IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

Speaker Verification Based on User-Customized Password

Mohamed F. BenZeghiba [a] Hervé Bourlard [a,b]
Johnny Mariethoz [a]

IDIAP–RR 01-13

May 15, 2001

revised in October 01

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] Institut Dalle Molle d'Intelligence Artificielle Perceptive (IDIAP), Martigny
[b] Swiss Federal Institute of Technology at Lausanne (EPFL), Switzerland

IDIAP Research Report 01-13

Speaker Verification Based on User-Customized Password

Mohamed F. BenZeghiba Hervé Bourlard Johnny Mariethoz

May 15, 2001

revised in October 01

Abstract. In this report, we describe the approach used so far to implement and test a baseline version of a speaker verification system based on user-customized password, i.e., where the user can choose his/her own password in a short enrolment phase involving a few pronunciations of the password. These developments involved:

- Good formalisation of the theoretical background, including hidden Markov model (HMM) inference, parameter adaptation, scoring and decision threshold.
- Implementation of the automatic hidden Markov model (HMM) inference system, using local phonetic probabilities generated at the output of a speaker-independent artificial neural network (ANN). Given a few utterances of a specific password, this program will generate the most probable HMM topology.
- Adaptation of the ANN parameters towards the targeted speaker: Given the inferred HMM topology and a few utterances of the password, this software module adapts the parameters of the speaker independent ANN to better model the Characteristics of the user's voice. Different adaptation approaches (through ANN constraints), aiming at minimizing the number of parameters to be adapted, hence guaranteeing optimal generalisation properties, have been tested and compared.
- Finally, development of different scoring criteria and testing them on large reference databases.

The resulting system was extensively tested on PolyVar (a reference speaker verification task), yielding initial estimates of the Equal Error Rate (EER) performance. These tests were performed under the least favorable conditions where all the speakers would have the same password. Initial tests (discussed in the report enclosed here) show reasonable performance, although not competitive yet with state-of-the-art text-independent speaker verification systems. It looks like the main issues that will have to be investigated now are related to the improvement of the discriminant properties of the ANN between different speakers.

Contents

1 Introduction

This report presents initial results obtained with a particular form of **Speaker Verification (SV)** system allowing the speaker to use customized passwords.

Many automated services (i.e., electronic banking, electronic shopping, credit cards and calling cards) require secured access. Today, the most common approach towards access control consists in using a PIN (Personal Identification Number) code, possibly together with a password, that the customer has to enter (i,e., via a keyboard) to identify himself/herself. This approach, however, still suffers from several limitations, including the risk of fraud (in case an impostor has access to the PIN code and/or password), and the necessity to have access to a keyboard to enter the data.

For improved security and flexibility, the possibility of using the customer's voice print, by its own or as an additional security feature, is often considered. Voice prints indeed have characteristics that are specific to each user and are difficult to reproduce. Besides this, speaker verification is easy to use, with your personal print (voice) always available and not easily lost or stolen. Indeed, speech contains many characteristics that are specific to each individual, many of which are independent of the linguistic message. Furthermore, due to the development of telecommunications and speech recognition technologies, many voiced-based services are becoming available. In this case, access control based on speaker verification is even more important, and provides the only truly practical solution.

Speaker recognition is a generic term for the classification of a speaker's identity from an acoustic signal [6]. This problem can be divided into speaker identification and speaker verification. Today, speaker recognition has many potential applications, including: secured use of access cards (i.e., calling and credit credits), access control to databases (i.e., telephone and banking applications), access control to facilities, electronic commerce, information and reservation services, remote access to computer networks, etc.

In the case of **speaker identification**, the speaker is classified as being one of a finite set of speakers. As in the case of speech recognition, this will require the comparison of a speech utterance with a set of known (registered) references for each potential speaker.

For the case of **speaker verification**, the speaker is classified as having the purported identity or not. That is, the goal is to automatically accept or reject an identity that is claimed by the speaker. In this case, the user will first identify herself/himself (i.e., by introducing or uttering a PIN code), and the distance between the associated reference and the pronounced utterance will be compared to a threshold that is determined during training.

Speaker identification thus requires $N + 1$ decisions for a population size of N speakers (deciding to associate the unknown voice as belonging to one of the N registered speakers or as being none of them). Speaker verification, on the other hand, simply involves an hypothesis testing and thus requires a simple binary decision, i.e., accept or reject the claimed identity, regardless of the population size. Therefore, speaker identification performance will tend to decrease as the population size increases while speaker verification performance is quite independent of the population size.

Speaker identification and verification can be based on **text-dependent** or **text-independent** utterances, depending on whether or not the recognition process is constrained to a pre-defined text or not. In the case of text independent speaker verification, the lexical content of the utterance used for verification cannot be predicted. In text-dependent speaker verification, the system knows in advance the access password (or sentence) that will be used by the user. For each individual, there is a model that encodes both the speaker characteristics as well as the lexical content of the password.

Since verification is based on both the speaker characteristics and the lexical content of a secret password, text dependent speaker verification systems are generally more robust than text-independent systems. However, both kinds of systems (text-dependent and text-independent) are susceptible to fraud, since for typical applications the voice of the speaker could be captured, recorded, and reproduced. In the case of a text-dependent system, even a password could be captured. To limit this risk, particular kinds of text-dependent speaker verification systems based on **prompted text** have been developed. In this case, for each access, a recorded or synthetic prompt will ask the user to pronounce a different random sentence [5]. The underlying lexicon which could either be very large or limited to just the 10 digits, would then be used to generate random digit strings. The advantage of such an approach is that impostors cannot predict the prompted sentence. Consequently, pre-recorded utterances from the customer will be of no use to the impostor.

In the present report, we study yet another approach towards text-dependent speaker verification, based on **user-customized password**. In this case, each user can choose his/her own password in a short enrolment phase involving a few pronunciations of the password. Although such a system is still subject to fraud, market studies have recently shown that customers usually felt more comfortable in having the possibility to choose their own password.

2 User-Customized Password Based on Hybrid HMM/ANN

In real-world services, it often desirable to give the possibility to the user to choose his/her own password (on which verification will be performed) with no constraints in vocabulary words. Such a system, referred to as **User-Customized Password Speaker Verification (SV-UCP)**, should increase performance, flexibility and security. Indeed, for each individual, there is a model that encodes both the **speaker characteristics** as well as the **lexical content of the password**. Speaker validation can thus use these two features. It is also more flexible for the user, who can choose the password. Finally, given that the password is chosen from an unconstrained vocabulary, it makes it more difficult to an impostor to guess the customer's password.

