IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

Recent Developments in Speaker Verification at IDIAP

Bojan Nedic [a]   Hervé Bourlard [a]

IDIAP–RR 00-26

September 2000

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] IDIAP, Martigny, Switzerland

IDIAP Research Report 00-26

Recent Developments in Speaker Verification at IDIAP

Bojan Nedic Hervé Bourlard

September 2000

Abstract. This report presents recent developments in speaker verification at IDIAP. The first part is mostly related to text-independent speaker verification with the special emphasis on NIST'99 competition in which IDIAP took place with the system based on Gaussian Mixture Models (GMM) that proved to yield state-of-the-art performance. The second part of the report is devoted to text-dependent speaker verification, more precisely speaker verification based on user-customized password. A very desirable feature in real-world applications is that the user can choose his/her password instead of being prompted by the system as in text-prompted speaker verification (customization of the password). In the case of text-prompted speaker verification system knows in advance the (prompted) text, and thus the hidden Markov model (HMM) associated with the training and test utterances whereas in user-customized password, the password and its associated model is a priori unknown. This essentially imposes two issues that are discussed in this report, namely *HMM inference*, in which the system has to automatically infer the best HMM model from a few repetitions of the password and *HMM adaptation*, where the parameters of the obtained model will have to be adapted to the characteristics of the speaker's voice. In our case, these issues are addressed in the framework of hybrid HMM/ANN systems, using artificial neural Networks (ANN) to estimate local HMM posterior probabilities.

Contents

1 Introduction

Speaker recognition is the process of automatically recognizing/verifying who is speaking on the basis of the individual characteristics arising in the speaker's voice signal. Many of these speaker specific characteristics are independent of the linguistic content of the utterance and are in general considered as a source of degradation in speech recognition. As opposed to speech recognition, it is thus important to use acoustic features that preserve the speaker's voice characteristics.

Speaker recognition generically refer to two different problems: speaker identification and speaker verification. Speaker identification is the process of determining which of the registered speakers a given utterance comes from. In other words, its goal is to determine which one of the known voices best matches the input voice sample. In this sense, the speaker identification process is similar to the spoken word recognition process since they are both related to a classification problem and have to determine which reference template is most similar to the input speech. On the other hand, speaker verification is essentially a problem of hypothesis testing and has to validate or reject a claimed identity. Indeed, the goal is then to determine from a voice sample if a person is well who he/she claims to be. In speaker identification, the number of decision alternatives is thus equal to the size of the population (possibly plus one, in the case of an "open set" problem, i.e., when the test speaker does not necessarily belong to the set of registered speakers). In the case of speaker verification, there are only two alternatives (accept or reject), regardless of the population size [1].

The common structure of speaker recognition systems is shown on **Figure 1**. In the "Feature extraction" block, the acoustic parameters (which should ideally be specific to the speaker's voice characteristics) are extracted from the speech waveform. When the switch is in position (1), we are training the parameters of the speaker models (i.e., enrolling speakers) for each potential customer. The most popular statistical models are GMMs (Gaussian Mixture Models), Hidden Markov Models (HMMs), codebooks obtained from Vector Quantization (VQ), or Artificial Neural Networks (ANNs). When the switch is in position (2), we are recognizing (testing) test sequences by comparing their feature parameters extracted from the speech wave with the stored reference models of registered speakers. In the case of speaker identification, we simply select the speaker with the best similarity score. In the case of speaker verification, the decision is made according to an hypothesis test computing some kind of similarity measure and comparing it with a decision threshold.

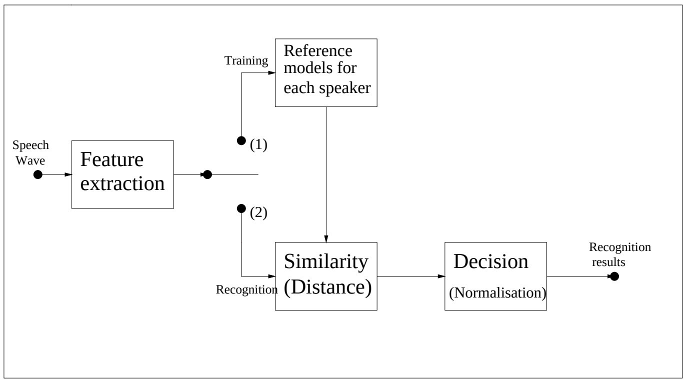


Figure 1: Principal structure of speaker recognition systems

Speaker verification (SV) technology has many potential applications. It makes possible to verify the identity of people accessing systems by using voice access control, which is considered as far more convenient than using PIN codes, cards, keys, or other artificial means. Among the numerous potential applications, we can mention the following:

- Banking transactions over the telephone network.

- Databases access services, including personal information access.
- Electronic commerce/telephone shopping.
- Security control for confidential information areas, e.g., in banking or voice mail applications.
- Remote access to computers.

Through such applications, speaker recognition technology is expected to create new services that make our daily lives more convenient. Finally, speaker verification and identification can also be used in military and forensic applications such as in criminal investigations to determine which of the suspects produced the voice recorded at the scene of the crime. Since the possibility that the real criminal is not one of the suspects always exists, the identification process should involve the combined process of speaker verification and speaker identification.

The present report focuses on the last results obtained in text independent SV, as well as on recent developments in user-customized password SV. After a quick overview of the different families of speaker verification systems, our current state-of-the-art text-independent speaker verification system (based on GMM's) will be described, and the results obtained on the last NIST'99 evaluation campaign (to which IDIAP participated) will be presented. Section 4 will then discuss the problem of text-dependent SV, and more particularly user-customized password SV. The proposed approach, based on hybrid HMM/ANN systems [2], will be presented and includes two main topics: HMM inference and ANN adaptation. Finally, the summary of what is to be done in this project in the near future will be given.

2 Text-Independent, Text-Dependent, Text-Prompted, and User-Customized Speaker Verification

Speaker verification can be based on text-dependent (fixed-text) or text-independent (free-text) methods. In the case of the text-dependent methods, it is required that the speaker uses the same sentence (or keyword) in both training and recognition, whereas in text-independent there is no such constraint. Consequently, text-dependent SV will also make use of the lexical information, on top of the speaker's voice characteristics, while text-independent SV will only use the latter. In general, because of the higher acoustic-phonetic variability of text-independent input, more training material will be required to reliably characterize (model) a speaker, comparatively to text-dependent approaches. The text-dependent methods are usually based on template matching techniques in which the time axis of an input speech sample is aligned with each reference template or reference model of registered speakers, and the computed similarity between them is accumulated from the beginning to the end of the utterance. The structure of text-dependent recognition methods is therefore rather simple. Since this method directly exploits the voice individuality associated with each phoneme or syllable, it generally achieves higher recognition performance than text-independent systems. There are some applications in which predetermined key words cannot be used. For example, utterances of the same words cannot always be compared in criminal investigations. It is quite often the case that speakers are uncooperative in criminal investigations in that they intentionally change their speaking rate and manner, making a text-independent method essential. Besides, human beings can recognize speakers irrespective of the content of the utterance.

Both text-dependent and text-independent verification systems have a serious problem. The speaker voice can be recorded and then reproduced. To cope with this problem, a solution, referred to as text-prompted speaker verification, consists in using a synthetic prompt to ask the user to repeat a different random sequence of key words, which are typically taken from a small lexicon (for example digits). In this case, speaker-independent speech recognition will usually be performed before speaker verification to make sure that the user actually repeated what he/she was prompted for. Therefore, pre-recorded utterances from a customer will be of no use to the impostor since the impostor can no longer predict in advance the prompted sequence. However, even this system can be defeated with the usage of advanced electronic equipment that can reproduce key words in requested order. Therefore a new solution has been proposed in which prompted random sequences are taken from very large vocabulary. What can happen in these methods is that even a registered speaker is rejected if the recognized text of the utterance differs from the prompted text. In that case the system will prompt the user with an additional (different) sentence etc..

