3 Training/Segment Condition Combinations

The matrix of training and test segment condition combinations is shown in **Table 1**. Note that only 13 (out of 24) condition combinations will be included in this year's evaluation. Each test consists of a sequence of trials, where each trial consists of a target speaker, defined by the training data provided, and a test segment. The system must decide whether speech of the target speaker occurs in the test segment. The shaded box labeled "required" in **Table 1** is the **core test** for the 2008 evaluation. All participants are required to submit results for this test. Each participant may also choose to submit results for all, some, or none of the other 12 test conditions. For each test for which results are submitted, results for **all** trials must be included.

2.2.4 Unsupervised Adaptation Mode

The unsupervised adaptation mode allows systems to update themselves based on previous trial segments for the target model involved (up to and including the current trial segment). This is in contrast to the non-adaptive mode in which the system is static and the target (and background) speaker models are a function only of the target speaker training data. (The speaker models of course also benefit from speech data used and knowledge acquired during system development.)

[2] Each conversation side will consist of five minutes of a longer conversation, and will exclude the first minute. This will eliminate from the evaluation data the less-topical introductory dialogue, which is more likely to contain language that identifies the speakers.

In the unsupervised adaptation mode it is required that the trials for each target model be processed in the order given in the test index file (see section 8.3x). The trials for each model will be grouped together, and the test segments for each of these target models will be listed in chronological order. Within the testing for each target model, the target (and background) models may be updated by the system after each trial using the test segment data processed thus far for that target model. No reprocessing of earlier trials is permitted. The adaptation, however, must be discarded and the system reset to its initial unadapted state whenever a new model is encountered in the test index file.

Table 1: Matrix of training and test segment conditions. The shaded entry is the required core test condition.

		Test Segment Condition			
		10sec	**short3**	**long**	**summed**
Training Condition	**10sec**	optional			
	short2	optional	**required**		optional
	3conv		optional		optional
	8conv	optional	optional		optional
	long		optional	optional	
	3summed		optional		optional

For each test performed in unsupervised adaptation mode results must also be submitted for that test in non-adaptive mode in order to provide a contrast between adaptive and non-adaptive performance. The unsupervised adaptation techniques used should be discussed in the system description (see section 10).

3 PERFORMANCE MEASURE

There will be a single basic cost model for measuring speaker detection performance, to be used for all speaker detection tests. For each test, a detection cost function will be computed over the sequence of trials provided. Each trial must be independently judged as "true" (the model speaker speaks in the test segment) or "false" (the model speaker does not speak in the test segment), and the correctness of these decisions will be tallied.[3]

This detection cost function is defined as a weighted sum of miss and false alarm error probabilities:

$$C_{\text{Det}} = C_{\text{Miss}} \times P_{\text{Miss|Target}} \times P_{\text{Target}} + C_{\text{FalseAlarm}} \times P_{\text{FalseAlarm|NonTarget}} \times (1-P_{\text{Target}})$$

The parameters of this cost function are the relative costs of detection errors, C_{Miss} and $C_{\text{FalseAlarm}}$, and the *a priori* probability of the specified target speaker, P_{Target}. The parameter values in **Table 2** will be used as the primary evaluation of speaker recognition performance for all speaker detection tests.

Table 2: Speaker Detection Cost Model Parameters for the primary evaluation decision strategy

C_{Miss}	$C_{\text{FalseAlarm}}$	P_{Target}
10	1	0.01

To improve the intuitive meaning of C_{Det}, it will be normalized by dividing it by the best cost that could be obtained without processing the input data (i.e., by either always accepting or always rejecting the segment speaker as matching the target speaker, whichever gives the lower cost):

$$C_{\text{Default}} = \min\left\{\begin{matrix} C_{\text{Miss}} \times P_{\text{Target}}, \\ C_{\text{FalseAlarm}} \times (1-P_{\text{Target}}) \end{matrix}\right\}$$

and

$$C_{\text{Norm}} = C_{\text{Det}} / C_{\text{Default}}$$

In addition to the actual detection decision, a confidence score will also be required for each test hypothesis. This confidence score should reflect the system's estimate of the probability that the test segment contains speech from the target speaker. Higher confidence scores should indicate greater estimated probability that the target speaker's speech is present in the segment. The confidence scores will be used to produce *Detection Error Tradeoff (DET)* curves, in order to see how misses may be traded off against false alarms. Since these curves will pool all trials in each test for all target speakers, it is necessary to normalize the confidence scores across all target speakers.

The ordering of the confidence scores is all that matters for computing the detection cost function, which corresponds to a particular application defined by the parameters specified in section 3, and for plotting DET curves. But these scores are more informative, and can be used to serve any application, if they represent actual probability estimates. It is suggested that participants provide as scores estimated log likelihood ratio values (using natural logarithms), which do not depend on the application parameters. In terms of the conditional probabilities for the observed data of a given trial relative to the alternative target and non-target hypotheses the likelihood ratio *(LR)* is given by:

LR = prob (data | target hyp.) / prob (data | non-target hyp.)

Sites are asked to specify if their scores may be interpreted as log likelihood ratio estimates.

A further type of scoring and graphical presentation will be performed on submissions whose scores are declared to represent log likelihood ratios. A log likelihood ratio (*llr*) based cost function, which is not application specific and may be given an information theoretic interpretation, is defined as follows:

$$C_{\text{llr}} = 1 / (2 * \log 2) * (\sum \log(1+1/s)/N_{TT}) + (\sum \log(1+s))/N_{NT})$$

where the first summation is over all target trials, the second is over all non-target trials, N_{TT} and N_{NT} are the total numbers of target and non-target trials, respectively, and s represents a trial's likelihood ratio.[4]

[3] This means that an explicit speaker detection decision is required for each trial. Explicit decisions are required because the task of determining appropriate decision thresholds is a necessary part of any speaker detection system and is a challenging research problem in and of itself.

[4] This reasons for choosing this cost function, and its possible interpretations, are described in detail in the paper "Application-independent evaluation of speaker detection" in Computer Speech

Graphs based on this cost function, somewhat analogous to DET curves, will also be included. These may serve to indicate the ranges of possible applications for which a system is or is not well calibrated.[5]

4 EVALUATION CONDITIONS

Performance will be measured, graphically presented, and analyzed, as discussed in section 3, over all the trials of each of the 13 tests specified in section 2, and over subsets of these trials of particular evaluation interest. Comparisons will be made of performance variation across the different training conditions and the different test segment conditions which define these tests. The effects of factors such as language, telephone transmission type, and microphone type, will be examined. The possible performance benefit of unsupervised adaptation will be considered. Several common evaluation conditions of interest, each a subset of the core test, will be defined. And relevant comparisons will be made between this year's evaluation results and those of recent past years.

4.1 Training Data

As discussed in section 2.2.1, there will be six training conditions. NIST is interested in examining how performance varies among these conditions for fixed test segment conditions.

Most of the training data will be in English, but some telephone training conversations involving bi-lingual speakers will be collected in a number of other languages. Thus it will then be possible to examine how performance is affected by whether or not the training language matches the language of the test data. For the training conditions involving multiple conversations, the effect of having a mix of languages in the training may also be examined. The language used in all training data files will be indicated in the file header and available for use.

Another performance factor of interest for English telephone conversations will be whether or not the speaker is a native U.S. English speaker. Information on this will not, however, be provided to systems.

The sex of each target speaker will be provided to systems in the test index file (see section 8.33).

For all training conditions, English language ASR transcriptions of all data will be provided along with the audio data. Systems may utilize this data as they wish. The acoustic data may be used alone, the transcriptions may be used alone, or all data may be used in combination.[6]

4.1.1 10-second Excerpts

As discussed in section 2.2.1, one of the training conditions is an excerpt of a telephone conversation containing approximately 10 seconds of estimated speech duration in the channel of interest. The actual duration of target speech will vary (so that the excerpts include only whole turns whenever possible) but the target speech duration will be constrained to lie in the range of 8-12 seconds.

4.1.2 Two-channel Conversations

As discussed in section 2.2.1, there will be training conditions consisting of one, three, and eight two-channel telephone conversational excerpts of a given speaker. (The first of these conditions will also include short interview segments.) These will each consist of approximately five minutes from a longer original conversation. The excision points will be chosen so as not to include partial speech turns.

4.1.3 Short Interview Segments

As discussed in section 2.2.1, one of the training conditions involves short conversational interview segments (along with single two-channel telephone conversations). These will each consist of approximately three minutes from a longer interview session. The excision points will be chosen so as not to include partial speech turns. Three minute segments are expected on average to include about as much speech from the speaker of interest as do five minute segments from telephone conversations. Two channels will be provided, the first from a microphone placed somewhere in the interview room, and the other a zero channel provided for consistency across the training condition. Information on the microphone type of the first channel will not be available to systems.

The microphone data will be provided in 8-bit μ-law form that matches the telephone data provided.

The speech of the short interview segments will all be in English.

4.1.4 Long Interview Segments

As discussed in section 2.2.1, one of the training conditions involves long conversational interview segments. These will each consist of eight minutes or more from an interview session. Any excision points will be chosen so as not to include partial speech turns. Only a single channels will be provided from a microphone placed somewhere in the interview room. Information on the microphone type will not be available to systems.

The microphone data will be provided in 8-bit μ-law form that matches the telephone data provided.

The speech of the long interview segments will all be in English.

4.1.5 Summed-channel Conversations

As discussed in section 2.2.1, one of the training conditions will consist of three summed-channel telephone conversation segments of about five minutes each. Here the two sides of each conversation, in which both the target speaker and another speaker participate, will be summed together. Thus the challenge is to distinguish speech by the intended target speaker from speech by other participating speakers. To make this challenge feasible, the training conversations will be chosen so that each non-target speaker participates in only one conversation, while the target speaker participates in all three.

The difficulty of finding the target speaker's speech in the training data is affected by whether the other speaker in a training conversation is of the same or of the opposite sex as the target. Systems will not be provided with this information, but may use automatic gender detection techniques if they wish. Performance results will be examined as a function of how many of the three training conversations contain same-sex other speakers.

& Language, volume 20, issues 2-3, April-July 2006, pages 230-275, by Niko Brummer and Johan du Preez.

[5] See the discussion of *Applied Probability of Error (APE)* curves in the reference cited in the preceding footnote.