However, SV-UCP presents new difficulties since the system has to automatically infer the HMM model associated with the password simply based on a few repetitions of the user's password. Compared to text-prompted speaker verification, this is quite different since in the latter case the system knows the prompted text (hence, i.e., its phonetic transcription) used during training and testing.

The approach studied in the present report is exploiting the advantages of hybrid HMM/ANN systems, in which an **Artificial Neural Network (ANN)** is used to estimate HMM local posterior probabilities. In this framework, ANNs have been shown to yield very good phonetic recognition rates, and this property will be exploited here to automatically infer HMM topologies.

SV-UCP raises two difficult issues. The first problem consists in finding the topology of the HMM model which better represents the password chosen by the user; this aims at capturing/modeling the **lexical content** of the password. The second problem is to quickly adapt the ANN parameters towards the targeted speaker: this aims at capturing/modeling the **speaker characteristics**.

Due to the small size of the enrolment (training) data, the most interesting approach tested here was to use a large speaker-independent ANN (in our case a multilayer perceptron, MLP) to perform the HMM inference and then to start from that inferred model to adapt the speaker independent ANN parameters to the characteristics of the user.

In this report, we describe the approach investigated so far and present the initial results. Thus, after a general description of the approach investigated here (Section 2.2), we describe in detail each of its main components, i.e., HMM inference (Section 3) and ANN parameter adaptation (Section 4), including the comparison of different MLP architectures. Section 5 then describes a set of experiments and the results obtained using the *PolyVar* database. Finally, Section 6 will draw the initial conclusions and will discuss some of the future research directions.

2.1 Databases

In the present study, we used two databases, the *PolyPhone* database [4] for training the Speaker-Independent MLP and the Swiss-French *PolyVar* database [4] to perform customer enrolment and speaker verification tests.

The PolyPhone database

The Swiss-French *PolyPhone* database 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 the 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).

The PolyVar database

For capturing intra-speaker variability, the *PolyVar* database was also designed and recorded at IDIAP as a complement to the Swiss French *PolyPhone* database, to address inter-speaker variability issues, and is particularly relevant for speaker verification research.

This database comprises telephone recordings from about 143 speakers (85 male speakers and 58 female speakers). Each speaker recorded between 1 and 229 sessions. Several speakers pronounced the same set of words several times, which makes this database particularly well suited to test user-customized speaker verification systems, i.e., by:

- Assigning each of the words to one specific customer, thus
- Providing enrolment utterances of that word, as well as 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.

2.2 General SV-UCP Approach

All the work described in the present report is thus based on a hybrid HMM/ANN system, which raises two challenging issues:

1. How to exploit the benefits of HMM/ANN systems exhibited in state-of-the-art speech recognition system in speaker verification system.
2. How to develop robust speaker verification systems based on user-customized password. Indeed this seems to still be an open problem, even with standard approaches (typically based on Gaussian Mixture Models).

The general idea of the approach developed here (see also [10]) is to use:

1. A large speaker independent ANN (with parameters Θ), in our case a large **speaker independent multilayer perceptron (SI-MLP)**, in order to estimate local posterior probabilities used in an **ergodic HMM** M to infer the best user-specific password HMM topology M_j (for a specific user S_j). Using all the enrolment utterances and the inferred model M_j, the SI-MLP parameters Θ are then adapted to yield a set of speaker dependent parameters Θ_j. The (M_j, Θ_j) set represents the final customer model, which will be used to perform speaker verification.
2. A world HMM model M, defined as an ergodic HMM, with each phoneme represented by a single state with a minimum duration constraint or with transition probabilities reflecting this duration constraint. this world model will be used to (1) infer the HMM model associated with the customer specific password, and (2) to normalize the utterance likelihood (score normalisation) for comparison with the decision threshold.

In the following, and as illustrated by **Figure 1**, we briefly summarize the main steps of the approach that has been implemented and tested so far.

1. A new customer S_j pronounces L times his/her password X_j^ℓ, $\ell = 1, ..., L$ where X_j^ℓ represents the sequence of acoustic vectors associated with the ℓ^{th} utterance.

2. Match each of the enrolment utterance X_j^ℓ with an ergodic HMM model M using the SI-MLP parameters Θ, to extract the most likely phonetic transcription for each enrolment utterance, together with its associated likelihood.

3. From each of the enrolment utterances, choose the phonetic transcription yielding the highest likelihood, and use it to build a reference user-customized HMM model M_j representing the password of client S_j (see Section 3).

4. Match each of the enrolment utterances X_j^ℓ on the speaker specific model M_j using the SI-MLP parameters Θ to yield the phonetic segmentation of all the enrolment utterances. Indeed, the adaptation of the SI-MLP parameters to the targeted speaker requires this segmentation (in order to provide target phonetic outputs).

5. Adapt the SI-MLP parameters Θ by using the above segmentation to provide the target output and by minimizing the square error between the observed output vector generated by each input vector of the enrolment utterances and the target output vector. The result is a Speaker-Dependent MLP (SD-MLP) Θ_j (see Section 4).

More precisely, the main blocks of **Figure 1** can be described as follows:

1. Feature Extraction:
   The preprocessing of the speech signal consisted of a RASTA-PLP feature calculation, resulting in 12 RASTA-PLP coefficients, complemented by their first temporal derivatives (12 $\triangle$-RASTA-PLP), as well as the first and second temporal derivatives of the log energy ($\triangle - \log - E$ and $\triangle\triangle - \log - E$, thus resulting in a total of 26 features (calculated every 10ms over 30ms windows).

2. SI-MLP (Θ):
   A good **Speaker-Independent Multi-Layer Perceptron (SI-MLP)** was previously trained on a subset of the *PolyPhone* database (described in Section 2.1), using the 10 phonetically rich sentences read by 400 speakers (200 male and 200 female speakers).

   This SI-MLP consisted of 234 input units (containing 9 consecutive acoustic frames of 26 features each), 600 hidden units and 36 outputs (associated with the 36 phones defined for PolyPhone). The output nonlinearity was the "softmax" function, ensuring that the class posterior probability outputs always sum up to one. During the experiments, different kinds of irregularities (i.e. noise in the recording, strange utterances) were discovered, and the training set was finally reduced to 3,272 sentences, corresponding to approximately 5 hours of speech. The SI-MLP training was performed using the standard error back-propagation method. The resulting SI-MLP achieved a frame-based phonetic recognition rate higher than 85% on the *PolyPhone* test data, which is particularly high and shows its good potential at performing HMM inference.

