Signal Pre-Pocessing and Speech Quality Validation XM2VTSDB Testing

J.Jordá, E. Bailly-Balière, B. Ruiz, A. García Crespo O. Monterrubio
Departamento de Informática
Universidad Carlos III de Madrid
Av. Universidad 30, 28911 Leganés, Madrid, Spain
{ }

ABSTRACT
In this paper we give a detailed description of the progress achieved in the techniques for feature extraction, signal preprocessing, parametric representation and robustness against noise necessary to implement a speaker verification system useful in a real environment. We also present the completly experimentation environment and the results obtained.

KEY WORDS
Signal Preprocessing, Speech Validation, Feature extraction

1 Introduction

In this document we present the work done by the Universidad Carlos III de Madrid within the Signal Preprocessing under the supervision of the *BANCA project*[1]. A testing platform has been developed in order to test the software developed and work with different databases.

In the algorithm section we show how the developed softwafe (Automatic Feature Extractor) works and which steps follows.

After that, we present the results obtained with the experimentation of the Automatic Feature Extractor developed at Universidad Carlos III de Madrid and the XM2VTS[2] data base.

2 Experimentation platform

The defined experimentation platform has three main blocks: the database, the experimentation protocol and the algorithm. The databases used are multimodal which includes voice and image in order to perform context independent verification using audio signals and dynamic images. After that process, the results will be fusioned. In the present document we show the results obtained within voice processing. The mentioned components will be explained in the following sections.

2.1 Data Bases

The databases used have been recorded trying to represent the prototype's operating environments. Two voice data bases have been used. The first one is the well known XM2VTS and the second one is a new data base developed at the *BANCA project*. Following we present a brief description of each one.

2.1.1 XM2VTS

XM2VTS is the M2VTS[3] successor and also offers much more experimentation data. It follows he subsequent configuration: Clients (Training, test: for performance statistics purposes, Evaluation: for system parameters' tunning), Impostors (Evaluation and Test) and World (male and female). Each user will have four sessions with two samples per session. The contents are:

- Item 1: 0 1 2 3 4 5 6 7 8 9
- Item 2: 5 0 6 9 2 8 1 3 7 4
- Item 3: 'Joe took fathers green shoe bench out'

2.1.2 BancaDB

BancaDB database has been recorded in several european countries usign 5 different languages. There are 260 speakers, splitted in 5 groups with 52 persons per group (each group will use a language: English, French, German, Italian and Spanish). The data used for the spanish session has been recorded at the Universidad Carlos III de Madrid, including clients from several communities. The *BancaDB* database is a multisession database, useful to study time effects and how can the model training be improved with the training models

[1] BANCA:, European Union Project IST-1999-11159

[2] XM2VTS: European Project ACTS-102 Multimodal verification

[3] M2VTS: Multimodal project ACTS-102, Multimodal verification

with different sessions recordings. Each group has 26 male and 26 female user and there are 12 sessions per user. The time between two sessions was between 4 and 8 days. Each group of 26 is divided into two of 13 and each client is faked by each of the other 12 group members (one per session).

2.2 Experimentation Protocol

2.2.1 XM2VTS

For the experimentation with the XM2VTS database we have followed the sequent configurations defined in [3].