[6] Note, however, that the ASR transcripts will all be generated by an English language recognizer, regardless of the actual language being spoken.

Note that an interesting contrast will exist between this training condition and that consisting of three two-channel conversations.

4.2 Test data

As discussed in section 2.2.2, there will be four test segment conditions. NIST is interested in examining how performance varies among these conditions for fixed training conditions.

Most of the test data will be in English, but some telephone speech will be in other languages, generally involving bilingual speakers who also speak English. The language used in all test data files will be indicated in the file header and available for use.

For all test conditions, English language ASR transcriptions of the data will be provided along with the audio data. Systems may utilize this data as they wish. The acoustic data may be used alone, the ASR transcriptions may be used alone, or all data may be used in combination.

4.2.1 Excerpts

As discussed in section 2.2.2, one of the test conditions is an excerpt of a telephone conversation containing approximately 10 seconds of estimated speech duration in the channel of interest. The actual duration of target speech will vary (so that the excerpts include only whole turns whenever possible) but the target speech duration will be constrained to lie in the range of 8-12 seconds.

4.2.2 Two-channel Conversations

As discussed in section 2.2.2, one of the test conditions involves single two-channel telephone conversational excerpts (or a short interview segment). Each excerpt will consist of approximately five minutes from a longer original conversation. The excision points will be chosen so as not to include partial speech turns.

4.2.3 Short Interview Segments

As discussed in section 2.2.2, one of the test conditions involves short conversational interview segments (along with two-channel telephone conversations). These will each consist of approximately three minutes from a longer interview session. The excision points will be chosen so as not to include partial speech turns. Three minute segments are expected on average to include about as much speech from the speaker of interest as do five minute segments from telephone conversations. Two channels will be provided, the first from a microphone placed somewhere in the interview room, and the other a zero channel provided for consistency across the test condition. Information on the microphone type of the first channel will not be available to systems.

The microphone data will be provided in 8-bit μ-law form that matches the telephone data provided.

The speech of the short interview segments will all be in English.

4.2.4 Long Interview Segments

As discussed in section 2.2.2, one of the training conditions involves long conversational interview segments. These will each consist of eight minutes or more from an interview session. Any excision points will be chosen so as not to include partial speech turns. Only a single channel will be provided from a microphone placed somewhere in the interview room. Information on the microphone type will not be available to systems.

The microphone data will be provided in 8-bit μ-law form that matches the telephone data provided.

The speech of the long interview segments will all be in English.

4.2.5 Summed-channel Conversations

As discussed in section 2.2.2, one of the test conditions is a single summed-channel conversational excerpt of about five minutes. Here the two sides of the conversation will be summed together, and one of the two speakers included may match a target speaker specified in a trial.

The difficulty of determining whether the target speaker speaks in the test conversation is affected by the sexes of the speakers in the test conversation. Systems will not be told whether the two test speakers are of the same or opposite sex, but automatic gender detection techniques may be used. Performance results will be examined with respect to whether one or both of the test speakers are of the same sex as the target. (For all trials there will be at least one speaker who is of the same sex as the target speaker.)

Note that an interesting contrast will exist between this condition and that consisting of a single two-channel conversation.

4.3 Factors Affecting Performance

All trials will be *same-sex* trials. This means that the sex of the test segment speaker in the channel of interest (two-channel), or of at least one test segment speaker (summed-channel), will be the same as that of the target speaker model. Performance will be reported separately for males and females and also for both sexes pooled.

This evaluation will focus on examining the effects of channel on recognition performance. This will include in particular the comparison of performance involving telephone segments with that involving microphone segments. Since each trial has a training and a test segment, four combinations may be examined here. For test segments only, performance on telephone channel telephone conversations will be compared with performance on microphone channel telephone conversations and with performance on microphone interview segments.

For trials involving microphone segments, it will be of interest to examine the effect of the different microphone types tested on performance, and the significance on performance of the match or mismatch of the training and test microphone types.

All trials involving telephone test segments will be *different-number* trials. This means that the telephone numbers, and presumably the telephone handsets, used in the training and the test data segments will be different from each other.

Past NIST evaluations have shown that the type of telephone handset and the type of telephone transmission channel used can have a great effect on speaker recognition performance. Factors of these types will be examined in this evaluation to the extent that information of this type is available.

Telephone callers are generally asked to classify the transmission channel as one of the following types:

- Cellular
- Cordless
- Regular (i.e., land-line)

Telephone callers are generally also asked to classify the instrument used as one of the following types:

- Speaker-phone
- Head-mounted
- Ear-bud
- Regular (i.e., hand-held)

Performance will be examined, to the extent the information is available and the data sizes are sufficient, as a function of the telephone transmission channel type and of the telephone instrument type in both the training and the test segment data.

4.4 Unsupervised Adaptation

As discussed in section 2.2.4, an unsupervised adaptation mode will be supported for each test. Performance with and without such adaptation will be compared for participants attempting tests with unsupervised adaptation.

4.5 Common Evaluation Condition

In each evaluation NIST has specified a common evaluation condition, a subset of trials in the core test that satisfy additional constraints, in order to better foster technical interactions and technology comparisons among sites. The performance results on these trials are treated as the basic official evaluation outcome. Because of the broader scope of the 2008 evaluation and the multiple types of audio data included in the core test, several common evaluation conditions will be specified. At the same time, it will not to appropriate to examine performance results over all trials of the core test lumped together. The common conditions to be considered will include the following subsets of all of the core test trials:

- All trials involving only interview speech in training and test
- All trials involving interview speech from the same microphone type in training and test
- All trials involving interview speech from different microphones types in training and test
- All trials involving interview training speech and telephone test speech
- All trials involving telephone training speech and non-interview microphone test speech
- All trials involving only telephone speech in training and test
- All trials involving only English language telephone speech in training and test
- All trials involving only English language telephone speech spoken by a native U.S. English speaker in training and test

4.6 Comparison with Previous Evaluations

In each evaluation it is of interest to compare performance results, particularly of the best performing systems, with those of previous evaluations. This is generally complicated by the fact that the evaluation conditions change in each successive evaluation. For the 2008 evaluation the test conditions involving conversational telephone speech (with test segments possibly recorded over a microphone channel) are essentially identical those used in 2006. Thus it will be possible to make fairly direct comparisons between 2008 and 2006 for these conditions. Comparisons may also be made with the results of earlier evaluations for conditions most similar to those in this evaluation.

While the test conditions will match those used previously, the test data will be different. The 2008 target speakers may include some used in the 2006 evaluation, but most will not have appeared previously. The question always arises of to what extent are the performance differences due to random differences in the test data sets. For example, are the new target speakers in the current evaluation easier, or harder, on the average to recognize? To help address this question, sites participating in the 2008 evaluation that also participated in 2006 are strongly encouraged to submit to NIST results for their (unmodified) 2006 (or earlier year) systems run on the 2008 data for the same test conditions as previously. Such results will not count against the limit of three submissions per test condition (see section 7). Sites are also encouraged to "mothball" their 2008 systems for use in similar comparisons in future evaluations.

5 Development Data

All of the previous NIST NRE evaluation data, covering evaluation years 1996-2006 may be used as development data for 2008. This data will be sent to prospective evaluation participants by the Linguistic Data Consortium on a hard drive provided the required license agreement is signed and submitted to the LDC.[7]

A limited amount of development data representing the interview scenario that is new for 2008 will also be made available. This will include interview sessions involving six speakers, which speakers will not be targets in the 2008 evaluation data. This data will be provided on DVD by request to all sites that have submitted the LDC license agreement described above.

Participating sites may use other speech corpora to which they have access for development. Such corpora should be described in the site's system description (section 10).

6 Evaluation Data

Both the target speaker training data and the test segment data, including the interview data, will have been collected by the Linguistic Data Consortium (LDC) as part of the various phases of its Mixer project.[8] The telephone collection part of the Mixer project invited participating speakers to take part in numerous conversations on specified topics with strangers. The Fishboard platform used to collect this data automatically initiated calls to selected pairs of speakers for most of the conversations, while participating speakers also initiated some calls themselves, with the collection system contacting other speakers for them to converse with. Speakers were encouraged to use different telephone instruments for their initiated calls.

The speech data for this evaluation will be distributed to evaluation participants by NIST on a firewire drive. The LDC license agreement described in section 5, which non-member sites must sign to participate in the evaluation, will govern the use of this data for the evaluation. The ASR transcript data, the estimated speech intervals of interview target speakers, and any other auxiliary data which may be supplied, will be made available by NIST in electronic form to all registered participants.

Since both channels of all telephone conversational data are provided, this data will not be processed through echo canceling

[7] Find link at http://www.nist.gov/speech/tests/sre/2008/index.html

[8] A description of the recent Mixer collections may be found at: http://papers.ldc.upenn.edu/Interspeech2007/Interspeech_2007_Mixer_345.pdf

software. Participants may choose to do such processing on their own.[9]

All training and test segments will be stored as 8-bit μ-law speech signals in separate SPHERE[10] files. The SPHERE header of each such file will contain some auxiliary information as well as the standard SPHERE header fields. This auxiliary information will include the language of the conversation, whether or not the data was recorded over a telephone line, and whether or not the data is from an interview session.

The header will not contain information on the type of telephone transmission channel or the type of telephone instrument involved. Nor will the microphone type be identified for the interview data, as noted in section 4.

The 10-second two-channel excerpts to be used as training data or as test segments will be continuous segments from single conversations that are estimated to contain approximately 10 seconds of actual speech in the channel of interest. When both channels are channels of interest for different trials, then each will contain approximately 10 seconds of actual speech.

The two-channel conversational excerpts to be used as training data or as test segments will be approximately five minutes in duration, while all the short interview segments will be approximately three minutes in duration. The primary channel of interest will be specified. Note that for the short interview segments the second, non-primary, channel contain all zeroes. Each segment will be identified as coming either from a telephone conversation or from an interview.

The single channel long interview segments to be used as training data or as test segments will be eight minutes or longer in duration.

The summed-channel conversational excerpts to be used as training data or as test segments will be approximately five minutes in duration

6.1 Numbers of Models

Table 3 provides estimated upper bounds on the numbers of models (target speakers) to be included in the evaluation for each training condition.