3. Posterior Probabilities Estimation:
   Using the above SI-MLP (of parameters Θ), we generate for each acoustic vector x_n of the enrolment data a set of 36 posterior probabilities $p(q_k|x_n)$, for $k = 1, \ldots, K = 36$, q_k being one of the possible phones.

4. HMM inference:
   The above posterior probabilities $p(q_k|x_n)$ are then used as emission probabilities of the ergodic HMM model M to extract the best phonetic transcription associated with each enrolment utterance X^ℓ. This inference, yielding the client-specific model M_j will be discussed in more detail in Section 3.

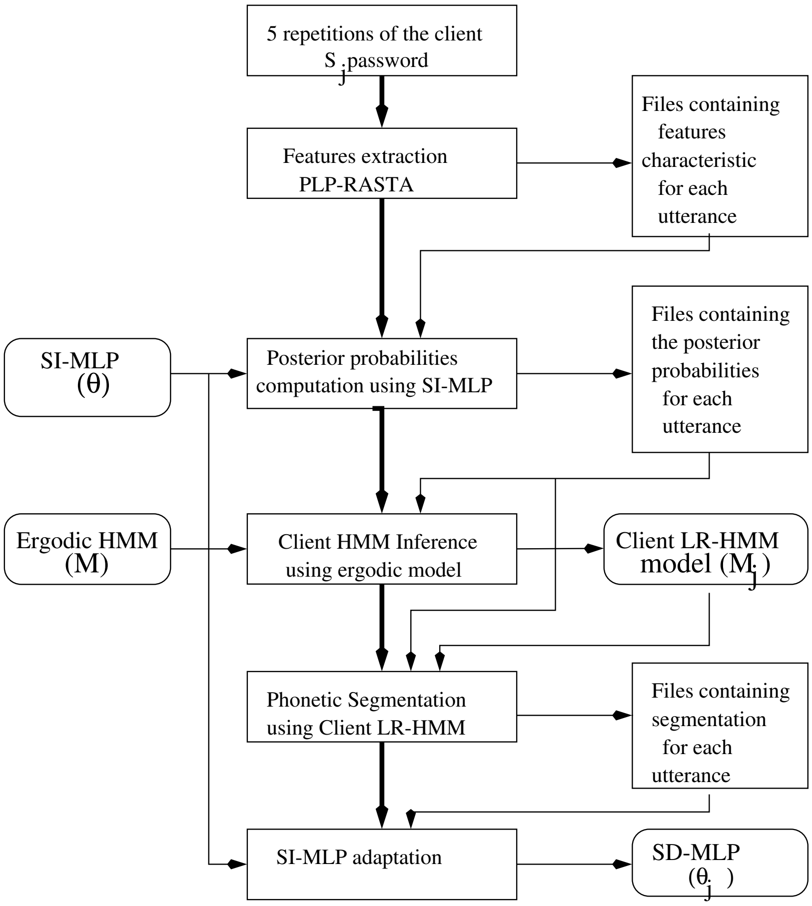

Figure 1: *Block-diagram of the SV-UCP enrolment process: 5 utterances are used to infer the topology of the user-customized password HMM (M_j), using a good speaker independent MLP (SI-MLP) previously trained on a large, speaker-independent, continuous speech, database. The resulting HMM is used to segment the enrolment utterances and to adapt the SI-MLP parameters to the targeted speaker, yielding SD-MLP (of parameters Θ_j).*

5. Phonetic segmentation using the client-specific HMM:
   Still using the SI-MLP parameters, all the enrolment utterance are aligned (by forced Viterbi) on model M_j to provide a phonetic segmentation.

6. The resulting segmentation is then used as target to adapt the SI-MLP parameters. As usually done in training hybrid HMM/MLP recognition systems, these last two steps (segmentation through forced Viterbi and MLP adaptation) could also be iterated to improve the quality of the resulting model. This adaptation, involving different MLP architectures, will be discussed in Section 4.

3 HMM Inference

The first enrolment step of a user-customized speaker verification system (SV-UCP) is to automatically infer the best HMM topology (model) from a few (five in our case) repetitions of the password and a good speaker independent set of phonetics parameters. The inferred model should be representative of the lexical content of the password. This was already discussed in [10] and consists in finding the HMM model matching best the enrolment utterances. Typically this is done by matching each of the enrolment utterance on an ergodic phonetic model, which provides us with a phonetic segmentation of each utterance. We could then merge all the resulting models into a single HMM, or simply choose the one yielding the highest likelihood as the reference model.In more detail

We took from *PolyVar* five (different) repetitions of the same word of the same speaker. For each acoustic sequence $X = \{x_1, x_2, ..., x_N\}$ 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 $p(q_k|x_n)$, for $k = 1, ..., K$ of the different phones q_k associated with the MLP outputs. Using these phone posterior probabilities, and an ergodic HMM world model M (containing the set of fully connected phonetic states, each of them being associated with a particular SI-MLP output) with minimum state duration constraints[1] and phone transition probability[2], a simple dynamic programing algorithm [1] is applied to estimate the underlying phonetic sequence.

At this point, the problem is to infer from these different phonetic transcriptions the best model M_j for the password of customer S_j. There are several ways to do this, the simplest one, which was used in this work, consisting in choosing the phonetic transcription yielding the highest global log-posterior probability $\log P(M|X, \Theta)$, and to use it to build a reference HMM model M_j representing the password of the user S_j. This HMM model then 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.

4 Speaker-Independent MLP Adaptation

4.1 Motivation

It is clear that the best way to cope with the individual speaker characteristics is to train a new MLP for each speaker (i.e., starting from a random initialization of the weights), but this method requires a large amount of training data, which is not always available. In these cases, speaker-adaptation techniques have been proposed (including for hybrid HMM/ANN systems) and will be used here. The main idea of these approaches is to start from a Speaker-Independent system and to use a limited amount of training data from a new target speaker to move the parameters of the system towards the characteristics of the speaker.

[1] Several values of minimum duration have been tested on a few words and it was observed that the "optimal" minimum duration was 3, which is consistent with what is usually observed with hybrid HMM/MLP recognition systems.

[2] Also, several values have been tested. We have observed that this probability have no effect on the topology of the model. We thus chose 0.5 as a uniform value for transition probability.

In our case, this step consisted of adapting the Speaker Independent MLP parameters Θ to the characteristics of each user (speaker) S_j, using -only- the enrolment utterances (five repetitions). The result is a Speaker Dependent MLP (SD-MLP) Θ_j for each user. Precisely, we want to be able to calculate $P(M_j|X,\Theta_j)$ from $P(M_j|X,\Theta)$.