In real-world services, it is often desirable that the user can choose his/her own password on which verification will be performed, which is then referred to as **user-customized password** SV. This enables better user-friendliness, which is considered as an essential functionality by some service providers for implementing speaker verification technology. This method also enables increased security as fraudulent access requires prior knowledge of the client's password and the user has the possibility to change his password whenever he wants to. If we compare this method with the text-prompted speaker verification, we can see that the main difference is that in the latter the system knows the text of the training utterances. In the case of user-customized password SV, the system first has to automatically infer the lexical characteristics of the customized password, simply on the basis of a few pronunciations of that password. As discussed later, another difficulty related to this approach will be the normalization of the score, usually using a "world model", i.e., a model of the way the rival speakers (impostors) would pronounce this password. So, the problem is also to infer speaker-independent model of the password from the single-speaker set of examples. There are several ways how to approach these problems, and some of them are investigated in the European PICASSO project (in which IDIAP is one of the partners). The approach discussed in the present report (Section 4) consists in using hybrid HMM/ANN systems.

3 Text-Independent Speaker Verification

3.1 Introduction

In the case of text-independent speaker verification, words or sentences cannot be predicted, and it is thus impossible to model all possible word sequences. Different approaches to this problem have been investigated so far [3]:

1. Long-Term-Statistics methods are based on simple statistics such as the long-term mean and variance that are calculated over a series of utterances. Recently, a new approach was investigated in which statistics of dynamic features are modeled by multi-dimensional autoregressive model [4].

2. VQ based methods in which spectral characteristics of each speaker are modeled by one or more codebooks that are obtained by clustering training data of each speaker. In recognition an input utterance is vector-quantized using the codebook of each reference speaker (or codebooks associated with the cohort, for normalization score) and distortion accumulated over the entire utterance is used for decision [5].

3. Fully connected (ergodic) HMM's that are used as speaker models. HMM's are usually trained according to a maximum-likelihood criterion and can have several (single or multi) Gaussian states, or just a single multi-Gaussian state. The latter case, which is a very degenerated case of HMM (with only one state!) is usually referred to as **Gaussian Mixture Model (GMM)** [6] and has proved to yield good performance in text-independent speaker verification. Today, GMM's seem to correspond to the state-of-the-art in text-independent speaker verification, and they will be discussed in more detail below.

4. Artificial neural network based methods in which each speaker has, after training, his own neural network. Neural nets have usually one or two outputs and are trained positively with the training utterances of the customer and negatively with the training utterances of the impostors [7].

5. Event-specific characteristics based methods in which specific events are extracted from the speech wave that are thought to have good discrimination properties. However, this method didn't have as good results as the other mentioned methods [8].

Below, we briefly discuss further the approach based on GMM's which, as already mentioned, seems to yield state-of-the-art performance, and has been used at IDIAP for several years, including for the last NIST'99 campaign.

3.2 The Gaussian Mixture Model

Today, GMM's usually demonstrate state-of-the-art performance in text-independent speaker verification (as well as in speaker identification), and has shown to be a robust statistically based representation of speaker identity. In this case, the distribution of feature vectors extracted from a speaker-specific speech signal is simply modeled by a Gaussian mixture density. A Gaussian mixture density is a weighted linear combination of M unimodal Gaussian densities, which can also be viewed as a single state HMM with a Gaussian mixture observation density. For a feature vector $\vec{x}$, and a claimed speaker class c (represented by a set of parameters λ_c), the Gaussian mixture density is then defined as:

$$p(\vec{x}|\lambda_c) = \sum_{i=1}^{M} w_i^c b_i^c(\vec{x}) \tag{1}$$

where each term b_i^c is D-variate Gaussian function (where D is the dimension of the input vector $\vec{x}$) of the form:

$$b_i^c(\vec{x}) = \frac{1}{(2\pi)^{D/2}|\Sigma_i^c|^{1/2}} \exp\left\{ -\frac{1}{2}(\vec{x}-\vec{\mu}_i^c)' \sum\nolimits_i^{-1} (\vec{x}-\vec{\mu}_i^c) \right\} \tag{2}$$

where $\vec{\mu}_i^c$ and Σ_i^c represent the mean vector and the covariance matrix of the i-th mono-Gaussian density for the claimed speaker c. The mixture weights satisfy the constraint $\sum_{i=1}^{M} w_i^c = 1$, which ensures that the mixture is a true probability density function. The covariance matrix Σ_i^c can be full or diagonal. To reduce the number of parameters, it is also possible to use a common covariance matrix for all component densities, the referred to as shared or global covariance matrix (which itself can be diagonal). Both of these constraints can be used to limit the number of free parameters that will have to be estimated from limited training data. So, the Gaussian mixture density for customer c is parametrized by the mean vectors, the covariance matrices, and the mixture weights:

$$\lambda_c = \left\{ w_i^c, \vec{\mu}_i^c, \sum\nolimits_i^c \right\} \qquad i = 1, \ldots, M. \tag{3}$$

During enrolment, given a sequence of speaker-specific training feature vectors, the GMM parameters are usually estimated according to a **Maximum Likelihood (ML)** procedure, typically the Expectation-Maximization (EM) algorithm, maximizing the likelihood of the observed data given the model (model's parameters). To avoid overtraining, the use of some cross-validation data may be used on which we verify the generalization properties of the model. It is also possible to use **Maximum A Posteriori (MAP)** estimation, in which some a priori information about the speaker-specific parameters (e.g., starting from speaker independent parameters) are used. This was successfully used by IRISA in the last NIST'99 campaign [9].

During testing, the GMM log likelihood (computed as the accumulated log likelihood over all the acoustic vectors) of the claimed speaker is compared to the GMM likelihood of the "world model" (estimating the likelihood that it is not the claimed speaker) and compared to a threshold above which the identity of the claimed speaker is validated. In other words, if the **likelihood ratio**

$$\frac{p(X|\lambda_c)}{p(X|\lambda_{\bar{c}})} \geq \Delta_c \tag{4}$$

where X represents the sequence of acoustic vectors and $\bar{c}$ represents all the speakers different than c (the "world model"), then the identity of the claimed speaker is validated, otherwise the speaker is rejected. Criterion (4) is often calculated in the log domain, so we have **log-likelihood ratio** criterion

$$\log p(X|\lambda_c) - \log p(X|\lambda_{\bar{c}}) \geq \log \Delta_c \tag{5}$$

As further discussed later, as well as in [3], different scoring and normalization strategies can be used at this stage.

3.3 NIST'99 competition

Since 1996, the National Institute of Standards and Technology (NIST) has coordinated evaluation campaigns of text-independent speaker recognition in the context of conversational telephone speech. Both academic and industrial laboratories take part each year in the evaluations whose main objectives are: exploring new ideas in text-independent speaker recognition, developing and implementing technology that will incorporate these ideas and assessment or measuring the performance of this technology. Each year, the evaluation conditions aim at specific goal. Since 1996, focus has been put on the effect of handset types. In 1999 though, more emphasis was put on multi-speaker recordings, and three tasks had been defined [10]:

One-speaker detection : This is the conventional task in which one must determine whether a target speaker is speaking during a given test segment or not. Each test segment was judged as true or false on eleven hypothesized speakers, one of which was the true speaker. In addition to this binary detection, a decision score was also required to fraw **Detection Error Tradeoff (DET)** performance curves showing how false rejections may be traded off against false acceptances as a function of the decision threshold. DET curves provide a convenient way to compare several systems (or the same system under several conditions) on the same graph. IDIAP participated only in this task.