Table 3: Upper bounds on numbers of models by training condition

Training Condition	Max Models
10sec	2000
short2	4000
3conv	2000
8conv	2000
long	2000
3summed	2000

[9] One publicly available source of such software is http://www.ece.msstate.edu/research/isip/projects/speech/software/legacy/fir_echo_canceller/

[10] ftp://jaguar.ncsl.nist.gov/pub/sphere_2.6a.tar.Z

6.2 Numbers of Test Segments

Table 4 provides estimated upper bounds on the numbers of segment to be included in the evaluation for each test condition.

Table 4: Upper bounds on numbers of segments by test condition

Test Conditions	Max Segments
10sec	5000
short3	8000
long	2000
summed	5000

6.3 Numbers of Trials

The trials for each of the speaker detection tests offered will be specified in separate index files. These will be text files in which each record specifies the model and a test segment for a particular trial. The number of trials for each test condition is expected not to exceed *100,000*.

7 Evaluation Rules

In order to participate in the 2008 speaker recognition evaluation a site must submit complete results for the core test condition (without unsupervised adaptation) as specified in section 2.2.3.[11] Results for other tests are optional but strongly encouraged.

Participating sites, particularly those with limited internal resources, may utilize publicly available software designed to support the development of speaker detection algorithms.[12] The software used should be specified in the system description (section 10).

All participants must observe the following evaluation rules and restrictions in their processing of the evaluation data:

- Each decision is to be based only upon the specified test segment and target speaker model. Use of information about other test segments (except as permitted for the unsupervised adaptation mode condition) and/or other target speakers is **not** allowed.[13] For example:
  - Normalization over multiple test segments is **not** allowed, except as permitted for the unsupervised adaptation mode condition.
  - Normalization over multiple target speakers is **not** allowed.

[11] It is imperative that results be complete for every test submission. A test submission is complete if and only if it includes a decision and confidence score for every trial in the test.

[12] One publicly available source is the Mistral software for biometric applications developed at the University of Avignon along with other European sites: http://mistral.univ-avignon.fr/en/

[13] This means that the technology is viewed as being "application-ready". Thus a system must be able to perform speaker detection simply by being trained on the training data for a specific target speaker and then performing the detection task on whatever speech segment is presented, without the (artificial) knowledge of other test data.

  - Use of evaluation data for impostor modeling is **not** allowed, except as permitted for the unsupervised adaptation mode condition.
  - Speech data from past evaluations may be used for general algorithm development and for impostor modeling, but may not be used directly for modeling target speakers of the 2008 evaluation.
- If an unsupervised adaptation condition is included, the test segments for each model must be processed in the order specified.
- The use of manually produced transcripts or other human-created information is **not** allowed.
- Knowledge of the sex of the *target* speaker (implied by data set directory structure as indicated below) **is** allowed. Note that no cross-sex trials are planned, but that summed-channel segments may involve either same sex or opposite sex speakers.
- Knowledge of the language used in all segments, which will be provided, is allowed.
- Knowledge of whether or not a segment involves telephone channel transmission is allowed.
- Knowledge of the telephone transmission channel type and of the telephone instrument type used in all segments is not allowed, except as determined by automatic means.
- Listening to the evaluation data, or any other human interaction with the data, is **not** allowed before all test results have been submitted. This applies to training data as well as test segments.
- Knowledge of any information available in the SPHERE header **is** allowed.

The following general rules about evaluation participation procedures will also apply for all participating sites:

- Access to past presentations – Each new participant that has signed up for, and thus committed itself to take part in, the upcoming evaluation and workshop will be able to receive, upon request, the CD of presentations that were presented at the preceding workshop.
- Limitation on submissions – Each participating site may submit results for up to three different systems per evaluation condition for official scoring by NIST. Results for systems using unsupervised adaptation and results for earlier year systems run on 2008 data will not count against this limit. Note that the answer keys will be distributed to sites by NIST shortly after the submission deadline. Thus each site may score for itself as many additional systems and/or parameter settings as desired.
- Attendance at workshop – Each evaluation participant is required to have one or more representatives at the evaluation workshop who must present there a meaningful description of its system(s). Evaluation participants failing to do so will be excluded from future evaluation participation.
- Dissemination of results
  - Participants may publish or otherwise disseminate their own results.
  - NIST will generate and place on its web site charts of all system results for conditions of interest and, unlike past practice, these charts may contain the site names of the systems involved. Participants may publish or otherwise disseminate these charts, unaltered and with appropriate reference to their source.
  - Participants may not publish or otherwise disseminate their own comparisons of their performance results with those of other participants without the explicit written permission of each such participant. Participants violating this rule will be excluded from future evaluations.

8 EVALUATION DATA SET ORGANIZATION

The organization of the evaluation data will be:

- A top level directory used as a unique label for the disk: "**sp08-NN**" where NN is a digit pair identifying the disk
- Under which there will be four sub-directories: "**train**", "**test**", "**trials**", and "**doc**"

8.1 train Subdirectory

The "**train**" directory contains three subdirectories:

- **data**: Contains four subdirectories which in turn contain the SPHERE formatted speech data used for training in the 6 training conditions:
  - **10sec**: 10-second training segments
  - **short2**: single two-channel telephone conversation (5-minute) and short interview two-channel segments used for the short2, 3conv, and 8conv training conditions
  - **long**: long interview single channel training segments
  - **summed**: summed channel single conversation (5-minute) segments used for the 3summed training condition
- **female**: Contains 6 training files that define the *female* models for each of the 6 training conditions. (The format of these files is defined below.)
- **male**: Contains 6 training files that define the *male* models for each of the 6 training conditions. (The format of these files is defined below.)

The latter two sub-directories will be empty on the drives as distributed. The files described below will be distributed to evaluation participants by electronic means and may be saved here if desired.

The 6 training files for both male and female models have similar structures. Each has one record per line, and each record contains two fields. The first field is the model identifier. The second includes a comma separated list of speech files (located in the "**data**" directory) that are to be used to train the model. For the two channel training conditions, each list item also specifies whether the target speaker's speech is on the "A" or the "B" channel of the speech file.

The 6 training files in each gender directory are named:

- "***10sec.trn***" for the 10 second two channel training condition; an example record looks like:

```
32324 mrpvc.sph:B
```

- "***short2.trn***" for the 1 conversation/short interview two channel training condition; an example record looks like:

```
42403 mrpzt.sph:A
```

- "***3conv.trn***" for the 3 conversation two channel training condition; an example record for this training condition looks like:

```
72101
mrpzt.sph:B,hrtzp.sph:A,nosty.sph:B
```

- "***8conv.trn***" for the 8 conversation two channel training condition.
- "***long.trn***" for the long interview single channel training condition; an example record looks like:

  ```
  33105 nrkwd.sph
  ```
- "***3summed.trn***" for the 3 conversation summed-channel training condition; an example record looks like:

  ```
  50472 nrfsx.sph,irtts.sph,porow.sph
  ```

8.2 test Subdirectory

The "**test**" directory contains one subdirectory:

- **data**: This directory contains four sub-directories which in turn contain the SPHERE formatted speech test data to be used the four test segment conditions. The file names will be arbitrary ones of five characters along with a ".sph" extension.
  - **10sec**: 10-second test segments
  - **short3**: single two-channel telephone conversation (5-minute) test segments, similar test segments with the channel of interest microphone recorded, and short interview two-channel test segments
  - **long**: long interview single channel test segments
  - **summed**: summed channel single conversation (5-minute) test segments
  -

8.3 trials Subdirectory

This sub-directory will be empty on the drives as distributed. The index files described below will be distributed to evaluation participants by electronic means and may be saved here if desired.

The "**trials**" directory is designed to contain 13 index files, one for each of the evaluation tests. These index files define the various evaluation tests. The naming convention for these index files will be "*TrainCondition-TestCondition*.ndx" where *TrainCondition*, refers to the training condition and whose models are defined in the corresponding training file. Possible values for *TrainCondition* are: 10sec, short2, 3conv, 8conv, long, and 3summed. "*TestCondition*" refers to the test segment condition. Possible values for *TestCondition* are: 10sec, short3, long, and summed.

Each record in a *TrainCondition-TestCondition.ndx* file contains four fields and defines a single trial. The first field is the model identifier. The second field identifies the gender of the model, either "*m*" or "*f*". The third field is the test segment under evaluation, located in the **test/data** directory. This test segment name will not include the .sph extension. The fourth field specifies the channel of the test segment speech of interest, either "A" or "B". (This will always be "A" for the summed channel test condition and for interview test segments.) For example, for the train on three conversations two channel and test on one conversation/short interview index file "3conv-short3.ndx" a record looks like: "`72116 m nrbrw B`".

The records in these 13 files are ordered numerically by model identifier, and within each model's tests, by test segment type and chronologically by the recording dates of the test segments. Thus each index file specifies the processing order of the trials for each model. (This order of processing is mandatory when unsupervised adaptation is used.)

8.4 doc Subdirectory

This will contain text files that document the evaluation and the organization of the evaluation data. This evaluation plan document will be included.

9 SUBMISSION OF RESULTS

Sites participating in one or more of the speaker detection evaluation tests must report results for each test in its entirety. These results for each test condition (1 of the 13 test index files) must be provided to NIST in a single file using a standard ASCII format, with one record for each trial decision. The file name should be intuitively mnemonic and should be constructed as "SSS_N", where

- SSS identifies the site, and
- N identifies the system.

9.1 Format for Results

Each file record must document its decision with the target model identification, test segment identification, and decision information. Each record must contain nine fields, separated by white space and in the following order:

1. The training type of the test – **10sec**, **short2**, **3conv**, **8conv**, **long**, or **3summed**
2. Adaptation mode. "**n**" for no adaptation and "**u**" for unsupervised adaptation.
3. The segment type of the test – **10sec**, **short3**, **long**, or **summed**
4. The sex of the target speaker – **m** or **f**
5. The target model identifier
6. The test segment identifier
7. The test segment channel of interest, either "a" or "b"
8. The decision – **t** or **f** (whether or not the target speaker is judged to match the speaker in the test segment)
9. The confidence score (where larger scores indicate greater likelihood that the test segment contains speech from the target speaker)

9.2 Means of Submission

Submissions may be made via email or via ftp. The appropriate addresses for submissions will be supplied to participants receiving evaluation data. Sites should also indicate if it is the case that the confidence scores in a submission are to be interpreted as log likelihood ratios.