To adapt the SI-MLP, we took the same five repetitions which are used to infer the user-specific model. The first three of them were used to adapt the SI-MLP, while the last two were used to test the generalization properties of the resulting SD-MLP. In order to perform this adaptation, it is necessary to have the phonetic segmentation associated with each utterance. In our case, it means that each frame in the acoustic utterance must be assigned to one of the phones in the infered customer HMM M_j. To yield this segmentation, we match each of the enrolment utterances on the inferred customer specific model M_j using a forced Viterbi alignment.

During MLP adaptation, the standard error back-propagation algorithm was used to minimize a least mean square error (LMSE)

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

where

- $X = \{x_1, x_2, ..., x_N\}$ is the acoustic vector sequence, associated with the adaptation utterances (three in our case), x_n representing the acoustic vector at time n and N the total number of training vectors.
- $d(x_n,\Theta)$ represents the target output vector associated with each input vector x_n and corresponding to the phonetic segmentation obtained from the SI-MLP θ.
- $g(x_n,\Theta_j)$ represents the observed MLP output vector given the current values of the parameters Θ_j:

$$g(x_n,\Theta_j) = \{g_1(x_n,\Theta_j),\ldots,g_k(x_n,\Theta_j),\ldots,g_K(x_n,\Theta_j)\} \tag{2}$$

At the end of the training, the MLP should estimate speaker-dependent phone posterior probabilities, i.e., for speaker S_j:

$$g(x_n,\Theta_j) = \{p(q_1|x_n),\ldots,p(q_k|x_n),\ldots,p(q_K|x_n)\} \tag{3}$$

4.2 Adaptation approaches tested here

Two adaptation approaches, training different sets (or subsets) of ANN parameters have been compared:

1. **Retrained Speaker Independent (RSI)**:
   In this case, we attempted to completely retrain the SI-MLP with the speaker data, adapting all the parameters to the new speaker. The advantage of this approach is that we start from a well trained SI-MLP (i.e., the weights are initialized to a good values) which is an important criterion in the adapting procedure. The inconvenience is the large number of parameters that we are attempting to adapt $(162,636)$ compared to the small size of the adapting data.

2. **Linear Input Network (LIN)**:
   This approach [11] introduces a new (trainable) Linear Input Network (LIN) to map the Speaker-Dependent (customer) input vectors to the SI-MLP. The parameters of additional linear layer were trained by minimizing the LMS error (as described above) at the output of the SI-MLP whose parameters have been frozen. The major advantage of this technique is the important reduction of the parameters to be adapted, going from 162,636 in the case of RSI down to 54,756 (or less).

As the amount of adaptation data is very limited, different LIN architectures were tested, using different connectivities between the additional linear layer and the input layer nodes of the SI-MLP.

4.3 Type of connections

In the present work, we thus used an additional input layer (LIN) to perform a (linear) mapping from the targeted speaker acoustic features to the already trained speaker-independent ANN. To update the LIN weights, we used the gradient descent Error Back-Propagation (EBP) approach, resulting in the following update equation:

$$w_{ij}^{(\tau)} = w_{ij}^{(\tau-1)} + \triangle w_{ij}^{(\tau)} \tag{4}$$

with

$$\triangle w_{ij}^{(\tau)} = -\alpha \nabla E \mid_{w^{(\tau)}} \tag{5}$$

where w_{ij} represents the weight between node i of the input layer and node j of the additional linear layer, $\nabla E \mid_{w^{(\tau)}}$ the partial derivative of the cost function (1) with respect to that weight, and α the learning rate. The type of connections used in this work are :

1. **LIN1–Fully Connected LIN**:
   As shown in Section 2, we used an MLP with 9 frames of acoustic context, with 26 nodes for each frame, thus resulting in an additional LIN layer of 234 nodes. In the **LIN1** architecture, as illustrated in **Figure 2(a)**, all possible connections of the LIN network are used, i.e., all the nodes in the LIN are connected to all nodes in the input layer of SI-MLP. Consequently, in our case, the number of weights (parameters) to be adapted is equal to (234 $\times$ 234=54756).

2. **LIN2 and LIN3–Frame-To-Frame connections**:
   As illustrated in **Figure 2(b)**, connections between the LIN input and the SI-MLP are limited to the 26 nodes of the associated frames, without inter-frame connections. This thus results in a significant reduction of the number of parameters to be adapted, now equal to ((26 $\times$ 26) $\times$ 9=6084). In this type of connections, we distinguish between two kinds of architectures:
   LIN2: where each frame in the additional layer has its own transformation matrix, and
   LIN3: where all frame transformation matrices are forced to be the same.

3. **LIN4: Node-To-Node connections**:
   As illustrated in **Figure 2(c)**, the LIN4 architecture limits the parameters to node-to-node connections where each node in the LIN is only connected to its corresponding node in the input layer of SI-MLP, resulting in a further reduction of the number of parameters. In this case, the number of parameters to be adapted is simply equal to 234.

4.4 Adaptation and Generalization Properties

During adaptation of the SI-MLP, several points have to be taken into account, including:

1. **Generalization properties**:
   To test the generalization properties of the different architectures, we used a cross-validation technique, where the adaptation set is split into a training set on which the parameters are adapted and a cross-validation set on which the generalization properties are evaluated and which can be used as stopping criterion to avoid overtraining. So the five repetitions were divided into two parts, the first three repetitions for adaptation process and the last two repetitions for testing the generalization properties. As described in [2], after each iteration (presentation of three adapting repetitions) we test the performance of the resulting SD-MLP on the cross-validation set (containing two repetitions of the password) and continue the adaptation only when the performance on the cross-validation set improves. These generalization properties were also compared to other speaker pronouncing the same customer password, to the same user pronouncing a different word and to another speaker pronouncing a different word.