Two-speaker detection : In this task test, segments contain both sides of a telephone call and each test segment was judged as true or false for each of twenty-two hypothesized speakers, two of them were really present in the test segments.

Speaker tracking : Like in the two-speaker detection task speech segments contain speech from two speakers, but now the task is to determine the intervals where a hypothesized speaker is speaking in the test segment.

In the following, after a brief description of the database used in NIST'99, the system used and evaluated by IDIAP will be described. Finally, results and comments will be given.

3.3.1 Database

The data for this evaluation were drawn from the Switchboard-2 Phase-3 corpus, collected by the Linguistic Data Consortium (LDC). This corpus consists of about 2700 recordings of five-minute telephone conversations between over 600 English speaking subjects. The majority of subjects were recruited from the South of the United States [11]. It was done to obtain speakers with similar dialects. Training data were provided for 230 male speakers and for 309 female speakers. For each speaker there was a total of two minutes of training data that originated from two separate conversations (one minute of each) from the same line number. Successive speech segments were concatenated together, removing silences. The nominal duration of 60 seconds varied slightly to allow whole segments to be included.

Test segments were collected in different sessions from the training data and can be originating from the same telephones as those used for recording the training data or from different ones. Telephone handsets can be of two different types, using *electret* or *carbon* based microphones, which showed to have a strong influence on recognition performance. The one-speaker test segments were created from the separate channels of each two-speaker segment by concatenating consecutive turns of particular speaker, i.e., corresponding to each summed speech segment in the two-speaker test there will be two segments in the one-speaker test, one from each original channel. The duration of test segment were between few seconds and one minute as opposed to last years' evaluations when different durations (3,10 and 30 seconds) were provided and evaluated separately.

The one-speaker detection task consisted of 37,620 trials, 3,157 of them being target trials while the remaining 34,463 were impostor trials.

3.3.2 Description of the IDIAP System

In the following, we briefly describe the main features of the IDIAP text-independent SV system used for the NIST'99 evaluation campaign.

Acoustic parameters: Sixteen LPC-cepstral coefficients (LPCC) were used from the 16th order LPC analysis. Sixteen delta LPCC and delta log energy were added, so the dimension of the input feature vector was 33. These coefficients were computed on 32ms window shifted every 10ms. Cepstral mean subtraction was also applied for channel equalization.

Modelling: Speaker models and the the "world model" (also called background) were based on 256 ($M = 256$ in (1)) component Gaussian Mixture Models with diagonal covariance matrices. Expectation-maximization algorithm was used for training these models with a maximum of 20 iterations. Two world models were used: handset dependent and gender independent, one for carbon and one for electret handset. Each model was trained on 30 seconds test segments taken from NIST'98 evaluation data: about 180 of test segments for electret and about 125 of them for carbon, half of them coming from females and half from males.

Scoring: The log-likelihood of the test segment is first computed with the claimed speaker model and then with the world model that corresponds to the handset type of the claimed speaker. The information of the handset type of the claimed speaker was available and could be read from the NIST header in the file. For a test utterance $X = \left\{\vec{x}_1, \ldots, \vec{x}_T\right\}$ and a claimed speaker model λ_c, the log-likelihood ratio is defined as:

$$\Lambda(X) = \log p(X|\lambda_c) - \log p(X|\lambda_{\bar{c}}) \tag{6}$$

The term $p(X|\lambda_c)$ is the likelihood of the test utterance given the model of the claimed speaker and $p(X|\lambda_{\bar{c}})$ is the likelihood of the utterance given it is not from the claimed speaker. In our case, this second term is the likelihood of the test segment given the world models, depending on the handset type of the claimed speaker, which can be estimated as: $p(X|W_{hds(\lambda_c)})$. The terms of the likelihood ratio are estimated as:

$$\log p(X|\lambda_c) = \frac{1}{T}\sum_{t=1}^{T} \log p(\vec{x}_t|\lambda_c) \tag{7}$$

where $p(\vec{x}_t|\lambda_c)$ is computed from (1). The $\frac{1}{T}$ scale is used to normalize the likelihood with respect to the utterance duration. However, we didn't use this frame length normalization whereas ENST (Paris) used it. The second term in (6) of the likelihood given the world model is computed in the same way.

Now we can rewrite log-likelihood ratio from (6) using different notation for the second term:

$$\Lambda(X) = \log p(X|\lambda_c) - \log p(X|W_{hds(\lambda_c)}) \tag{8}$$

where $W_{hds(\lambda_c)}$ represents the world model associated with the handset used by claimed speaker C.

Normalization and threshold setting: In the previous NIST campaigns, the **handset-dependent normalization (hnorm)** technique was applied to (8) and appeared to be an efficient way to compensate for the variability of the log-likelihood ratios across speakers and handset types. To apply *hnorm*, the distribution of impostor log-likelihood mean and standard deviation is determined for each speaker using same-sex impostor segments taken from the last year 30 sec test data. This is done separately for carbon and electret impostor segments and is used in (9) depending on the handset type of the test segment. The handset type of the test segment was available and could be read from the header of the file.

Based on this, we can define the normalized log-likelihood ratio as:

$$\Lambda'(X) = \frac{\Lambda(X) - \mu_{\lambda_c, hds(X)}}{\sigma_{\lambda_c, hds(X)}} \tag{9}$$

where $\Lambda(X)$ is from the equation (8), $\mu_{\lambda_c, hds(X)}$ and $\sigma_{\lambda_c, hds(X)}$ are log-likelihoods mean and standard deviation (respectively) computed for a claimed speaker on impostor segments of the same handset type as the one of the test segment X (usually denoted as hds(X)). This kind of normalization will be referred to as *hnorm*, for "handset-dependent normalization". As a possible alternative we also considered the *znorm*, where the mean and variance are handset-independent, i.e., they are not estimated separately for electret and carbon impostor segments, but are rather using the same mean mean and variance for each claimed speaker.

Finally, this normalized likelihood ratio is compared to a speaker and gender independent threshold Θ estimated to minimize the **Decision Cost Function (DCF)**. Consequently, if $\Lambda'(X) > \Theta$, the claimed speaker is accepted, while he/she will be rejected otherwise.

To set the threshold, tests were ran on a subset of the NIST'98 evaluation corpus (development set) for a two session training condition and all test durations (3, 10, 30sec). The DCF function was then defined as:

$$DCF = C_{fr} P_{target} P_{fr} + C_{fa} P_{\overline{target}} P_{fa} \tag{10}$$

where C_{fr} (resp. C_{fa}) is the cost of a false rejection (resp. of a false acceptance) and P_{target}(resp. $P_{\overline{target}}$) is the prior probability of a genuine speaker trial (resp. an impostor trial). For the NIST'99 evaluation, the costs (given by NIST) were set to $C_{fr} = 10$ and $C_{fa} = 1$ while the prior probabilities were $P_{target} = 0.01$ and $P_{\overline{target}} = 0.99$. In (10), P_{fr} and P_{fa} are the measured false rejection and false acceptance rates.

3.3.3 Results

During the experiments that were conducted in preparation for the competition, a little improvement in performance for 256 GMM's comparing to 128 GMM's was observed. Experiments were conducted for *unorm* (no normalization), *znorm* and *hnorm* and separately for the systems in which one- and two-world models are used (indeed, as explained above, there are two handset dependent world models one for electret and the other for carbon handset type whereas one-world model is handset independent).