10 SYSTEM DESCRIPTION

A brief description of the system(s) (the algorithms) used to produce the results must be submitted along with the results, for each system evaluated. A single site may submit the results for up to three separate systems for evaluation for each particular test, not counting test results using unsupervised adaptation and not counting results for earlier year systems run on the 2008 data. If results for more than one system are submitted for a test, however, the site must identify one system as the "primary" system for the test prior to performing the evaluation. Sites are welcome to present descriptions of and results for additional systems at the evaluation workshop.

For each system for which results are submitted, sites must report the CPU execution time that was required to process the evaluation data, as if the test were run on a single CPU. This should be

reported separately for creating models from the training data and for processing the test segments, and may be reported either as absolute processing time or as a multiple of real-time for the data processed. The additional time required for unsupervised adaptation should be reported where relevant. Sites must also describe the CPU and the amount of memory used.

11 SCHEDULE

The deadline for signing up to participate in the evaluation is March 31, 2008.

The evaluation data set will be distributed by NIST so as to arrive at participating sites on April 7, 2008.

The deadline for submission of evaluation results to NIST is May 8, 2008 at 11:59 PM, Washington time.

Initial evaluation results will be released to each site by NIST on May 19, 2008.

The deadline for site workshop presentations to be supplied to NIST in electronic form for inclusion in the workshop CD-ROM is (a date to be determined).

Registration and room reservations for the workshop must be received by (a date to be determined).

The follow-up workshop will be held June 17-18, 2008 at McGill University in Montreal, Quebec, Canada. All sites participating in the evaluation must have one or more representatives in attendance to discuss their systems and results.

12 GLOSSARY

Test – A collection of trials constituting an evaluation component.

Trial – The individual evaluation unit involving a test segment and a hypothesized speaker.

Target (model) speaker – The hypothesized speaker of a test segment, one for whom a model has been created from training data.

Non-target (impostor) speaker – A hypothesized speaker of a test segment who is in fact not the actual speaker.

Segment speaker – The actual speaker in a test segment.

Target (true speaker) trial – A trial in which the actual speaker of the test segment *is in fact* the target (hypothesized) speaker of the test segment.

Non-target (impostor) trial – A trial in which the actual speaker of the test segment *is in fact not* the target (hypothesized) speaker of the test segment.

Turn – The interval in a conversation during which one participant speaks while the other remains silent.

The NIST Year 2012 Speaker Recognition Evaluation Plan

1 INTRODUCTION

The year 2012 speaker recognition evaluation (SRE12) is the next in an ongoing series of speaker recognition evaluations conducted by NIST. These evaluations serve to support speaker recognition research and to calibrate the performance of speaker recognition systems. They are intended to be of interest to all researchers working on the general problem of text independent speaker recognition. To this end the evaluation is designed to be simple, to focus on core technology issues, to be fully supported, and to be accessible to those wishing to participate.

The basic task in NIST's speaker recognition evaluations is speaker detection, i.e., to determine whether a specified target speaker is speaking during a given segment of speech. While the basic task in SRE12 remains unchanged, SRE12 task conditions represent a significant departure from previous NIST SRE's. In previous evaluations, the evaluation test set, which is released at the beginning of the evaluation period, has contained both the training data and the test data. In SRE12, however, most target speakers will be taken from previous SRE corpora, with the training data being provided to evaluation participants at the time of registration, well in advance of the evaluation period. Furthermore, in SRE12 the training data for each such target speaker comprises all of the data from previous SRE's, both training and test, and will include a fairly large number of speech segments taken from multiple recording sessions. Similar to SRE10, all of the speech in SRE12 is expected to be in English, though English may not be the first language of some of the speakers included.

Participation in the evaluation is invited for all who find the task and evaluation of interest and are able to comply with the evaluation rules set forth in this plan. Further, participants must be represented at the evaluation workshop, to be held in Orlando, Florida, USA on December 11-12, 2012. To register, please fill out and follow the instructions on the registration form.[1] .

2 TECHNICAL OBJECTIVE

This evaluation focuses on speaker detection in the context of conversational speech over multiple types of channels. The evaluation is designed to foster research progress, with the goals of:

- Exploring promising new ideas in speaker recognition.
- Developing advanced technology incorporating these ideas.
- Measuring the performance of this technology.

2.1 Task Definition

The year 2012 speaker recognition task is **speaker detection**, as described briefly in the introduction. This has been NIST's speaker recognition task over the past sixteen years. The task is to determine whether a specified target speaker is speaking during a given segment of speech. More explicitly, one or more samples of speech data from a speaker (referred to as the "target" speaker) are provided to the speaker recognition system. These samples are the "training" data. The system uses these data to create a "model" of the target speaker's speech. Then a sample of speech data is provided to the speaker recognition system. This sample is referred to as the "test" segment. Performance is judged according to how accurately the test segment is classified as containing (or not containing) speech from the target speaker.

SRE12 includes an optional evaluation of human-assisted speaker recognition (HASR12). The HASR12 task and evaluation is described in section 11.

In previous NIST evaluations the system output consisted of a detection decision and a score representing the system's confidence that the target speaker is speaking in the test segment. NIST has recently encouraged expressing the system output score as the natural logarithm of the estimated likelihood ratio, defined as:

$$LLR = \log(\text{pdf}(\text{data} \mid \text{target hyp.}) / \text{pdf}(\text{data} \mid \text{non-target hyp.}))$$

Because of the general community acceptance of using the log likelihood ratio as a score, in SRE12 NIST is requiring that the system output score for each trial be the natural logarithm of the likelihood ratio. Further, since the detection threshold may be determined from the likelihood ratio, system output in SRE12 will not include a detection decision.

2.2 Task Conditions

The speaker detection task for 2012 is divided into 9 distinct and separate tests (not counting the HASR test discussed in section 11). Each of these tests involves one of three training conditions and one of five test conditions. One of these tests is designated as the core test which all participants must complete (except for those doing only the HASR test). Participants may also choose to do one or more of the other tests. Results must be submitted for ***all*** trials in each test for which results are submitted.

In SRE12 knowledge of all targets is allowed in computing each trial's detection score. This differs from all previous SRE's. Previously systems were restricted to use only knowledge of the single target speaker that was specified as the trial target. To test the effect of this knowledge on system performance, the SRE12 evaluation data will also include data from new speakers (for the non-target trials), to provide a basis for comparison of performance under the two conditions (of having versus not having knowledge of non-target speakers).

All of the speech in SRE12 will be in English.

2.2.1 Training Conditions

Target speaker training data in SRE12 will comprise all of the speech data associated with the target speakers chosen from the LDC speaker recognition speech corpora used in previous SRE's. There will be no more than 2,250 target speakers. A list of target speakers will be supplied, along with the relevant LDC speech corpora, when participants register to participate in the SRE12 evaluation. In addition, some previously unexposed target speakers, along with their relevant speech training data, will be supplied at evaluation time. Some of these additional speakers may have only one training segment. It should be noted that no

[1]. http://nist.gov/itl/iad/mig/upload/registration_sre12-v0.pdf .
For more information, please send email to speaker_poc@nist.gov

additional restrictions are placed upon the use of these previously unexposed target speakers; in particular, knowledge of these targets is allowed in computing each trial's detection score.

The three training conditions to be included involve target speakers defined by the following data:

1. ***Core***: All speech data, including microphone and telephone channel recordings, available for each target speaker.
2. ***Telephone***: All telephone channel speech data available for each target speaker. This condition prohibits the use in any way of the microphone data from any of the designated target speakers. Microphone data from speakers other than those specified as target speakers may be used, for example, for background models, speech activity detection models, etc.
3. ***Microphone***: All microphone channel speech data available for each target speaker. This condition prohibits the use in any way of the telephone data from any of the designated target speakers. Telephone data from speakers other than those specified as target speakers may be used, for example, for background models, speech activity detection models, etc.

2.2.2 Test Segment Conditions

The test segments in the 2012 evaluation will be mostly excerpts of conversational telephone speech but may contain interviews. There will be one required and four optional test segment conditions:

1. ***Core***: One two-channel excerpt from a telephone conversation or interview, containing nominally between 20 and 160 seconds of target speaker speech. Some of these test segments will have additive noise imposed.
2. ***Extended***: The test segments will be the same as those used in ***Core***. The number of trials in Extended tests will exceed the number of trials in Core tests.
3. ***Summed***: A summed-channel excerpt from a telephone conversation or interview, containing nominally between 20 and 160 seconds of target speaker speech formed by sample-by-sample summing of its two sides.
4. ***Known***: The trial list for the known test segment condition will be the same as in ***Extended.*** The system should presume that all of the non-target trials are by known speakers.
5. ***Unknown***: The trial list for the unknown test segment condition will be the same as in ***Extended.*** The system should presume that all of the non-target trials are by unknown speakers.

2.2.3 Training/Test Segment Condition Combinations

The matrix of training and test segment condition combinations is shown in Table 1. Note that only 9 (out of 15) of the possible condition combinations will be included in the 2012 evaluation. Each test consists of a sequence of trials, where each trial consists of a target speaker, defined by the training data provided, and a test segment. The system must decide whether speech of the target speaker occurs in the test segment. The highlighted text labeled "required (Core Test)" in Table 1 is the **Core test** for the 2012 evaluation, and all participants (except those completing HASR only) are required to complete the core test. Participants are encouraged, but not required, to submit results for one or more of the other eight optional tests. For each test for which results are submitted, results for **all** trials must be included.

Table 1: Matrix of training and test segment conditions. The shaded entry is the required Core test

		Training Condition		
		Core	**Microphone**	**Telephone**
Test Segment Condition	**Core**	**required (Core test)**	optional	optional
	Extended	optional	optional	optional
	Summed	optional		
	Known	optional		
	Unknown	optional		

3 PERFORMANCE MEASURES

The primary performance measure for SRE12 will be a detection cost, defined as a weighted sum of miss and false alarm error probabilities. There are two significant changes from past practice regarding how this primary cost measure will be computed in SRE12:

- First, no detection decision output is needed because trial scores are required to be log likelihood ratios. Thus the detection threshold is a known function of the cost parameters, and so the trial detection decisions are determined simply by applying this threshold to the trials' log likelihood scores.
- Second, the primary cost measure in SRE12 will be a combination of two costs, one using the cost parameters from SRE10 and one using a greater target prior. This is intended to add to the stability of the cost measure and to increase the importance of good score calibration over a wider range of log likelihoods.