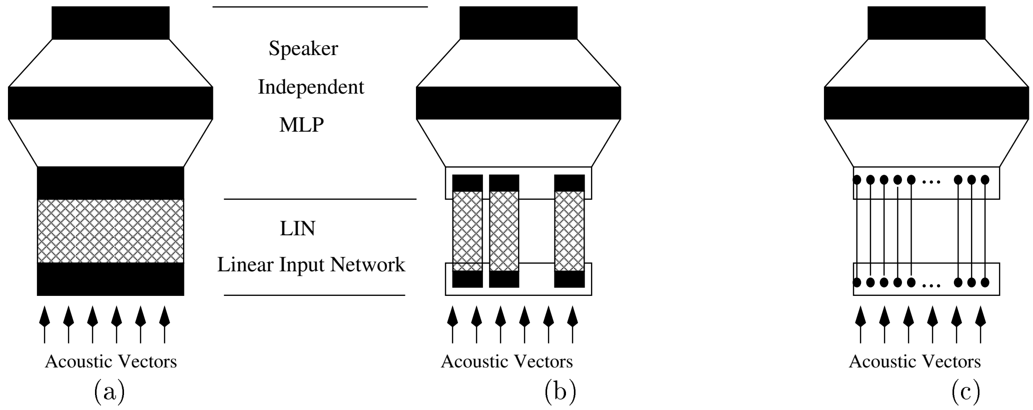

Figure 2: *Different types of LIN connections: (a) fully connected LIN, (b) frame-to-frame connections, and (c) node-to-node connections.*

2. **Learning rate**:
   As in [2], we started with a learning rate α equal to 0.1. Each time the performance on the cross-validation data degraded, we divided this learning rate by a factor 2 for the next iteration. This process was iterated until the learning rate was below 0.0001, at which point adaptation process was then considered as complete.

3. **Input Normalization**:
   This process normalizes each SI-MLP input to have zero mean and unit variance. It has been found [11] that the speaker-adaptation performance can be greatly enhanced by estimating a new input normalization transformation from the speaker-adaptation data and using that same transformation for testing. In our case, we used the normalization transformation of the SI-MLP.

4. **LIN initialization**:
   To train the LIN for a new speaker, the weights of the linear input layer are initialized to an identity matrix [11], which guarantees that the initial performance of the adapted MLP is at least equivalent to the SI-MLP.

5 Experimental Results

Our experiments were conducted on the *PolyVar* database. This database is split into three subsets, the client subset, the pseudo-impostors subset and the world subset. Our tests were performed on the client subset, which contains data from 38 speakers (24 male and 14 female speakers), **all the speakers using the same password** (in our case, the word *"annulation"*). This choice guarantees that we are working in the most difficult conditions, since we cannot discriminate among speakers using the lexical information contained in the password. Each speaker (in the client subset) recorded between 26 and 229 sessions. The first five sessions were used to infer the user HMM model and to adapt the SI-MLP. For each customer, all the other speakers were then used as impostors. For each speaker, the true accesses were between 15 and 26, and the impostors accesses were 36, amounting to 761 true speaker accesses and 1368 impostor accesses.

5.1 "Optimal" MLP architecture

In a first set of experiments, we compared the possible architectures used in the SI-MLP adaptation in order to chose the one which had the best generalization propreties. In this respect, we plotted in **Figure 3** the least mean square error variations, normalized per frame, on the adaptation and different test data, as a function of the number of MLP adapting iterations. It is better to believe here that (for each speaker) the adapting data contains 3 utterances and the cross-validation data contains 2 utterances. **Figure 3** represents only the LMS error in the case of LIN1, because the performance of the resulting SD-MLP was better. It has been found that the difference between the average LMS error on the cross-validation data and the test data of a different speaker with the true password *"annulation"* is greater in the case of LIN1 architecture than others architectures. For this, we decided -in the first time- to perform our experiments with this architecture (LIN1).

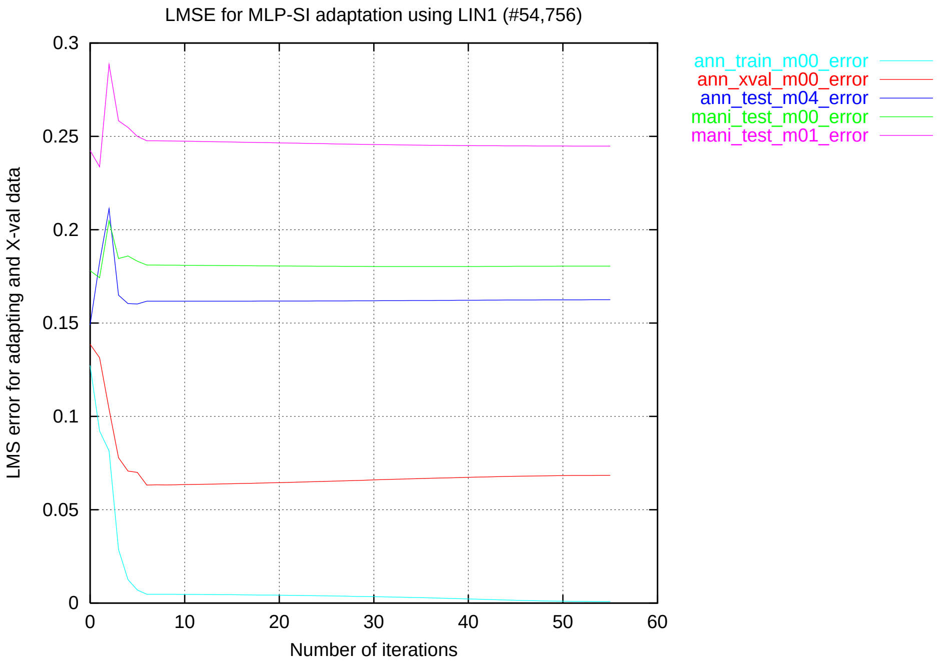

Figure 3: *LMS error normalized per frame for training and different test data with LIN1*

From **Figure 3**, we can see that the resulting average LMS error on the cross-validation data for the true speaker M00 (red curve) is better (lower) than the average LMS error on the test data for a different speaker (M04) with the same word *"annulation"* (blue curve). This difference is greater in the case of the true speaker (M00) with a different word *"manifestation"* (green curve). Moreover, the difference is even greater in the case of a different speaker (M01) with a different utterance of the word "*manifestation*" (magenta curve).

5.2 Speaker Verification Evaluation

The goal of this experiment is to evaluate the performance of the system when we use the LIN1 architecture. In speaker verification, an identity claim is made by an unknown speaker, and an utterance of the unknown speaker is compared with the model for the speaker whose identity is claimed; if the calculated score is above a certain threshold, the identity claim is verified.

In our approach , the **Verification** of a speaker S, pronouncing X and claiming to be S_j consists of the following steps:

1. Load the HMM model M_j associated with the password of S_j, the speaker specific MLP model of parameters Θ_j, the world HMM model M with its associated MLP parameters Θ.

2. Perform Viterbi matching of X on model M_j and parameters Θ_j to compute $P(X|M_j, \Theta_j)$, representing the likelihood that X was actually produced by S_j (since using Θ_j) pronouncing M_j.