Comparison of *unorm*, *znorm* and *hnorm* for the systems in which one-world and two-world models are used can be seen from the following two tables. In **Table 1**, the results for the system in which one-world model is used are shown. The first row in the table corresponds to the SNST case (same phone number and same handset type for training and test segments). We see that in this case (SNST) the results are better when there is no normalization (*unorm*). Handset-dependent normalization *hnorm* yields slightly worse performance than *unorm*, but better performance than *znorm*. In the case of DNST (different phone number and the same handset type) results are more or less equal what is marked as "=" in the table. Finally, in the case of DNDT (different phone number and different handset type), *hnorm* outperformed *unorm* and *znorm*.

In **Table 2**, the same experiments were repeated with a system using two world models. The reported results clearly show that the handset dependent normalization *hnorm* combined with handset dependent world models outperforms both *znorm* and *unorm*. As in the case of the system with one world model, *hnorm* was much better in the DNDT case.

	unorm	znorm	hnorm
SNST	1	3	2
DNST	=	=	=
DNDT	3	2	1

Table 1:The rank of results for the one-world model (2 sessions training and 30 seconds test segments)

	unorm	znorm	hnorm
SNST	2	3	1
DNST	3	2	1
DNDT	3	2	1

Table 2: The rank of results for the two-world model (2 sessions training and 30 seconds test segments)

However, tables (1) and (2) don't show what is the relation between two systems that use one and two-world models. It can be seen from the **Table 3** for the case of *hnorm* with the same training and testing conditions as above (two session training and 30 seconds test segments). The results show that the system with two world models is better in the mismatch case (DNST and DNDT) what is marked with the "+" sign in the table.

To conclude we can say that, in general, *hnorm* is better than *unorm* and *znorm* and that systems using two-world model were better in the mismatch case (DNST and DNDT).

	I world	II world
SNST	+	
DNST		+
DNDT		+

Table 3: Comparison of *hnorm* for the systems using one and two world models (2 sessions training and 30 seconds test segments)

On **Figure 2**, we can see DET curves for all NIST'1999 participants in a 1-Speaker Detection complete task (all trials). DET curve of IDIAP is marked as "idp1". Complete task means that tests from all three possible cases SNST, DNST and DNDT are put together. For each participant (in one specific color), the DET curve represents the probability of missing the speaker as a function of false alarm (detecting the speaker when not present) for different operation points of the decision threshold. Lower the DET is and better the results are.

On **Figure 3**, we can see IDIAP result for 1-Speaker Detection task all trials (left) and 1-Speaker Detection task Primary measure (right). Primary measure in NIST'99 evaluation was DNST, both training and test segments from electret microphones and test segment durations of 15-45sec. Results for the primary measure are a little better. It also means that results for SNST condition are worse. Both conclusions had been observed in experimentation phase before competition.

On **Figure 4**, we can see the performance comparison for all the systems of the **ELISA** Consortium in 1-Speaker Detection task. The **ELISA** is a Consortium founded in 1997 by a group of European laboratories, namely ENST (Paris), EPFL (Lausanne, Switzerland), IDIAP (Martigny, Switzerland), IRISA (Rennes, France) and LIA (Avignon, France), with the goal to build a common speaker recognition platform and participate to the NIST campaigns. The aim of the Consortium is to facilitate scientific communication and exchange in the field of text-independent speaker recognition sharing some of the software modules, resources and experimental protocols.

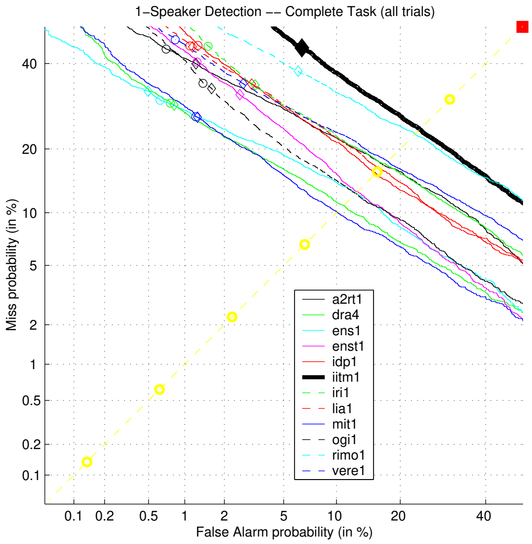
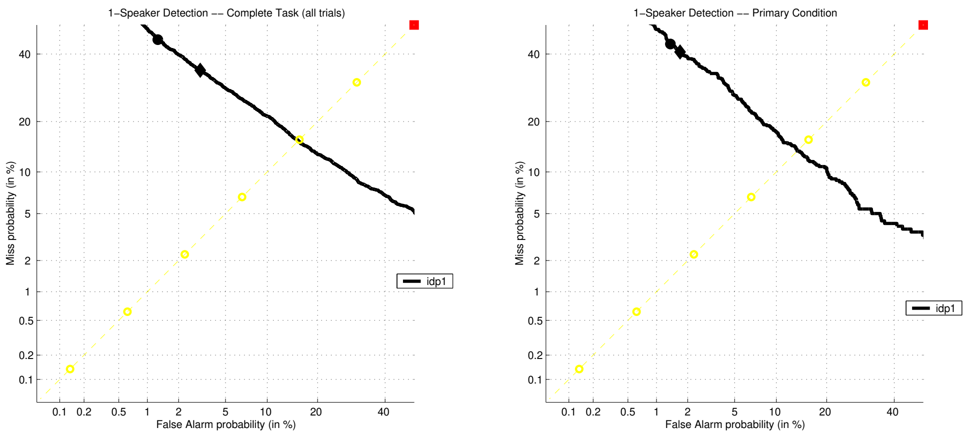


Figure 2: DET curves for 1-Speaker Detection complete test (all trials)

On **Figure 5**, we can see results of all members of the ELISA Consortium for 1-Speaker Detection task but separately for three possible cases (SNST, DNST and DNDT). We can notice that IDIAP results are better for the mismatch case (DNST and DNDT) than for SNST case. This result confirms the experimentation phase where the system with two world models showed better performance in the mismatch conditions. The IDIAP system showed good performance in the mismatch case, but the cost is worse performance in the match case (SNST). As the number of mismatch trials was pretty big, the overall performance was, however, good, what can be seen from the **Figure 4**.



(a) (b)

Figure 3: DET curves for IDIAP (a) 1-Speaker Detection complete task (all trials) and (b) 1-Speaker Detection Primary condition

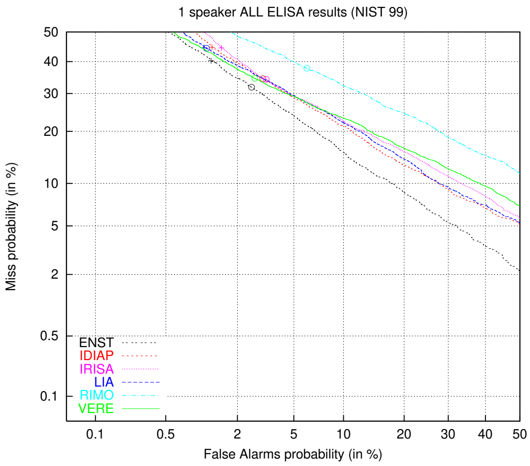


Figure 4: DET curves for 1-Speaker Detection complete test (all trials) for ELISA

4 Text-Dependent Speaker Verification

4.1 Introduction

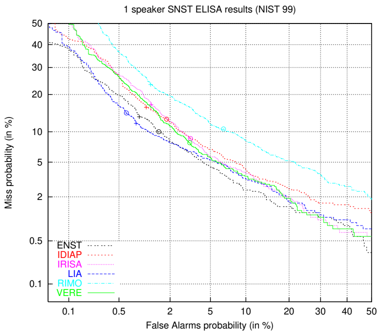
In text-dependent speaker verification systems, training and test utterances should contain the same lexical information. In this case, model of each speaker contains both speaker specific characteristics as well as characteristics of lexical content of the utterance. Since this method exploits directly the voice individuality associated with each phoneme or syllable, it generally achieves higher recognition performance than the text-independent method. In general, because of the higher acoustic-phonetic variability of text-independent input, more training material is necessary to reliably characterize (model) a speaker than with text-dependent methods.