The cost function used in SRE12 to compute costs accounts separately for known and unknown non-target speakers:

$$C_{Det} = C_{Miss} \times P_{Target} \times P_{Miss|Target} + C_{FalseAlarm} \times (1-P_{Target}) \times (P_{FalseAlarm|KnownNonTarget} \times P_{Known} + P_{FalseAlarm|UnknownNonTarget} \times (1-P_{Known}))$$

The parameters of this performance measure are:

- C_{Miss}, the cost of a miss,
- $C_{FalseAlarm}$, the cost of a false alarm,
- P_{Target}, the *a priori* probability that the segment speaker is the target speaker[2], and
- P_{Known}, the *a priori* probability that the non-target speaker is one of the evaluation target speakers[3].

[2] Note that P_{Target}, the target prior used to compute system performance, is not the same as the prior probability of target trials in the corpus.

Table 2: Speaker Detection Cost Model Parameters

		C_{Miss}	C_{FA}	$P_{Target-A1}$	$P_{Target-A2}$	P_{Known}
Test Segment Condition	**Core Extended Summed**	1	1	0.01	0.001	0.5
	Known					1
	Unknown					0

To improve the intuitive meaning of C_{Det}, it will be normalized by dividing it by the best cost that could be obtained without knowledge of the input data:

$$C_{Norm} = C_{Det} / C_{Default}$$

where $C_{Default} = C_{Miss} \times P_{Target}$

Thus

$$C_{Norm} = P_{Miss|Target} + \beta \times \frac{P_{Known} \times P_{FalseAlarm|KnownNontarget} + (1 - P_{Known}) \times P_{FalseAlarm|UnknownNontarget}}{}$$

where $\beta = \frac{C_{FalseAlarm}}{C_{Miss}} \quad \frac{1-P_{target}}{P_{target}}$

Actual detection costs will be computed from the trial scores by applying detection thresholds of $\log(\beta)$ for the two values of β, with β_{A1} (for $P_{Target-A1}$) being 99 and β_{A2} (for $P_{Target-A2}$) being 999.

The primary cost measure for SRE12 is defined as:

$$C_{primary} = \frac{C_{Norm}\ \beta_{A1} + C_{Norm}\ \beta_{A2}}{2}$$

Also, a minimum detection cost will be computed by using the detection thresholds that minimize the detection cost.

In addition to the primary performance measure, an alternative, information theoretic measure will be computed that considers how well all scores represent the likelihood ratio and that penalizes for errors in score calibration. This performance measure is defined as:

$$C_{llr} = 1 / (2 * \log 2) * ((\sum \log(1+1/s)/N_{TT}) + (\sum \log(1+s))/N_{NT}))$$

where the first summation is over all target trials, the second is over all non-target trials, N_{TT} and N_{NT} are the total numbers of target and non-target trials, respectively, and s represents a trial's likelihood ratio.[4]

A useful variant of C_{llr} is to limit evaluation to the low false alarm region. The motivation for doing this is to improve the informative power of C_{llr} in the low false alarm region. This is important because the large majority of non-target scores, which are of no interest (since they are correctly rejected), nonetheless have a major influence on the computed value of C_{llr}. A simple way of focusing the low false alarm region is to limit the trials in the calculation of C_{llr} to only those for which P_{Miss} is greater than the minimum over the range of interest. A reasonable minimum value of P_{Miss}, given the current state of technology, is 10%. Using this value, this variant of C_{llr} may be called $C_{llr-M10}$.

In order to foster interest in speaker recognition performance measurement, NIST would like to encourage participants to propose additional performance measures for use in future NIST SRE's. Sites wishing to submit proposals should send email to speaker_poc@nist.gov for details.

4 EVALUATION CONDITIONS

Performance will be measured, graphically presented, and analyzed, as a function of various conditions of interest. These will include the training and test conditions.

For all training conditions, English language ASR transcriptions of all data will **NOT** be provided along with the audio data. This is a change from recent SRE's, where ASR transcripts were provided.

4.1.1 Two-channel Conversations

As mentioned in section 2.2.2, there will be test segments each consisting of an excerpt from a two-channel telephone conversation. These will vary in duration and amount of speech. The effect of longer or shorter segment durations on performance may be examined. The excision points will be chosen to minimize the likelihood of including partial speech turns.

The telephone channel data will be provided in 8-bit μ-law form that differs from the microphone data provided.

4.1.2 Interview Segments

As mentioned in section 2.2.2, there will be test segments each consisting of an excerpt from an interview. These will vary in duration and amount of speech. The effect of longer or shorter segment durations on performance may be examined. Two channels will be provided, the first from a microphone placed somewhere in the interview room, and the other from the interviewer's head mounted close-talking microphone. Information on the microphone type of the first channel will not be available to systems.

The microphone channel data will be provided in 16-bit linear-pcm form that differs from the telephone data provided.

4.1.3 Summed test segment condition

As mentioned in section 2.2.2, there will be test segments each consisting of an excerpt from a telephone conversation where the two sides of each conversation, in which both the target speaker and another speaker participate, are summed together. Thus the challenge is to be able to correctly detect the target speaker despite the presence of speech from another speaker.

4.2 Factors Affecting Performance

All trials will be *same-sex* trials. This means that the sex of the test segment speaker in the channel of interest (or of at least one test segment speaker for the summed test segment condition), will be the same as that of the target speaker model. Performance will be

[3] Note that P_{Known}, the known non-target prior used to compute system performance, is not the same as the prior probability of known non-target trials in the corpus.

[4] The reasons for choosing this cost function, and its possible interpretations, are described in detail in the paper "Application-independent evaluation of speaker detection" in Computer Speech & Language, volume 20, issues 2-3, April-July 2006, pages 230-275, by Niko Brummer and Johan du Preez.

reported separately for males and females and also for both sexes pooled.

This evaluation will include an examination of the effects of channel on recognition performance. This will include in particular the comparison of performance involving telephone segments with that involving microphone segments.

For trials involving microphone test segments, it will be of interest to examine the effect of the different microphone types tested on performance, and the significance on performance of the presence of the test microphone in the training data.

All or most trials involving telephone test segments will be *different-number* trials. This means that the telephone numbers, and presumably the telephone handsets, used in the training and the test data segments will be different from each other. If some trials are same-number, primary interest will be on results for different-number trials, which may be contrasted with results on same-number trials.

Some of the test segments will include additive noise (noise added as a post-processing step after recording) or will be recorded in an intentionally noisy environment or both. The impact of noise on performance will be examined in this evaluation.

The Core test will include relatively large amounts of training data distributed in advance of the evaluation period as well as limited training data distributed at the start of the evaluation period. NIST will compare performance of speakers in these training conditions.

Past NIST evaluations have shown that the type of telephone handset and the type of telephone transmission channel used can have a great effect on speaker recognition performance. Factors of these types will be examined in this evaluation to the extent that information of this type is available.

Telephone callers are generally asked to classify the transmission channel as one of the following types:

- Cellular
- Cordless
- Regular (i.e., land-line)

Telephone callers are generally also asked to classify the instrument used as one of the following types:

- Speaker-phone
- Head-mounted
- Ear-bud
- Regular (i.e., hand-held)

4.3 Common Evaluation Condition

In each evaluation NIST has specified one or more common evaluation conditions, subsets of trials in the core test that satisfy additional constraints, in order to better foster technical interactions and technology comparisons among sites. The performance results on these trial subsets are treated as the basic official evaluation outcomes. Because of the multiple types of test conditions in the 2012 core test, and the likely disparity in the numbers of trials of different types, it is not appropriate to simply pool all trials as a primary indicator of overall performance. Rather, the common conditions to be considered in 2012 as primary performance indicators will include the following subsets of all of the core test trials:

1. All trials involving multiple segment training and interview speech in test without added noise in test
2. All trials involving multiple segment training and phone call speech in test without added noise in test
3. All trials involving multiple segment training and interview speech with added noise in test
4. All trials involving multiple segment training and phone call speech with added noise in test
5. All trials involving multiple segment training and phone call speech intentionally collected in a noisy environment in test

4.4 Comparison with Previous Evaluations

In each evaluation it is of interest to compare performance results, particularly of the best performing systems, with those of previous evaluations. This is generally complicated by the fact that the evaluation conditions change in each successive evaluation. This is particularly problematic for SRE12, given the change in task conditions as discussed in section 1. For the 2012 evaluation the training condition released at evaluation time and consisting of a single segment will be similar to the task condition in 2010. Thus it will be possible to make relatively direct comparisons between 2012 and 2010 in this limited circumstance.

To help address the desire to make comparison with previous efforts, sites participating in the 2012 evaluation that also participated in 2010 are encouraged to submit to NIST results for their (unmodified) 2010 (or earlier year) systems run on the 2012 data for the same test conditions as previously. Such results will not count against the limit of three submissions per test condition (see section 7). Sites are also encouraged to "mothball" their 2012 systems for use in similar comparisons in future evaluations.

5 DEVELOPMENT DATA

All of the previous NIST SRE evaluation data, covering evaluation years 1996-2010, may be used as development data for 2012. This includes the additional interview speech used in the follow-up evaluation to the main 2008 evaluation. All of this data, or just the data not already received, will be sent to prospective evaluation participants by the Linguistic Data Consortium on one or more hard drives or DVD's, provided the required license agreement is signed and submitted to the LDC.[5] This development data includes the SRE12 training data for most of the target speakers (training data for some target speakers will be released at the beginning of the evaluation period along with the test data).

Participating sites may use other speech corpora to which they have access for development. Such corpora must be described in the site's system description (section 10).

6 EVALUATION DATA

The test data for this evaluation (other than that for the HASR test, described in section 11) will be distributed to evaluation participants by NIST on a USB hard drive. The LDC license agreement described in section 5, which all sites must sign to participate in the evaluation, will govern the use of this data for the evaluation.