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 (whence the looped world model). Another possibility, underestimating the likelihood of the world model, would be to use $P(X|M_j, \Theta)$, thus estimating the probability that the right password has been produced by a speaker different than S_j.

4. Compute the likelihood ratio and check whether or not it is above a speaker-specific threshold.

5.2.1 Scoring Criterion

In [9] a new similarity measure based on *a posteriori probability* has been proposed. The experiment showed that the *a posteriori probability* and the *likelihood ratio* measures perform equally the speaker separability. As our proposed method is based on the hybrid HMM/ANN system in which the ANN is used to estimate the *local posterior probabilities*, we want to take advantage of this and use the *a posteriori probability* as a similarity measure.
The decision to validate a speaker S claiming to be S_k, should be based on:

$$P(S = S_j) \;\; \triangle \;\; P(M_j|X, \Theta_j) \tag{6}$$

resulting in the hypothesis test:

$$S = S_j \;\; \text{if} \;\; P(M_j|X, \Theta_j) \geq \delta_j \tag{7}$$

Where $P(M_j|X, \theta_j)$ representing the accumulated posterior probability that the correct word M_j was pronounced by the true (correct) speaker (because we use the SD-MLP Θ_j to compute the local posterior probabilities).

It is also known (see, e.g., [8]) that, assuming equal class priors, this criteria is also equivalent to the likelihood ratio test:

$$\mathcal{L}_j = \frac{P(X|M_j, \Theta_j)}{P(X|M, \Theta)} \geq \Gamma_j \tag{8}$$

usually used in speaker verification, and where the denominator is often referred to as the likelihood of the “world model”.

Since different utterances will have different lengths (T), we cannot compare the score against a given (length independent) threshold directly. In the spirit of [7], the score was normalized by the duration of the test utterance, thus becoming:

$$\hat{P}_1(M_j|X, \theta_j) = \frac{1}{T_X} \;\log\; P(M_j|X, \Theta_j) \geq \delta_j \tag{9}$$

where T_X represents the number of frames in the test utterance X. This creterion, will be referred to normalized accumulated log-posteriors (**TN**).
The final accept/reject decision was then taken by comparing these resulting scores to a pre-defined threshold.

In speaker verification systems, two types of errors have to be considered:

- **False rejection error**, when an authorized customer is classified as an impostor and reject.
- **False acceptance error**, when an impostor is accepted as a valid customer.

Consequently, the performance of the system is measured in terms of equal error rate (EER), corresponding to the decision threshold where the false rejection rate is equal to the false acceptance rate. However, in real life applications, we are often more interested in the (threshold) point, where the system has the highest performance rate. For this point, we first compute the rate of the two types of errors (the false acceptance rate and the false rejection rate). Then we compute another type of error which is called HTER (Half Total Error Rate) corresponding to the mean of the false acceptance and false rejection rates. $HTER = (FA + FR)/2$. In the table 1 we present the performance of the system in term of EER and HTER, with speaker-dependent threshold (i.e., having a specific threshold associated with each speaker) and speaker-independent threshold (i.e., comparing the score of a test utterance to a global threshold common for all speakers). The first two columns (FA and FR) represent the false acceptance rate and false rejection rate only in the case of a speaker-independent threshold.

scoring criterion	FA	FR	HTER_SI	EER_SI	HTER_SD	EER_SD
S_c	22.0%	26.8%	24.4%	25.0%	14.7%	17.9%

Table 1: *Performance of the system using LIN1*

From Table 1, we can draw the following conclusions:

1. As opposed to the result obtained in the first experiment where the adapted SD-MLP discriminates well between the true speaker and the impostors, the performance of the system is not as good as expected. To better improve the performance, we used two others scoring criteria as explained below (Section 5.2.2).
2. From columns 4 and 6, we clearly see that the SV performance based on SD thresholds is better than that based on SI threshold.

5.2.2 Alternative scoring criteria

Removing the contribution of non-informative silence frames: TNS

Since silence frames do not contain any specific information about the user, but often contribute a lot to the matching score, we tried to remove the silence frames present at the beginning and the end of the test utterance. To do this, we first perform a Viterbi matching on the user-specific model M_j and parameters Θ_j, in order to yield the phonetic segmentation of the test utterance. Then we removed the frames which are considered as silence phones. Then we re-estimate the score as follows:

$$\hat{P}_2(M_j|X,\theta_j) = \left[\frac{1}{e_u^c - b_u^c + 1}\sum_{n=b_u^c}^{e_u^c} \log P(q_k^n|x_n)\right] \tag{10}$$

where q_k^n represents the particular phone associated with x_n, and b_u and e_u represent the beginning and the end of the test utterance respectively, after having removed the silence frames.

Performance of the resulting system is given in Table 2, line 2.

Scoring based on confidence measure: DN

In all previous approaches, all frames contribute in the same way to the matching score, and consequently different phones will have different contributions depending on their respective length. However, in the framework of recent developments in confidence levels [7], it was shown that the confidence of a model (quantifying how well a model matches some speech data), is better approximated by using a *double normalisation* of the score. This involves a normalisation over each phonetic segment (average score over each phonetic segment), followed by a normalisation over the number of phones. In our case, this yields the following scoring:

$$\hat{P}_3(M_j|X,\theta_j) = \left[\frac{1}{N_k} \sum_{k=1}^{N_k} \frac{1}{e_k^c - b_k^c + 1} \sum_{n=b_k^c}^{e_k^c} \log P(q_k^n|x_n) \right] \tag{11}$$

where b_k and e_k respectively represent the beginning and the end of phone q_k, and N_k is the number of phones in the test utterance.

This method thus takes into account the number of frames in each phone, as well as the number of phones in the test utterance. This method, introduced in [7] to estimate confidence levels, is usually referred to as *"double normalization"*.