The text-dependent methods are usually based on template matching techniques in which the time axis of an input speech sample and each reference template or reference model of registered speakers are aligned, and the similarity between them, accumulated from the beginning to the end of the utterance, is calculated. We distinguish two principal approaches in text-dependent speaker verification [3]:

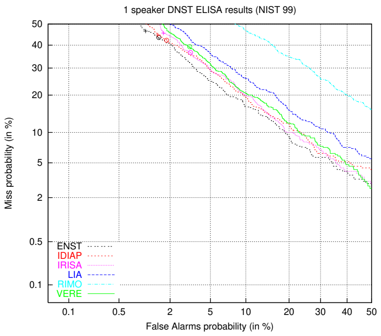
1. DTW (Dynamic time warping) methods. Each utterance (password) is represented by a sequence of small number acoustic templates. In the recognition phase a score of a new test utterance is computed via dynamic time warping algorithm against the reference models (templates). This method is rather simple and doesn't require much computational resources in the enrollment phase [12].

2. HMM methods. In this case, each password is modeled by a HMM. This model is not necessarily fully-connected (as in text-independent speaker verification) since it models a particular word. The parameters of the HMM's are trained from repetitions of the password and the amount of training material can be a problem in the practical applications if the used HMM's are large. HMM methods achieve better recognition performance compared to DTW methods (as also observed in speech recognition) but at the cost of higher computational requirements in the enrollment phase [13].

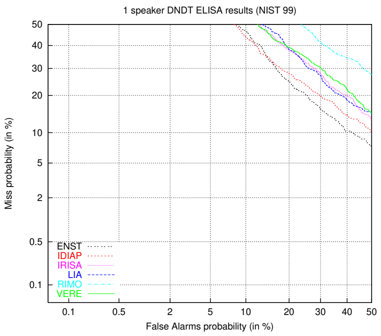
Text-dependent speaker verification system usually yields better performance than text-independent SV. However, it is less user-friendly since the password cannot be changed easily and is often imposed to the user. Consequently, a very desirable feature in real-world applications is that a user can select his/her own password instead of text chosen by the system (for example the speaker is prompted to



(a) SNST



(b) DNST



(c) DNDT

Figure 5: DET curves for ELISA systems 1-Speaker Detection task: (a) SNST(same number, same handset type), (b) DNST(different number, same handset type), (c) DNDT(different number, different handset type)

repeat random sequences of words in the text-prompted speaker verification). Although the techniques briefly mentioned above are still valid for SV based on user-customized password, one of the main problems is to infer the relevant models directly from the acoustics.

4.2 Speaker Verification based on User-Customized Password

4.2.1 Problem

In the case of text-prompted speaker verification, the system knows in advance the (prompted) text, and thus the HMM model associated with the training and test utterances. This model can be whole word HMM model or obtained by concatenating speaker-specific phoneme models. In user-customized password though, the password, and thus its associated model, is a priori unknown. Starting from a few pronunciations of his/her personal password, enrolment of a customer will thus generally consist of the following two steps [14]:

1. *HMM inference*: To first automatically **infer** the best HMM model associated with the user-specific password from a few (usually two or three) repetitions of the password. The inferred model should be representative of the lexical content of the password (for example, a sequence of pseudo phones).

2. *HMM adaptation*: The parameters of this model will then have to be adapted to the characteristics of the customer's voice.

So far, our work in user-customized SV has mainly focused on hybrid HMM/ANN systems, using **Artificial Neural Networks** (ANN) to estimate local posterior probabilities used as HMM emission probabilities. More specifically, and similarly to what has been done in speech recognition, we used a **Multilayer Perceptron** (MLP) with acoustic context (9 frames), to compute local posterior probabilities, which are then used to infer HMM topologies, and whose parameters will also be adapted to the customer. Although the approach discussed below is also valid for other probability estimators (such as Gaussian densities), we believe that there are several reasons to use MLPs, including:

- MLPs can accurately estimate posterior probabilities [20, 21], which are known to minimize classification error rates.

- Powerful (nonlinear) modeling capabilities, including the possibility of capturing some of the temporal correlation information (by providing the MLP input with some contextual information, in our case 9 successive acoustic frames).

- As a result, and most importantly in our case, MLPs usually demonstrate excellent phonetic recognition rate, even at the single acoustic frame level (typically above 70%) [2], which makes them particularly amenable to perform automatic inference of the HMM topologies from a few utterances of the password. This will be confirmed below, showing that a correct phonetic frame classification rate of 77SV database, after having trained the MLP on an independent telephone database (not specific to our SV task).

- This good phonetic recognition performance can be achieved with context independent phone models only (no need to model and use a large and complex set of context dependent phone models), making MLP even better suited to (lexicon independent) HMM inference.

4.2.2 General Approach

In the following, we briefly summarize the approach we are implementing, before presenting the current status of our work in the next sections. Our user-customized enrolment and verification system is first based on:

1. A well-trained **speaker independent MLP, of parameters** Θ, and which is known to perform very well on phonetic classification at the acoustic frame level. In our case, the MLP was trained on the *PolyPhone* (Swiss French) database [15], a large telephone database containing prompted and natural sentences pronounced by a large number of different speakers. The MLP consisted of 234 inputs (nine consecutive frames with 26 acoustic features each), 600 hidden units and 36 outputs (associated with 36 phones). Such an MLP typically achieved a frame-based phonetic recognition rate higher than 85% on the *PolyPhone* test data, as well as on the data that will be used here for speaker verification (see below).

2. A **world HMM model** M, defined as an ergodic (looped) HMM, of parameters Θ (in our case coming from the ANN), together with transition probabilities (which are usually not trained). In our case (hybrid HMM/ANN system), each phoneme was represented by a single state with a minimum duration constraint or with transition probabilities reflecting this duration constraint. As discussed below, this world model will be used (1) to infer the HMM model associated with the user specific password, and (2) to normalize the utterance likelihood (conditioned on the speaker specific model) during verification (denominator of the likelihood ratio).

The general **enrolment** steps can then be summarized as follows:

1. A new customer S_k pronounces J (typically, 2 or 3) times his/her password X_k^j, $j = 1, \ldots, J$, where X_k^j represents the sequence of acoustic vectors associated with the j-th utterance.

2. Match each of the enrolment utterances X_k^j with M, using the speaker independent parameters Θ, to generate a phonetic transcription of each utterance, together with its associated likelihood.

3. Choose the phonetic transcription yielding the highest likelihood, and use it to build a reference HMM model M_k representing the password of customer S_k. This HMM model is simply built up by concatenating, strictly left-to-right (with only loops and skips to the next state), HMM states corresponding to each of the phone in the phonetic sequence. Variants or alternative approaches can also be considered, such as:

   (a) Inferring a general HMM model from the multiple phonetic sequences associated with the enrolment utterances and resulting from the matching with M.

   One solution could consist in merging several phonetic sequences using dynamic programming algorithm as in [16]. The other solution as proposed in [17] consists in firstly finding the most probable phonetic sequence for each utterance, then force-align each of the transcriptions on each of the utterances. This gives the probability for each of the transcriptions to each of the input utterance. The best transcription is the one for which the product of the probabilities obtained for different input utterances is maximal, i.e., it is the sequence that is most likely to produce all utterances.