---

[5] http://nist.gov/itl/iad/mig/upload/2012_NIST_SRE_Data_Agreement-v3.pdf

Since both channels of all telephone conversational data are provided, this data will not be processed through echo canceling software. Participants may choose to do such processing on their own.[6]

All telephone channel test data will be encoded as 8-bit μ-law speech samples and all microphone channel data will be encoded as 16-bit linear pcm. All test data will be stored in separate SPHERE[7] files. In addition to the information that is contained in a standard SPHERE header, evaluation data will include in the header entries for channel (mic or tel) and speaking style (interview or phonecall). The SPHERE header will not contain information on the type of telephone transmission channel or the type of telephone instrument or microphone involved.

6.1 Numbers of Test Segments

Table 3 provides upper bounds on the numbers of segments[8] to be included in the evaluation for each test condition.

Table 3 Upper bounds on the number of test segments

Test Data	Max Segments
Core/Extended	100,000
Summed	100,000

6.2 Numbers of Trials

Table 4 gives upper bounds on the numbers of trials to be included in the evaluation for each test condition.

The trials for each of the speaker detection tests will be specified in separate index files. These will be text files in which each record specifies the target speaker id, the test segment, and the side for a particular trial.

Table 4 Upper bounds on the number of trials

Test Conditions	Max Trials
Core	1,000,000
Extended (optional)	100,000,000
Summed (optional)	1,000,000

7 Evaluation Rules[9]

In order to participate in the 2012 speaker recognition evaluation a site must submit complete results for the required test condition as specified in section 2.2. A test submission is complete if and only if it includes a score for every trial in the test.

All participants must observe the following evaluation rules and restrictions in their processing of the evaluation data (modified rules for the HASR test are specified in section 11.2).

- Each score is to be based only upon the training data and the specified test segment. Information about other test segments (including for example normalization of scores over multiple test segments) is **not** allowed.[10]
- The use of manually produced transcripts or other human-created information is **not** allowed.
- Knowledge of the sex of the *target* speaker **is** allowed. Note that no cross-sex trials are planned, but that summed-channel segments may include speech from an opposite sex speaker.
- Listening to the evaluation test data, or any other human interaction with the test data, is **not** allowed. It should be noted, however, that human interaction with the evaluation **training data** is permitted.
- Knowledge of any information available in the SPHERE header **is** allowed.
- The following general rules about evaluation participation procedures will also apply for all participating sites:
  - Access to past presentations – Each new participant that has signed up for, and thus committed itself to take part in, the upcoming evaluation and workshop will be able to receive, upon request, the CD of presentations that were presented at the preceding workshop.
  - Limitation on submissions – Each participating site may submit results for up to three different systems per evaluation condition for official scoring by NIST. Results for earlier year systems run on 2012 data will not count against this limit. Note that the answer keys will be distributed to sites by NIST shortly after the submission deadline. Thus each site may score for itself as many additional systems and/or parameter settings as desired.
  - Attendance at workshop – Each evaluation participant is required to have one or more representatives at the evaluation workshop who must present there a meaningful description of its system(s). Evaluation participants failing to do so will be excluded from future evaluation participation.
  - Dissemination of results
    - Participants may publish or otherwise disseminate their own results.
    - NIST will generate and place on its web site charts of all system results for conditions of interest, but these charts will not contain the site names of the systems involved. Participants may publish or otherwise disseminate these charts, unaltered and with appropriate reference to their source.
    - Participants may not publish or otherwise disseminate their own comparisons of their performance results with

[6] One publicly available source of such software is http://www.ece.msstate.edu/research/isip/projects/speech/software/legacy/fir_echo_canceller/

[7] ftp://jaguar.ncsl.nist.gov/pub/sphere_2.6a.tar.Z

[8] A segment is a single unique audio file and includes both sides of the conversation, either as two separate channels or a single summed channel.

[9] Rules for the HASR evaluation are specified in section 11.

[10] This means that the technology is viewed as being "application-ready". Thus a system must be able to perform speaker detection simply by being trained on the training data for a specific target speaker and then performing the detection task on whatever speech segment is presented, without the (artificial) knowledge of other test data.

those of other participants without the explicit written permission of each such participant. Furthermore, publicly claiming to "win" the evaluation is **strictly prohibited**. Participants violating this rule will be excluded from future evaluations.

8 EVALUATION DATA SET ORGANIZATION

This section describes the organization of the evaluation data other than the HASR data, which will be provided separately to those doing the HASR test.

The organization of the evaluation data will be:

- A top level directory used as a unique label for the disk: "**sp12-NN**" where NN is a digit pair identifying the disk
- Under which there will be three sub-directories: "**data**", "**test**", and "**doc**"

8.1 data Sub-directory

The "**data**" directory will contain all of the speech test segments as well as any training segments not previously released. Its organization will not be explicitly described. Rather the files in it will be referenced in other sub-directories.

8.2 train Sub-directory

The "**train**" directory will contain a table of all target speakers that provides links to their speech files located either in the data directory or in the training data distributed by the LDC. This table is a superset of the information that was also provided to evaluation participants at the time of registration.

8.3 trials specification

There will be three index files, named **core**.ndx, **summed**.ndx, and **extended**.ndx, to be used for the identically named test conditions. (The extended.ndx file will also be used for the **known** and **unknown** test conditions.)

Each record in the index files will correspond to one trial and will contain three comma separated fields:

1. The first field is a target speaker identification string.
2. The second is the file name of a test segment within the data directory.
3. The third is the channel designator (either "A" or "B").

These index files will be distributed to evaluation participants via FTP.

8.4 doc Sub-directory

This will contain text files that document the evaluation and the organization of the evaluation data. This evaluation plan document will be included.

9 SUBMISSION OF RESULTS

This section does not apply to the HASR test, whose submission requirements are described separately (section 11.4).

Results for each test must be provided to NIST in a single separate file using standard ASCII format, with one record for each trial.

Each file record must document its trial output with 4 comma separated fields:

1. The target speaker identification string
2. The test segment file name
3. The channel designator
4. The score. In SRE12 the score is required to represent the system's estimate of the log likelihood ratio (i.e., the natural logarithm of the target/non-target likelihood ratio).

Submissions must be made via ftp. The appropriate addresses for submissions will be supplied to participants receiving evaluation data.

New to SRE12, NIST will be releasing software that verifies a submission's validity. More information on the submission checker software will be made available to participants prior to the start of the evaluation.

10 SYSTEM DESCRIPTION

A brief description of the system(s) (the algorithms) used to produce the results must be submitted along with the results, for each system evaluated. This should include a description of any human interaction with the evaluation training data.

A single site may submit the results for up to three separate systems for evaluation for each particular test, not counting results for earlier year systems run on the 2012 data. Please note that a "primary" system for each test completed must be identified as part of the submission. Sites are welcome to present descriptions of and performance results for additional systems beyond those submitted at the evaluation workshop.

For each system for which results are submitted, sites must report the CPU execution time that was required to process the evaluation data, as if the test were run on a single CPU. This should be reported separately for creating models from the training data and for processing the test segments, and should be reported as a multiple of real-time for the data processed. This may be reported separately for each test. Sites must also describe the CPU(s) utilized and the amounts of memory used.

11 HASR TEST

The Human Assisted Speaker Recognition (HASR) test will contain a subset of the core test trials of SRE12 to be performed by systems involving, in part or in whole, human judgment to make trial decisions. The systems doing this test may include large amounts of automatic processing, with human involvement in certain key aspects, or may be solely based on human listening. The humans involved in a system's decisions may be a single person or a panel or team of people. These people may be professionals or experts in any type of speech or audio processing, or they may be simply "naïve" listeners. The required system descriptions (section 11.2) must include a description of the system's human element.

Forensic applications are among the applications that the HASR test serves to inform, but the HASR test should not be considered to be a true or representative "forensic" test. This is because many of the factors that influence speaker recognition performance and that are at play in forensic applications are controlled in the HASR test data, which are collected by the LDC following their collection protocols.

11.1 Trials and Data

To accommodate different interests and levels of effort, two test sets will be offered, one with 20 trials (HASR1), and one with 200 trials (HASR2). HASR participants may choose to perform either test.

Because of the small numbers of trials in the HASR test set, the difficulty of the test will be increased by selection of difficult trials.

Objective criteria will be used to select dissimilar test conditions for target trials and similar speakers for non-target trials.

Data used in the 2010 HASR pilot evaluation will be made available upon request to any site participating in the 2012 HASR evaluation.

11.2 Rules

The rules on data interaction as specified in section 7 not allowing human listening or transcript generation or other interaction with the data, do not apply, but the requirement for processing each trial separately and making decisions independently for each trial remains in effect. Specifically:

- Use of information about other trials is **not** allowed.

This presents a dilemma for human interactions, however, because humans inherently carry forward information from prior experience. To help minimize the impact of this prior exposure on human judgments, the trials will be released sequentially via an online automatic procedure. The protocol for this sequential testing will be specified in greater detail in mid-2012, but will basically work as follows:

- NIST will release the first trial as a three-field record as specified in section 8.3 for the core index file.
- The participant will process that trial and submit the result to NIST in the format specified in section 11.4.
- NIST will verify the submission format, and then make the next trial available for download to the participant.

The training and test speech data for each trial may be listened to by the human(s) involved in the processing as many times and in any order as may be desired. The human processing time involved must be reported in the system descriptions (see section 11.4 below).

The rules on dissemination of results as specified in section 7 will apply to HASR participants,

System descriptions are required as specified in section 10. They may be sent to NIST at any time during the processing of the HASR trials, or shortly after the final trial is processed. They should also describe the human(s) involved in the processing, how human expertise was applied, what automatic processing algorithms (if any) were included, and how human and automatic processing were merged to reach decisions. Execution time should be reported separately for human effort and for machine processing (if relevant).

Because HASR remains a pilot evaluation with an unknown level of participation, participating sites will not in general be expected to be represented at the SRE12 workshop. NIST will review the submissions, and most particularly the system descriptions, and will then invite representatives from those systems that appear to be of particular interest to the speaker recognition research community to attend the workshop and offer a presentation on their system and results. One workshop session will be devoted to the HASR test and to comparison with automatic system results on the HASR trials.

HASR is open to all individuals and organizations who wish to participate in accordance with these rules.