Performance of the resulting system is given in Table 2. line 3.

scoring criterion	FA	FR	HTER	EER	HTER_SD
TNS	15.9%	27.0%	21.5%	22.0%	12.4%
DN	16.6%	38.5%	27.5%	28.0%	14.0%

Table 2: *System performance using LIN1 (TNS: Time normalized without silence; DN: double normalization).*

From the results presented in the above tables, we can draw the following conclusions:

1. The TNS scoring criterion perform better the others scoring criteria.

2. Remarkably, the false rejection rate is also much higher (38.5%) in the case of double normalization (i.e., many valid accesses from the correct user were considered as impostors'accesses). It has been found [13], that double normalization is a useful measure for rejecting utterances that are out of domain, or that contain out-of-vocabulary words or speech disfluencies. If we use this result with the definition[3] of the measure confidence given in [15], we can conclude that many user's model much worse the user data. To verify if this conclusion is true or false , we chose the speaker which gave the worst HTER (39.0%), and we found that the inferred HMM contains -only- the last five phones ([sil][ll][aa][ss][yy][on][sil]). So this model did not represent correctly the lexical content of the user's password which made several user accesses considered as impostors accesses. It is better to remember here, that the HMM inference algorithm was based on the choice of the phonetic transcription yielding the highest global log-posterior probability. To show the importance of the HMM inference step. we took the same speaker,and we chose (from the inferred five transcriptions) another phonetic transcription (different to the best one used in the above experiment)) and we found that the HTER -for this speaker- was greatly improved (14.0%). Furthermore, we observed that minimizing the Least Mean Square error in the output layer of the SD-MLP did not automatically imply the maximization of the global log-posterior probability on the user HMM model.

5.3 SV performance with different MLP architectures

All the results so far have been obtained by using the LIN1 architecture. In this section, we test our conclusion on the other types of architectures discussed in Section 4.3. Of course, the same training/testing databases and protocol were used. The tables below present the results obtained with these different architectures.

From these tables, we can draw the following conclusions:

[3]A confidence measure may be defined as a statistic which quantifies how well a model matches the data.

	scoring criterion	FA	FR	HTER_SI	HTER_SD
RSI	TN	17.8%	28.5%	23.2%	15.6%
	TNS	19.5%	22.5%	21.0%	13.7%
	DN	21.8%	28.0%	24.9%	15.0%

Table 3: *Performance of the system using RSI architecture with different scoring criteria*

	scoring criterion	FA	FR	HTER_SI	HTER_SD
LIN2	TN	23.2%	22.6%	22.9%	14.9%
	TNS	19.8%	22.3%	21.1%	12.3%
	DN	17.6%	35.9%	26.7%	14.6%

Table 4: *Performance of the system using LIN2 architecture with different scoring criteria*

	Scoring Criterion	FA	FR	HTER_SI	HTER_SD
LIN3	TN	20.6%	27.1%	23.8%	15.1%
	TNS	20.1%	23.2%	21.6%	13.2%
	DN	18.6%	32.2%	25.4%	15.1%

Table 5: *Performance of the system using LIN3 architecture with different scoring criteria*

	scoring criterion	FA	FR	HTER_SI	HTER_SD
LIN4	TN	22.6%	31.4%	27.0%	18.1%
	TNS	29.3%	28.1%	28.7%	18.1%
	DN	30.2%	27.3%	28.8%	18.0%

Table 6: *performance of the system using LIN4 architecture with different scoring criteria*

1. The LIN techniques (except LIN4) perform better the RSI technique. but there is no significant difference between them.
2. The best results are often achieved by LIN2, independently of the scoring criterion used to calculate the score. This is in contradiction with the good discrimination properties of LIN1 exhibited in Section 5.1, Figure 3.
3. Usually, it is better to remove non-informative silence frames of the test utterance in the score computation. This is in agreement with [12] where it was shown that for text-independent speaker recognition, it is important to remove silence frames from both the training and the testing signal.

5.4 SV performance with other HMM inference criteria

As we have seen, one problem in the SV-UCP is to infer the phonetic tanscription (from the enrolment utterances) of the user password, which is then used to build the user HMM model. In the previous experiments, we chose the phonetic transcription yielding the highest accumulated posterior probability as a user model, and we found that the infered model match worse the user test data.
Many approaches were proposed to solve the pronunciation modeling problem. In this section we tested one of them based on the choice of the phonetic transcription which better match all the enrolment utterances.
As we have done in previous experiments, we chose the best phonetic transcription to build the user HMM model. The difference lies in the criteria used to evaluate this best phonetic transcription.
The inferred procedure is as follows:

1. Find the most probable phonetic transcription of each utterance, as we did in the HMM inference, (section 3).
2. Let i=1.
3. Choose the phonetic transcription P_i.
4. Force align each of the utterances X^ℓ on P_i, to estimate $\left[-\log P(X^\ell|P_i,\theta)\right]$. $1 \leq \ell \leq L$, where L is the number of the enrolment utterances. Two criteria were tested (see below).
5. Compute the average posterior probability (APP) of all the utterances X^ℓ, aligned on P_i.
$$APP(P_i) = \frac{1}{L}\sum_{\ell=1}^{L} -\log P(X^\ell|P_i,\theta) \tag{12}$$
6. i=i+1.
7. Go to 3, until i equal to the number of the enrollemnt utterances L.
8. Choose the phonetic transcription P_ℓ with the smallest $APP(P_\ell)$ as the best phonetic transcription and use it to build the user HMM model as we have already seen in the HMM inference (section 3).

To estimate $\left[-\log P(X^\ell|P_i,\theta)\right]$ (step 4), we used two criteria which have been already used as a confidence measures in the hybrid ANN/HMM framework [7]. Those two criteria are based on **normalized posterior confidence measure (NPCM)** and defined at the word level, but they used two different kinds of duration normalization.

- *frame-based*$\mathbf{NPCM(w)}$: which can be defined as follow

$$frame-based\mathbf{NPCM}(\mathbf{w}) = \frac{1}{\sum_{j=1}^{J}(e_j - b_j + 1)} \sum_{j=1}^{J} \sum_{n=b_j}^{e_j} \log P(q_j^n | x_n^\ell) \tag{13}$$

- *phone-based*$\mathbf{NPCM(w)}$: or double normalization and defined as follow:

$$phone-based\mathbf{NPCM}(\mathbf{w}) = \frac{1}{J} \sum_{j=1}^{J} (\frac{1}{e_j - b_j + 1} \sum_{n=b_j}^{e_j} \log P(q_j^n | x_n^\ell)) \tag{14}$$

where J is the number of phone segments, and b_j and e_j are respectively the first and the last frame of phone segment q_j.

The results are presented in table 7. The test utterance score was calculated using the **TNS** scoring criterion (after removing the silence frames). The first criterion (max. posterior prob), is the one used in the previous experiments.