   (b) If at least three enrolment utterances are used, check that they are all similar enough, and possibly reject outliers which could correspond to a different password. In this case, prompt the user for a new pronunciation of the password.

4. Match each of the enrolment utterances X_k^j, $j = 1, \ldots, J$ on the speaker specific model M_k to yield phonetic segmentation of these utterances.

5. As for the training of HMM/ANN systems, adapt the ANN parameters Θ by using the above segmentation to provide the ANN target outputs and to minimize (by the usual back-propagation algorithm) the square error between the observed output vector generated by each input vector of the enrolment utterances and the associated target output vector (obtained from the above segmentation). Given the large number of ANN parameters, it is expected that different training schemes will have to be investigated and tested, including

   (a) Use of cross-validation (e.g., on an extra enrolment utterance not used for training)

   (b) Keeping the weights of the speaker independent ANN fixed and only training the weights of an additional linear input layer (LIN) [18] as discussed in Section 4.3.2.

   (c) Possibly together with the above approaches, a large weight decay could also be used to prevent overtraining.

   This yields a speaker specific and user-customized MLP model (because of (4)), with parameters Θ_k.

6. Once the speaker-adapted ANN has been trained, possibly perform an additional Viterbi alignment on the HMM associated with the user-customized password (initially inferred with the speaker independent ANN) using the speaker specific ANN, thus generating new ANN targets used for further ANN training.

The **speaker verification** approach should then be the following. To verify the identity of a speaker S, pronouncing X, and claiming to be S_k:

1. Load the HMM model M_k associated with the password of S_k, the speaker specific MLP model of parameters Θ_k, the world HMM model M with its associated MLP parameters Θ.

2. Perform Viterbi matching of X on model M_k and parameters Θ_k to compute $P(X|M_k, \Theta_k)$, representing the likelihood that X was actually produced by S_k (since using Θ_k) pronouncing M_k.

3. Perform Viterbi matching of X on model M and parameters Θ to compute $P(X|M, \Theta)$, representing the probability that X was generated by another speaker (which, as for text-independent speaker verification, is estimated here from a speaker independent model) pronouncing any word (hence the looped world model). Another possibility, underestimating the likelihood of the world model, would be to use $P(X|M_k, \Theta)$, thus estimating the probability that the right password has been produced by a speaker different than S_k.

4. Compute the following likelihood ratios

$$\mathcal{L}_k^1 = \frac{P(X|M_k, \Theta_k)}{P(X|M, \Theta)} \gtrless \Gamma_k^1 \tag{11}$$

   and/or

$$\mathcal{L}_k^2 = \frac{P(X|M_k, \Theta_k)}{P(X|M_k, \Theta)} \gtrless Gamma_k^2 \tag{12}$$

   and check whether or not these are above some speaker-specific thresholds Γ_k^1 and Γ_k^2.

In the case of an impostor access, we then have two possibilities:

1. The impostor is pronouncing the right password: in this case, criterion (12) is probably better than (11) since we are estimating each likelihood with the right HMM model, and it is highly probable that the numerator (using customer specific parameters) will be smaller than the denominator (using speaker independent parameters).

2. The impostor is not pronouncing the right password: in this case, criterion (11) will probably be better than criterion (12). Indeed, in the case of (11), both the numerator and the denominator will have good matching likelihoods (since matching is performed on the wrong model), and it is thus difficult to predict the behaviour of the criterion. Using (12), however, the denominator (obtained by matching the utterance on the world model) will always yield a good score, while the numerator score will be very bad since we are matching the utterance to the wrong model, using the wrong speaker specific parameters.

In the case of a customer access:

1. If the customer does not pronounce the right password, criterion (11) will still yield a low score (given that the numerator will yield a low score, while the denominator will still yield a good score), thus rejecting the access.

2. If the customer uses the right password, the likelihood $P(X|M_k, \Theta_k)$ of the numerator should be higher than $P(X|M_k, \Theta)$ or $P(X|M, \Theta)$. Thus, although (12) should provide a more precise estimate of the likelihood ratio, criterion (11) should also yield good user-customized speaker verification performance.

In conclusion, it thus seems that the best compromise is to always use criterion (11), which will yield a good score only in the case of a customer pronouncing the right password. Although the denominator of the likelihood ratio is then always overestimated, it can be expected that the decision threshold will be able to accommodate this.

It is worth noting here that $P(X|M, \Theta_k)$ could also be used to perform text-independent speaker verification.

In the following sections, after a brief description of the databases used, we briefly discuss the current status of our work regarding HMM inference and HMM adaptation.

4.3 Databases and Initial Setup

In the following, we used the *PolyPhone* database for training the speaker-independent (world) model, and the Swiss-French *PolyVar* [15] to perform customer enrolment and speaker verification tests.

The Swiss-French *PolyPhone* database [15] contains telephone calls from about 4,500 speakers recorded over the Swiss telephone network. The calling sheets were made up of 38 prompted items and questions and were distributed to people from all over French speaking part of Switzerland. Among other items, each speaker was invited to:

- Read 10 sentences selected from different corpora to ensure good phonetic coverage for the resulting database.

- Simulate a spontaneous query to telephone directory (given the name and the city of subject), i.e., simulate a 111 information service call.

The training of the speaker-independent (world model) MLP used here was performed on a subset of this *PolyPhone* database, using the 10 phonetically rich sentences read by each of a database subset of 400 speakers (200 male and 200 female speakers). As already mentioned in Section 4.2.2, the MLP contained 234 inputs (representing nine consecutive acoustic frames with 26 features each), 600 hidden units and 36 outputs (associated with 36 phones). The output nonlinearity was the "softmax" function, ensuring that the class posterior probability outputs always sum up to one.

For capturing intra-speaker variability, the *PolyVar* database was designed and recorded at IDIAP, as a complement to the Swiss French *PolyPhone* database to address inter-speaker variability issues. In this database, several speakers pronounced the same set of words several times (5 times, spread over time), which makes it particularly well suited to test user-customized speaker verification systems, e.g., by:

- Assigning each of the words to one specific customer, thus

- Providing 3 enrolment utterances of that word, as well as 2 "accept" test utterances, as well as many impostor utterances pronouncing the right password.

- Providing several utterances associated with words different than the chosen password, from both the customer and potential impostors.

In the following experiments, we have chosen a subset of speakers from a *PolyVar* database and have assigned each of them a specific password. Since the number of speakers is larger than the number of words that are used in this specific subset, some speakers will have the same passwords, which is actually good since it reflects what could happen in the context of user-customized password SV, where different customers could have the same password.

After training on the *PolyPhone* subset, the phonetic recognition rate at the frame level on a *PolyPhone* test set was above 85% [19]. On a subset of the *PolyVar* database, recorded on different conditions, and on which we will test our speaker verification system, the correct frame recognition rate was still of 77,6%.

4.3.1 HMM Inference

As briefly discussed above (point 2 in "general enrolment steps" from section 4.2.2), the first enrolment step of a user-customized speaker verification system is to automatically infer a HMM topology from one or a few pronunciation of the password and a good speaker independent set of phonetic parameters.

In our case, this inference is performed by doing phonetic decoding of the enrolment utterances, matching each enrolment acoustic sequence X on an ergodic HMM word model M (containing the set of phonetic states, fully connected, each of them being associated with a particular MLP output) with minimum state duration constraints – see **Figure 6** (in the case of a minimum duration constraint of 4 for all phones). On top of this minimum duration constraint (which could be specifically tuned to each phone), we can also introduce a phone transition probability, represented in terms of a, the probability of staying on the last state of a phone (probability of staying more than the imposed minimum duration) and, consequently, $(1-\mathbf{a})/(\mathbf{K}-\mathbf{1})$ the probably of changing phone (where K denotes the total number of phones).