11.3 Scoring

Scoring for HASR will be very simple. Trial decisions ("**same**" if the segment speaker is judged to be the target speaker, otherwise "**different**") will be required. In light of the limited numbers of trials involved in HASR, we will simply report for each system the overall number of correct detections ($N_{correct}$ detections for N_{target} trials) and the overall number of correct rejections ($N_{correct}$ rejections on $N_{non\text{-}target}$ trials).

Scores for each trial will be required as in the automatic system evaluation, with higher scores indicating greater confidence that the test speaker is the target speaker. It is recognized, however, that when human judgments are involved there may only be a discrete and limited set of possible score values. In the extreme, there might only be two; e.g., 1.0 corresponding to "same" decisions and -1.0 corresponding to "different" decisions. This is acceptable. The scores will be used to produce *Detection Error Tradeoff (DET)* curves[11], or a discrete set of DET points, and compared with the performance of automatic systems on the same trial set.

For each submission, the system description (section 11.2) should specify how scores were determined. Where this is a discrete set, the meaning of each possible score should be explained. It should also be indicated whether the scores may be interpreted as log likelihood ratios. [12]

11.4 Submissions

HASR trial submissions should use the following record format:

1. The test condition – "HASR1" or "HASR2"
2. The trial index number (1 through 20 for HASR1, 1 through 200 for HASR2)
3. The decision as specified above in section 11.3
4. The score as specified above in section 11.3

12 SCHEDULE

The deadline for signing up to participate in the evaluation is August 1, 2012.

The HASR data set will become available for sequential distribution of trial data to registered participants in this test beginning on August 1, 2012

The evaluation data (other than the HASR data) set will be distributed by NIST so as to arrive at participating sites on September 24, 2012.

The deadline for submission of evaluation results (including all HASR trial results) to NIST is October 15, 2012 at 11:59 PM, Washington, DC time (EDT or GMT-5).

Initial evaluation results will be released to each site by NIST on November 5, 2012.

The deadline for site workshop presentations to be supplied to NIST in electronic form for inclusion in the workshop proceedings is December 3rd, 2012.

---

[11] For details regarding DET curves, see: http://www.itl.nist.gov/iad/mig/publications/storage_paper/det.pdf

[12] A possible description of multiple scoring classes, and how they might be viewed as corresponding to log likelihood ratios, is offered in "Forensic Speaker Identification", Taylor & Francis, 2002, by Philip Rose, on page 62.

The deadline for registration and room reservations for the workshop is to be determined.

The follow-up workshop will be held December 11th-December 12th, 2012 in Orlando, Florida, USA. All sites participating in the main evaluation must have one or more representatives in attendance to discuss their systems and results.

13 GLOSSARY

Test – A collection of trials constituting an evaluation component.

Trial – The individual evaluation unit involving a test segment and a hypothesized speaker.

Target speaker – The hypothesized speaker of a test segment, one for whom a model has been created from training data.

Non-target speaker – A hypothesized speaker of a test segment who is in fact not the actual speaker.

Segment speaker – The actual speaker in a test segment.

Target trial – A trial in which the actual speaker of the test segment *is in fact* the target (hypothesized) speaker of the test segment.

Non-target trial – A trial in which the actual speaker of the test segment *is in fact not* the target (hypothesized) speaker of the test segment.

Turn – The interval in a conversation during which one participant speaks while the other remains silent.

NIST 2016 Speaker Recognition Evaluation Plan

August 4, 2016

1 Introduction

The 2016 speaker recognition evaluation (SRE16) is the next in an ongoing series of speaker recognition evaluations conducted by NIST since 1996. These evaluations serve (1) to support speaker recognition research by exploring promising new ideas in speaker recognition and developing advanced technology incorporating these ideas and (2) to measure and calibrate the performance of speaker recognition systems. They are intended to be of interest to all researchers working on the general problem of text independent speaker recognition. To this end the evaluation is designed to be simple, to focus on core technology issues, to be fully supported, and to be accessible to those wishing to participate.

The basic task in NIST's speaker recognition evaluations is speaker detection, i.e., to determine whether a specified target speaker is speaking during a given segment of speech. Like previous SREs, SRE16 focuses on telephone speech recorded over a variety of handset types. However, there are several differences with previous SREs:

- Target speaker data will not be distributed in advance like in SRE12
- Fixed training condition is introduced to allow better cross-system comparisons
- Test segments will have more duration variability than in previous evaluations
- The enrollment and test data were collected outside North America
- The evaluation will be conducted using same and different phone number trials

Participation in the evaluation is open to all who find the evaluation of interest and are able to comply with the evaluation rules set forth in this plan. There is no cost to participate, but participants must be represented at the evaluation workshop to be held in San Juan, Puerto Rico on December 11-12, 2016. Information about evaluation registration can be found on the SRE16 website[1].

2 Task Description

2.1 Task Definition

As stated in the Introduction, the task for SRE16 is *speaker detection*: given a segment of speech and the target speaker enrollment data, automatically determine whether the target speaker is speaking in the segment. A segment of speech (test segment) along with the enrollment speech segments(s) from a designated target speaker constitutes a *trial*. The system is required to process each trial independently and to output a log likelihood ratio (LLR) for that trial. The LLR is defined as follows:

$$LLR = ln\left(\frac{pdf(data|TargetHyp.)}{pdf(data|NonTargetHyp.)}\right) \quad (1)$$

[1] http://www.nist.gov/itl/iad/mig/sre16.cfm

2.2 Training Conditions

The training condition is defined as the amount of data/resources used to build a Speaker Recognition (SR) system. The task described above can be evaluated over a *fixed* (required) or *open* (optional) training condition.

- **Fixed** – The fixed training condition limits the system training to specific data sets. They are:
  - data provided from the new corpus collection
  - previous SRE data
  - Switchboard corpora that contain transcripts
  - Fisher corpora

  The LDC license lists the actual catalog numbers of these corpora. Participants can obtain the data from the Linguistic Data Consortium (LDC) after they have signed the LDC data license agreement. For the fixed training condition, only the specified speech data may be used for system training and development, to include all sub-systems (e.g., speech activity detection) and auxiliary systems used for automatic labels/processing (e.g., language recognition). Publicly available, non-speech audio and data (e.g., noise samples, impulse responses, filters) may be used and should be noted in the system description. Participation in this condition is required.
- **Open** – The open training condition removes the limitations of the fixed condition. In addition to the data listed in the fixed condition, participants can use other publicly available data. LDC will make selected data from the IARPA Babel Program to be used in the open training condition. Participation in this condition is optional but encouraged.

Sites are strongly encouraged to participate in both the fixed and open conditions to demonstrate the gains that can be achieved with unconstrained amounts of data.

2.3 Enrollment Conditions

The enrollment condition is defined as the number of speech segments provided to create a target speaker model. However, unlike previous SREs, gender labels will not be provided. There are two enrollment conditions for SRE16:

- **One-segment** – the system is given only one approximately 60 secs[2] of segment to build the model of the target speaker.
- **Three-segment** – the system is given three approximately 60 secs segments to build the model of the target speaker, all from the same phone number.

2.4 Test Conditions

- The test segments will be uniformly sampled ranging approximately from 10 secs to 60 secs. The test segments that are less than 9 secs will not be included in the primary metric calculation but will be scored for analysis of systems' behavior.
- Trials will be conducted with test segments from both same and different phone numbers as the enrollment segment(s).
- There will be no cross-sex trials.
- There will be no cross-language trials.

[2] as determined by SAD output

3 Performance Measurement

3.1 Primary Metric

A basic cost model is used to measure the speaker detection performance and is defined as a weighted sum of miss and false alarm error probabilities:

$$\begin{aligned} C_{Det}(C_{Miss}, C_{FalseAlarm}, P_{Target}) = & C_{Miss} \times P_{Target} \times P_{Miss|Target} + \\ & C_{FalseAlarm} \times (1 - P_{Target}) \times P_{FalseAlarm|NonTarget} \end{aligned} \tag{2}$$

where the parameters of the cost function are C_{Miss} (cost of a missed detection) and $C_{FalseAlarm}$ (cost of a spurious detection), and P_{Target} (a priori probability of the specified target speaker) and are defined to have the following values:

Parameter ID	C_{Miss}	$C_{FalseAlarm}$	P_{Target}
1	1	1	0.01
2	1	1	0.005

Table 1: SRE16 cost parameters

To improve the intuitive meaning of C_{Det}, it will be normalized by dividing it by $C_{Default}$, defined as the best cost that could be obtained without processing the input data (i.e., by either always accepting or always rejecting the segment speaker as matching the target speaker, whichever gives the lower cost):

$$C_{Norm} = \frac{C_{Det}}{C_{Default}} \tag{3}$$

where $C_{Default}$ is defined as:

$$C_{Default} = min \begin{cases} C_{Miss} \times P_{Target}, \\ C_{FalseAlarm} \times (1 - P_{Target}) \end{cases} \tag{4}$$

Substituting either set of parameter values from Table 1 into Equation 4 yields:

$$C_{Default} = C_{Miss} \times P_{Target} \tag{5}$$

Substituting C_{Det} and $C_{Default}$ in Equation 3 with Equations 2 and 5, respectively, along with some algebraic manipulations yields:

$$C_{Norm} = P_{Miss|Target} + \beta \times P_{FalseAlarm|NonTarget} \tag{6}$$

where β is defined as:

$$\beta = \frac{C_{FalseAlarm}}{C_{Miss}} \times \frac{1 - P_{Target}}{P_{Target}} \tag{7}$$

Actual detection costs will be computed from the trial scores by applying detection thresholds of $\log(\beta)$ for the two values of β, with β_1 for $P_{Target_1} = 0.01$ and β_2 for $P_{Target_2} = 0.005$. Thus, the primary cost measure for SRE16 is defined as:

$$C_{Primary} = \frac{C_{Norm_{\beta_1}} + C_{Norm_{\beta 2}}}{2} \tag{8}$$

The evaluation data will be divided into 16 partitions. Each partition is defined as a combination of

enrollment (1-segment or 3-segment), language (Tagalog or Cantonese), sex (Male or Female), and phone number match (same or different). $C_{Primary}$ will be calculated for each partition, and the final results is the average of all the partitions' $C_{Primary}$'s.

Also, a minimum detection cost will be computed by using the detection thresholds that minimize the detection cost. Note that for minimum cost calculations, the counts for each condition set will be equalized before pooling and cost calculation (i.e., minimum cost will be computed using a single threshold not one per condition set).