HMM inference criterion	FA	FR	HTER_SI	HTER_SD
max. posterior prob.	19.8%	22.5%	21.1%	12.6%
frame-based$\mathbf{NPCM(w)}$	14.8%	30.2%	22.5%	13.8%
phone-based$\mathbf{NPCM(w)}$	16.1%	24.9%	20.5%	13.6%

Table 7: *Performance of the system using different criteria to infer the user HMM model and with TNS scoring creterion*

From the results, we can conclude that

- Depending on the threshold, the max. posterior probability criterion performed better the two other criteria when we used a speaker dependent threshold. However the *phone-based*$\mathbf{NPCM(w)}$ criterion is the best one when we used a speaker independent threshold.
- The results show that no one of these criteria gives a significant improvement. It was observed that when we use one of these criteria (e.x. *phone-based*$\mathbf{NPCM(w)}$) the test utterance was rejected, while the same test utterance was accepted when we use another criterion (e.x. *frame-based*$\mathbf{NPCM(w)}$). A solution consisted of merging all the phonetic transcriptions should be represent better the user password.

6 Conclusion and Future work

In this report, we presented a method based on a hybrid ANN/HMM system for SV-UCP, where the user can choose his/her password from an unconstrained vocabulary. So, there is not any prior knowledge about the phonetic transcription of the password. Then we described a set of experiments which are performed on a subset of the PolyVar database. For a difficult task (where all the users have the same password *"annulation"*, and all the accesses have been done with the same word *"annulation"*, the system show reasonable performance, although not competitive yet with state-of-the-art text-independent speaker verification systems. The results show that it is better to remove the non-informative silence frames from the test utterance before doing the score computation and that double normalization is a useful measure to evaluate an acoustic model in the hybrid HMM/ANN system framework.

In the future, we mainly intend to improve the baseline system described in the present report by further improving the HMM inference and the SI-MLP adaptation process. The possible research directions are the following:

1. System design:
   To adapt the SI-MLP to the characteristics of the user, the target output vector (segmentation) were computed using Viterbi alignment on the user specific HMM model. So, during the adaptation process, the neural network outputs were biased toward phones which constitute the user model M_j. Due to the variation in the signal, the speaker can not repeat an utterance precisely the same way from trial to trial, and, in this case, the system is not able to recognize some correct (pronounced by the true user) utterances (perhaps this is one reason that the false rejection rate is high). It is better to adapt the SI-MLP to recognize all phones in the user enrolment utterances. One way to do this, is to compute the segmentation (which is then used to adapt the SI-MLP as described in Section 4) by using Viterbi alignment on the ergodic model. We then use the resulting SD-MLP to compute the posterior probabilities which are further used to infer the speaker specific HMM (as described in Section 3). It is expected that the resulting SD-MLP will exhibit a strong bias toward phones frequently used by the targeted speaker.

2. HMM inference:
   As we have seen, the HMM inference is an important step in the process of the construction of the user model. The goal of this step (as mentioned in Section 3) is to automatically infer the best HMM topology. However, the definition of the best topology is not clear, depending whether the goal is to find the model with the highest likelihood or the model that corresponds best to the correct phonetic transcription of the word. In reality, the user cannot pronounce the same word several time in the same way, thus, the inferred model is a good model if it can properly model the pronunciation variations of the same word by the same speaker. Various approaches, inspired from current work in pronunciation modeling for speech recognition systems will have to be investigated here. One solution consists in merging several phonetic transcriptions using a dynamic programming algorithm as in [14].

3. SI-MLP adaptation:
   In the adaptation procedures tested so far, all the output units were used for training the weights to be adapted (i.e., back-propagation and gradients were computed using the error between the target and observed output over all possible ANN phonetic outputs. However, in the enrolment data, only a few phones are present (those constituting the customer password HMM model). In [3], a method was presented and consisted in adapting only the parameters (weights) from the hidden units to the targeted outputs (phones) (the outputs which correspond to phones in the user HMM model).

References

[1] H. Bourlard, Y. Kamp, H. Ney, and C. J. Wellekens. "Speaker-Dependent Connected Speech Recognition via Dynamic Programing and Statistical Methods". In *Speech and Speaker Recognition*, volume 12, pages 115–148. Karger, Basel, 1985.

[2] H. Bourlard and N. Morgan. *"Connectionist Speech Recognition: A hybrid approach"*. Kluwer Academic Publisher, 1994.

[3] D. C. Burnett. *"Rapid Speaker Adaptation for Neural Network Speech Recognizers"*. PhD thesis, Oregon Graduate Institute of Science, 1997.

[4] G. Chollet, J.-L. Cochard, A. Constantinescu, C. Jaboulet, and P. Langlais. "Swiss French Polyphone and Polyvar: telephone speech databases to model inter- and intra-speaker variability". IDIAP-RR 01, IDIAP, 1996.

[5] J. deVeth and H. Bourlard. "Comparison of Hidden Markov Model Techniques for Automatic Speaker Verification in real-world Conditions". *Speech Communication*, 17:81–90, 1995.

[6] S. Furui. "An Overview of Speaker Recognition Technology". In *ESCA workshop on Automatic speaker Recognition, Identification and Verification*, pages 1–9, 1994.

[7] G.Bernardis and H. Bourlard. "Improving Posterior based Confidence Measures in Hybrid HMM/ANN Speech Recognition Systems". In *Proc. ICSLP'98*, volume 3, pages 775–779, 1998.

[8] B. Gold and N. Morgan. *"Speech and Audio Processing"*. Wiley, 2000.

[9] T. Matsui and S. Furui. "Similarity Normalization method for Speaker Verification based on a Posteriori Probability". In *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, pages 59–62, 1994.

[10] B. Nedic and H. Bourlard. "Recent Developments in Speaker Verification at IDIAP". IDIAP-RR 26, IDIAP, 2000.

[11] J. Neto, L. Almeida, M. hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson. "Speaker-Adaptation for Hybrid HMM/ANN Continous Speech Recognition System". In *EUROSPEECH'95*, pages 2171–2174, 1995.

[12] D. A. Reynolds. "Speaker Identification and Verification using Gaussian Mixture Speaker Models". In *ESCA Workshop on Automatic Speaker Recogniton, Identification and Verification*, pages 27–30, 1994.

[13] Z. Rivlin, M. Cohen, V. Abrash, and T. Chung. "A Phone-Dependent Confidence Measure for Utterance Rejection. In *ICASSP'96*, volume 1, pages 515–517, Atlanta, 1996.

[14] M. G. Thomason and E. Granum. "Dynamic Programing Inference of Markov Networks from Finite Sets of Sample Strings". *IEEE transaction on Pattern Analysis and Machine Intelligence*, 8(4):491–501, 1986.

[15] G. Williams and S. Renals. "Confidence Measures for Hybrid HMM/ANN Speech Recognition". In *EUROSPEECH'97*, pages 1955–1958, Rhodes, Greece, 1997.