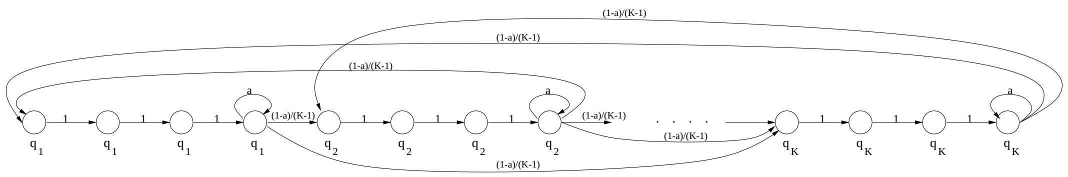


Figure 6: The ergodic HMM model with the constraint of staying a given numer of frames in the same state (four in this case).

For each acoustic sequence $X = \{x_1, \ldots, x_n, \ldots, x_N\}$ associated with a pronunciation of the user-specific password, the speaker-independent (world) MLP of parameters Θ provides, for each acoustic frame x_n, the posterior probabilities $P(q_k|x_n, \Theta)$, for $k = 1, \ldots, K$, of the K different phones q_k associated with the MLP outputs. Using these phone posterior probabilities, a simple Viterbi algorithm is applied on the world HMM model M to estimate the underlying phonetic sequence. Depending on the minimum duration constraint, the inferred phonetic transcription will be more or less smooth, and will more or less correspond to the actual phonetic transcription of the password. Below, we give a typical phonetic sequence obtained at the output of the inference algorithm for the French word "annulation", together with its associated phonetic segmentation.

Inferred phonetic sequence:
[sil] [aa] [nn] [uu] [ll] [aa] [ss] [yy] [on] [sil]
Associated phonetic starting points (frame numbers):
[0] [30] [38] [46] [53] [58] [67] [77] [90] [99]

4.3.2 Speaker Adaptation

This section mostly addresses point 5 of the "general enrolment steps" presented in Section 4.2.2. As explained there, the next step would be to adapt the speaker independent MLP parameters to each customer, using the enrolment utterances and the inferred HMM. This training can be done "from scratch" (i.e., starting from a random initialization of the weights), or starting from the speaker independent weights. In the initial experiments reported below, we decided to start from scratch to avoid the effect of a potential local minimum. Obviously, it is clear that we are attempting to train/adapt a large number of parameters, compared to the very small size of the training material, and that we will thus quickly observe overtraining. However, in these preliminary experiments, we were also interested to see how much overtraining (possibly limited through the use of cross-validation) would hurt generalization performance on the targeted speaker, compared to impostors. It could indeed well be that even an overtrained MLP still performs significantly better on the adapted speaker than on the impostors.

To perform these preliminary tests, we thus took from *PolyVar* five (different) repetitions of the same word of the same speaker. Three of them were for training the MLP, while the last two were used to test the generalization properties. These generalization properties were also compared to other speakers (impostors) pronouncing the same password, or to the customer (and impostors) producing 2 (and 5) test utterances of the correct password.

During MLP training, the standard error back-propagation algorithm was used to minimize a least mean square error (LMSE) criterion defined as:

$$E(X|\theta, \theta_k) = \frac{1}{N}\sum_{n=1}^{N} \|g(x_n, \theta_k) - d(x_n, \theta)\|^2 \tag{13}$$

where $X = \{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ is the acoustic vector sequence, e.g., associated with the three enrolment utterances (x_n being one of the input vectors and N the total number of training acoustic frames), and $g(x_n, \theta_k)$ being the MLP output vector (set of class posterior probabilities) given the current value of the speaker specific MLP parameters θ_k, i.e.:

$$g(x_n, \theta_k) = (g_1(x_n, \theta_k), \ldots, g_J(x_n, \theta_k))^t \tag{14}$$

with J being the number of phones (J=36, in our case). The targeted output vector $d(x_n, \theta)$ associated with each input vector x_n is given by:

$$d(x_n, \theta) = (d_1(x_n, \theta), \ldots, d_j(x_n, \theta), \ldots, d_J(x_n, \theta))^t \tag{15}$$

where $d_j(x_n, \theta) = \delta_{j\ell}(x_n, \theta)$, if q_ℓ represents the HMM state class associated with x_n and corresponding to the phonetic segmentation obtained from matching the inferred HMM model M_k on the enrolment utterances and using an MLP (emission probabilities) of parameters θ. Usually, the adaptation process will be initiated with the segmentation obtained from the speaker independent MLP (of parameters θ). However, this will then be followed by embedded Viterbi training where at iteration t the speaker specific MLP parameters θ_k^t are used to compute emission probabilities for matching the enrolment utterances on model M_k, thus yielding a new segmentation that can be used as targets to train a new set of parameters $\theta_k^{(t+1)}$.

As a preliminary test of this approach, **Figure 7** represents the resulting LMS error, on the training and different test data, as a function of number of MLP training iterations, and using targets obtained from the segmentation resulting of the speaker independent world model (i.e., the first iteration of the embedded Viterbi adaptation). As expected, we observe that the error (LMS) function keeps decreasing on the training data (3 enrolment utterances of the password "annulation" for speaker "f44") as a function of the number of iterations. The red curve represents the evolution of the LMS error over two test utterances of the same password (annulation) by the same speaker (f44). As expected, this LMS starts by decreasing, up to a point where the MLP starts overtraining and the LMS starts increasing. Ideally, the speaker adaptation process should stop at the minimum LMS by using some cross-validation data. However, interestingly, even in the case of overtraining the resulting average LMS on the test data is still better (lower) than the average LMS obtained in the case of a different speaker (f56) and the same word (annulation), as illustrated by the yellow curve. As could have been expected, this difference is even bigger in the case of a different speaker (m02) and a different word (gianadda), as illustrated by the green curve. Although these are still very preliminary results, they illustrate the potential of the proposed approach.

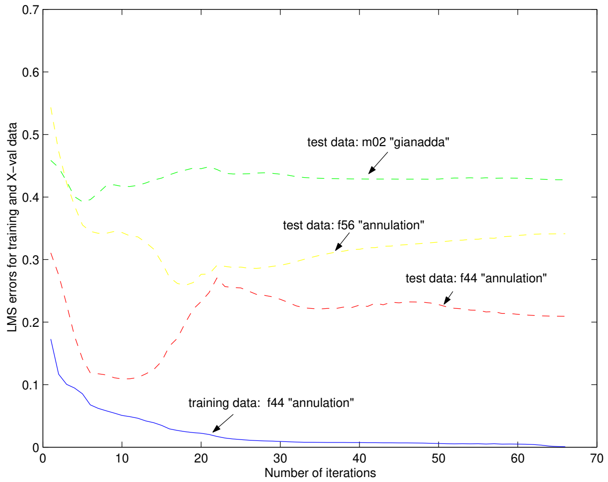


Figure 7: LMS errors normalized per frame for training (blue) and test data (red, green and yellow)

In the future, we will investigate different adaptation techniques, avoiding overtraining as much as possible, and will study the different scoring techniques, score distributions and equal error rates resulting of (11) and (12). To limit the risk of overtraining, on top of the obvious cross-validation techniques, we will also investigate the possible of the linear input network, as briefly described below.