NIST will make available the script that calculates the primary metric.

3.2 Alternative Metric

In addition to the primary metric, an alternative, information theoretic measure may be computed that considers how well all scores represent the likelihood ratio and that penalizes for errors in score calibration. This performance measure is defined as:

$$C_{llr} = \frac{1}{2 \times \log(2)} \times \left(\frac{\sum \log(1 + \frac{1}{s})}{N_{TT}} + \frac{\sum \log(1 + s)}{N_{NT}} \right) \tag{9}$$

where the first summation is over all target trials N_{TT}, the second is over all non-target trials N_{NT}, and s represents a trial's likelihood ratio[3].

4 Data Description

The data collected by the LDC as part of the Call My Net Speech Collection to support speaker recognition research will be used to compile the SRE16 test set, development set, and part of the training set[4].

The data are composed of telephone conversations collected outside North America, spoken in Tagalog and Cantonese (referred to as the *major* language) and Cebuano and Mandarin (referred to as the *minor* languages). The development set described below will contain data from both the major and minor languages, while the test set will be contain data from the two major languages. Recruited speakers (called *claque* speakers) made multiple calls to people in their social network (e.g., family, friends). Claque speakers were encouraged to use different telephone instruments (e.g., cell phone, landline) in a variety of settings (e.g., noisy cafe, quiet office) for their initiated calls and were instructed to talk for 10 minutes on a topic of their choosing.

All segments will be encoded as a-law sampled at 8kHz in SPHERE formatted files. The development and test sets will be distributed by NIST via Amazon Web Services (AWS).

4.1 Data Organization

The development and test sets follow a similar directory structure:

<base_directory>/

- README.txt
- data/
  - enrollment/
  - test/
  - unlabeled/ (in training set only)
- docs/
- metadata/ (in development set only)

[3]The reasons for choosing this cost function, and its possible interpretations, are described in detail in the paper "Application-independent evaluation of speaker detection" in Computer Speech & Language, volume 20, issues 2-3, April-July 2006, pages 230-275, by Niko Brummer and Johan du Preez.

[4]The entire training set also includes previous SRE corpora, Switchboard, and Fisher corpora. See Section 4.4.

4.2 Trial File

The trial file, named `sre16_{dev|eval}_trials.tsv` and located in the `docs` directory, is composed of a header and a set of records where each record describes a given trial. Each record is a single line containing three fields separated by a tab character and in the following format:

`modelid<TAB>segment<TAB>side<NEWLINE>`

where
`modelid` - The enrollment identifier
`segment` - The test segment identifier
`side` - The channel[5]

For example:

```
modelid segment side
1001_sre16 dtadhlw_sre16 a
1001_sre16 dtaekaz_sre16 a
1001_sre16 dtaekbb_sre16 a
```

4.3 Development Set

Participants in the SRE16 evaluation will receive data for development experiments that will mirror the evaluation conditions. The development data will be drawn from the minor languages and will include:

- 20 speakers, 10 each from Cebuano and Mandarin
- 10 calls per speaker
- Associated metadata which will be listed in the following files located in the `metadata` directory as outlined in section 4.1.
  - calls.tsv - information about the calls (e.g., conversations)
  - call_sides.tsv - information about the call sides
  - languages.tsv - information about the languages
  - subjects.tsv - information about the speakers

The development data may be used for any purpose.

4.4 Training Set

Section 2.2 describes the two training conditions: Fixed (required) and Open (optional). Participants in the SRE16 evaluation will receive a common set of data resources for training for the fixed training condition. An unlabeled (i.e., no speaker id, gender, language, or phone number information) set of approximately 2200 calls from the Call My Net collection will be made available divided into sets from the minor and major languages. In addition participants will receive data from all previous SRE corpora as well as Switchboard corpora that contain transcripts and the Fisher corpus with transcripts. To obtain this set, participants must sign the LDC data license agreement which outlines the terms of the data usage.

Additionally, LDC will be releasing selected data resources from the IARPA Babel Program for use in the open training condition. All training sets will be available directly from the LDC[6].

Participants are encouraged to submit results for the contrastive open training condition to demonstrate the value of additional data.

[5] SRE16 segments will be single channel so this field is always "a"
[6] http://www.ldc.upenn.edu

5 Evaluation Rules and Requirements

SRE16 is conducted as an open evaluation where the test data is sent to the participants who process the data locally and submit the output of their systems to NIST for scoring. As such, the participants have agreed to process the data in accordance with the following rules:

- The participants agree to abide by the terms guiding the training conditions (fixed or open).
- The participants agree to process at least the fixed training condition.
- The participants agree to process each trial independently. That is, each decision for a trial is to be based only upon the specified test segment and target speaker enrollment data. The use of information about other test segments and/or other target speaker data is not allowed.
- The participants agree not to probe the enrollment or test segments via manual/human means such as listening to the data or producing the transcript of the speech.
- The participants agree not to produce manual/human annotations of the unlabeled training data, such as employing a service like Amazon's Mechanical Turk. Informal listening and spectral analysis of subsets of the audio are acceptable.
- The participants are allowed to use any automatically derived information for training, development, enrollment, test segments, provided that the automatic system used conforms to the training data condition (fixed or open) for which it is used.
- The participants are allowed to use information available in the SPHERE header.
- The participants can submit up to three systems per training condition. Bug-fix does not count toward this limit.

In addition to the above data processing rules, participants agree to comply with the following general requirements:

- The participants agree to have one or more representatives at the evaluation workshop to present a meaningful description of their system(s). Evaluation participants failing to do so will be excluded from future evaluation participation.
- The participants agree to the guidelines governing the publication of the results:
  - Participants are free to publish results for their own system but must not publicly compare their results with other participants (ranking, score differences, etc.) without explicit written consent from the other participants.
  - While participants may report their own results, participants may not make advertising claims about winning the evaluation or claim NIST endorsement of their system(s). The following language in the U.S. Code of Federal Regulations (15 C.F.R. § 200.113) shall be respected[7]: *NIST does not approve, recommend, or endorse any proprietary product or proprietary material. No reference shall be made to NIST, or to reports or results furnished by NIST in any advertising or sales promotion which would indicate or imply that NIST approves, recommends, or endorses any proprietary product or proprietary material, or which has as its purpose an intent to cause directly or indirectly the advertised product to be used or purchased because of NIST test reports or results.*
  - At the conclusion of the evaluation NIST generates a report summarizing the system results for conditions of interest, but these results/charts do not contain the participant names of the systems involved. Participants may publish or otherwise disseminate these charts, unaltered and with appropriate reference to their source.

[7] See http://www.ecfr.gov/cgi-bin/ECFR?page=browse

– The report that NIST creates should not be construed or represented as endorsements for any participant's system or commercial product, or as official findings on the part of NIST or the U.S. Government.

6 Evaluation Protocol

To facilitate information exchange between the participants and NIST, all evaluation activities are conducted over a web-interface.

6.1 Evaluation Account

Participants must sign up for an evaluation account where they can perform various activities such as registering for the evaluation, signing the data license agreement, uploading the submission and system description. To sign up for an evaluation account, go to https://sre.nist.gov. The password must be at least 12 characters long and must contain a mix of upper and lowercase letters, numbers, and symbols. After the evaluation account is confirmed, the participant is asked to join a site or create one if it does not exist. The participant is also asked to associate his site to a team or create one if it does not exist. This allows multiple members with their individual accounts to perform activities on behalf of their site and/or team (e.g., make a submission) in addition to performing their own activities (e.g., requesting workshop invitation letter).

- A site is defined as a single organization (e.g., NIST)
- A team is defined as a group of organizations collaborating on a task (e.g., Team1 consisting of NIST and LDC)
- A participant is defined as a member or representative of a site who takes part in the evaluation (e.g., John Doe)

6.2 Evaluation Registration

One participant from a site must formally register his site to participate in the evaluation by agreeing to the terms of participation. For more information about the terms of participation, see Section 5.

6.3 Data License Agreement

One participant from each site must sign the LDC data license agreement to obtain the training data for the fixed training condition and Babel data for the open training condition.

6.4 Submission Requirements

Each team must participate in the fixed training condition. Teams are encouraged to participate in the open training condition to demonstrate the gains that can be achieved with unconstrained amounts of data. For each training condition, the team can submit up to three systems and must designate one as the *primary* system that NIST uses for cross-team comparisons. There should be one output file for each training condition per system.

Each team is required to submit a system description at the designated time (see Section 7). The evaluation results are given only after the system description is received.

6.4.1 System Output Format

The system output file is composed of a header and a set of records where each record contains a trial given in the trial file (see Section 4.2) and a log likelihood ratio output by the system for the trial. The order of the trials in the system output file must follow the same order as the trial list. Each record is a single line containing 4 fields separated by tab character in the following format:

`modelid<TAB>segment<TAB>side<TAB>llr<NEWLINE>`

where
`modelid` - The enrollment identifier
`segment` - The test segment identifier
`side` - The channel (always "a" for SRE16 since the data is single channel)
`llr` - The log likelihood ratio

For example:

```
modelid segment side llr
1001_sre16 dtadhlw_sre16 a 0.79402
1001_sre16 dtaekaz_sre16 a 0.24256
1001_sre16 dtaekbb_sre16 a 0.01038
```

There should be one output file for each training condition for each system. NIST will make available the script that validates the system output.

6.4.2 System Description Format

Each team is required to submit a system description. The system description must include a brief description of the systems/algorithms used to produce the results and a timing report. The timing report describes the CPU execution time that is required to process the test set as if running on a single CPU and as a multiple of real-time for the data processed. The timing report should identify the time for creating models from the enrollment data and the time needed for processing the test segments. The timing report should include the CPU(s) utilized and the amounts of memory used. The system description should follow the IEEE conference proceeding template. A copy of the template is available on the SRE16 website.

7 Schedule

Milestone	Date
Evaluation plan published	March 2016
Registration period	April 19 - September 13, 2016
Training data available	May, 2016
Evaluation data available to participants	September 20, 2016
System output due to NIST	October 11, 2016
Preliminary results released	October 25, 2016
Post evaluation workshop co-located with SLT in San Juan, Puerto Rico	December 11-12, 2016