The Linear Input Network: The idea is to use an adaptation technique that has been shown to be quite efficient for speech recognition systems. In this case, a **LIN (Linear Input Network)** [18] will be used to map the speaker specific input feature vectors to the speaker-independent system. As illustrated by **Figure 8**), the idea is to put a small linear input layer whose weights will be adapted by a gradient procedure (error back-propagation) to minimize the LMS criterion, keeping the speaker-independent parameters frozen. After having adapted the parameters of this linear transform, the associated weights can easily be integrated into the initial input-to-hidden speaker independent weights, thus resulting in a speaker adapted MLP.

The idea with LIN is particularly appropriate for several reasons:

- Speaker-independent MLP is usually very large (in our case it has 162,636 parameters) and to train such network very large amount of adaptation material would be necessary. In the context of user-customized password application the user usually pronounces only two times his password. So, the number of parameters to be adopted, in a case of LIN, would be $N \times N$ where N is the number of input units (in our case it is almost three times less). Reducing the number of free parameters in this linear transformation, it is then possible to even reduce further the number of parameters to be adapted.

- Since LIN is simply a linear transformation matrix, it will be easily included after training in the speaker independent MLP just by multiplying this matrix with the initial input-to-hidden MLP weight matrix.

- Since the training of the LIN is performed by a gradient procedure it is possible to use online

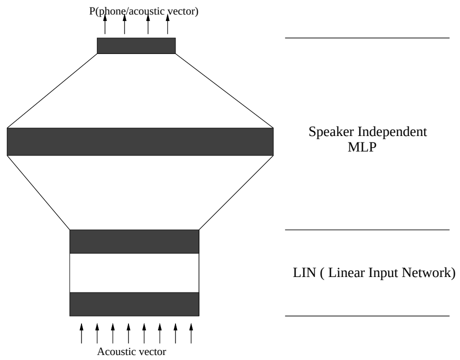


Figure 8: Speaker-independent MLP with added Liner Input Network (LIN) at the input.

gradient training to update parameters each time the password is validated, which could be important in the context of online adaptation to speaker medium/long term variabilities.

5 Conclusions and directions for the future work

In this report, we first discussed our latest work in text-independent speaker verification systems based on Gaussian Mixture Models (GMM), including the results obtained in the framework of the last NIST (National Institute of Standard and Technology, USA) competition. In this case, we mainly enhanced our existing system by improving the scoring and decision rules, mainly by introducing handset specific models, normalization, and thresholds.

In the second part of the present report, we presented our initial views on the development of a user-customized password SV verification, involving automatic HMM inference and user adaptation, based in our case on hybrid HMM/ANN systems. In this framework, an HMM inference software has been developed and adapted from the PICASSO project to allow the inference of HMM word models (user-customized models) in terms of the phonetic transcription generated from a few pronunciations of the user-specific password. A new neural network (MLP) training program was also written to allow quick speaker adaptation of the MLP parameters. Starting from a speaker independent network (trained on a large speaker independent database), each speaker-specific neural network can be adapted on the enrollment sentences containing the user-customized password to maximize the likelihood of the inferred HMM models. Initial tests to measure the discriminant capabilities of these networks with respect to same speakers (customers) and impostors were carried out, showing the real potential of the approach.

Our future work will now concentrate on investigating different HMM inference techniques, as well as different adaptation schemes (including cross-validation and LIN). Different feature parameters that better encode speaker specific information should also be tested. Finally, different normalization and decision threshold strategies will be compared. For example, it was shown in [22] that a normalization based on a posteriori probabilities was yielding better results. It is thus interesting to validate this conclusion in the context of our work, mainly based on posterior probabilities.

References

[1] Sadaoki Furui. An overview of speaker recognition technology. *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, Martigny, Switzerland*, pages 1–9, 1994.

[2] Herve Bourlard and Nelson Morgan. *Connectionist Speech Recognition - A hybrid Approach*. Kluwer Academic Publishers, 1994.

[3] H. Bourlard and N. Morgan. *Speaker Verification: A Quick Overview*. IDIAP research report, 1998.

[4] C. Montacie, P. Deleglise, F. Bimbot, and M.J. Caraty. Cinematic techniques for speech processing: temporal decomposition and multivariate linear prediction. *Proc IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 1:153–156, 1992.

[5] F.K. Soong, A.E. Rosenberg, L.R. Rabiner, and B.H. Juang. A vector quantization approach to speaker recognition. *Proc IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, pages 387–390, 1985.

[6] D.A. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, Martigny, Switzerland*, pages 27–30, 1994.

[7] J. Oglesby and J.S. Mason. Optimization of neural models for speaker identification. *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, (Albuquerque, NM)*, pages 261–264, 1990.

[8] Y.-H. Kao, P.K. Rajasekaran, and J.S. Baras. Free-text speaker identification over long distance telephone channel using hypothesized phonetic segmentation. *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, (San Francisco)*, 2:177–180, 1992.

[9] M. Seck, R. Blouet, and F. Bimbot. The irisa/elisa speaker detection and tracking systems for the nist'99 evaluation campaign. *DSP Journal (Special Issue on the Nist Speaker Recognition Workshop)*, 1(1), 2000.

[10] B. Nedic, F. Bimbot, R. Blouet, J.-F. Bonastre, G. Caloz, J. Cernocky, G. Chollet, G. Durou, C. Fredouille, D. Genoud, G. Gravier, J. Hennebert, J. Kharroubi, I. Magrin-Chagnolleau, T. Merlin, C. Mokbel, D. Petrovska, S. Pigeon, M. Seck, P. Verlinde, and M. Zouhal. The elisa systems for the nist'99 evaluation in speaker detection and tracking. *DSP Journal (Special Issue on the Nist Speaker Recognition Workshop)*, 1(1), 2000.

[11] M. Przybocki and A. Martin. The 1999 nist speaker recognition evaluation, using summed two-channel telephone data for speaker detection and speaker tracking. *Eurospeech*, 5:2215–2218, 1999.

[12] S. Furui. Cepstral analysis technique for automatic speaker verification. *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 29:254–272, 1981.

[13] A.E. Rosenberg, C.-H. Lee, and S. Gokcen. Connected word talker verification using whole word hidden markov modeling. *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (Toronto,Canada)*, pages 381–384, 1991.

[14] F. Bimbot, M. Blomberg, L. Boves, G. Chollet, C. Jaboulet, B. Jacob, J. Kharroubi, J. Koolwaaij, J. Lindberg, J. Mariethoz, C. Mokbel, and H. Mokbel. An overview of the picasso project research activities in speaker verification for telephone applications. *Eurospeech*, 5:1963–1966, 1994.

[15] G. Chollet, J.-L. Cochard, A. Constantinescu, C. Jaboulet, and Ph. Langlais. Swiss french polyphone and polyvar: telephone speech databases to model inter- and intra-speaker variability. research report 01, IDIAP, April 1996.

[16] M.G. Thomason and E. Granum. Dynamic programming inference of markov networks from finite sets of sample strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):491–501, 1986.

[17] Houda ABI AKL MOKBEL. *Inference de variantes de prononciation a partir de signaux acoustiques pour la reconnaissance automatique de la parole.* PhD thesis, Universite de Rennes 1, 12 1998.

[18] J. Neto, C. Martins, and L. Almeida. Speaker-adaptation in a hybrid hmm-mlp recognizer. *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 1996.

[19] J.M. Anderson, G. Caloz, and H. Bourlard. Swisscom "avis" project (no. 392) advanced vocal interfaces services technical report for 1997. Technical Report 06, IDIAP, December 1997.

[20] H. Gish. A probabilistic approach to the understanding and training of neural network classifiers. *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, pages 1361–1364, 1990.

[21] H. Bourlard and C.J. Wellekens. Links between markov models and multilayer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1167–1178, 1990.

[22] T. Matsui and S. Furui. Concatenated phoneme models for the text-variable speaker recognition. *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2:391–394, 1993.