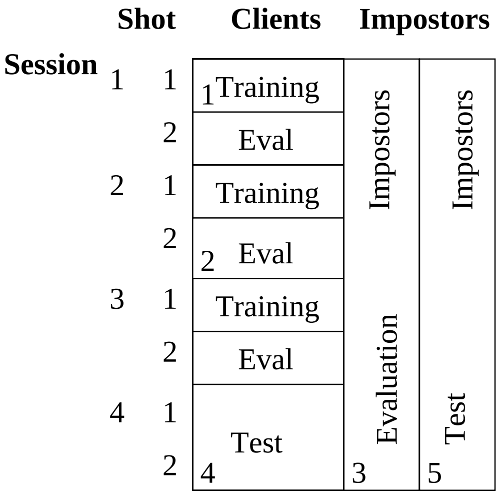

Figure 1: XM2VTS configuration 1

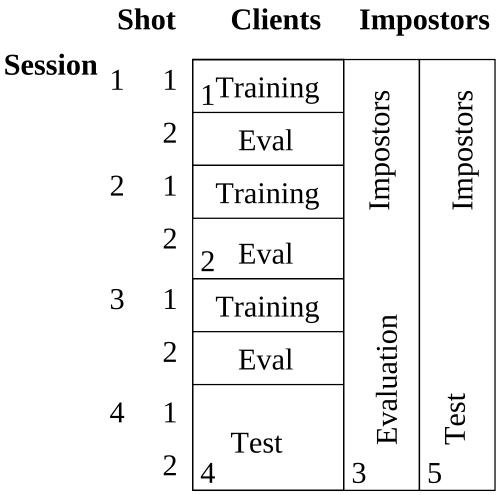

Figure 2: XM2VTS configuration 2

2.2.2 BancaDB

Seven different training-test configuration sets must be considered for the *BancaDB* database [2].

- Matched controlled
- Matched degraded
- Matched adverse
- Unmatched degraded
- Unmatched adverse
- Pooled test
- Grand test

The subsequent table shows the $BancaDB$ configurations.

Scenario c **(Controlled)**	**Session** **Records**	1 T , I	2 T , I	3 T , I	4 T, I
Scenario d **(Degraded)**	**Session** **Records**	5 T , I	6 T , I	7 T , I	8 T, I
Scenario a **(Adverse)**	**Session** **Records**	9 T , I	10 T , I	11 T , I	12 T, I

For each user (in both development and test sets), the following sequence of sessions/records are avaiable (a 'T' record is a genuine client acces, while an 'I' record is an impostor access).

2.3 System perfomance measures

In order to carry out the system evaluation process, the following values will be calculated:

- False acceptance rate (FAr): Users wrongly accepted.
- False rejection rate (FRr): Users wrongly non accepted (they should have been accepted).
- HTERR: Shows the system's efficiency. It's calculated as the FA and FR pondered average.

$$HTERR = \frac{FA\,\alpha + FR\,(1-\alpha)}{2} \qquad (1)$$

where α is a value between 0 and 1.

In order to visualize the results a DET curve graph is used which plots the FRR (False rejection rate) as a function of the FAR (False acceptance rate).

3 Algorithm

The features extracted will help to discriminate among the speakers or verify a claimed identity. In order to do that we use a GMM (Gaussian Mixeture Model) based algorithm. The system has two different parts:

- Training: Several kinds of voice samples are given to the system in order to generate a model for each client.
- Verification: This is the function performed when the system is in operating mode. The system checks the probability of producing the incoming speech utterance given the claimed client model against the probability of the speech utterance being produced by a world model.

3.1 Architecture

Feature extractor is a part of a complete speaker verification system which is shown in figure 3 as a block diagram.

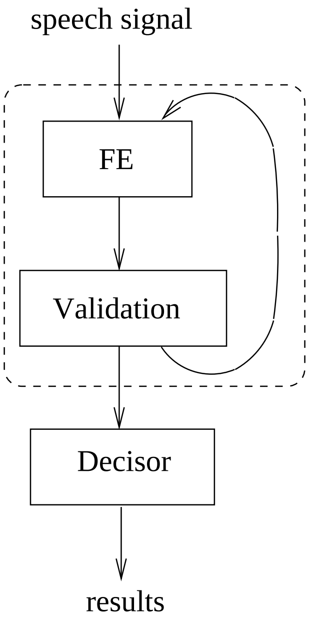

Figure 3: System's flowchart

The input is a speech signal in raw format. Currently we represent the wave using a file.

The implemented $fetureextractor$ is represented by the $FE's$ box. For the experimentation it has been used another extra software too: HTK. Despite of HTK is a good software several problems appeared while the testing was done: It's a generic software and very complex, which implies a high computational time cost and it needs lot of time to configurate all the parameters for a simple small test. These problems encouraged us to develop our own FE much more simple, time efficient and also real time execution.

3.2 FE: Details

The system will require a 8 Khz sampled signal and it will process it aplying several algorithms, explained in the sequents paragraphs.

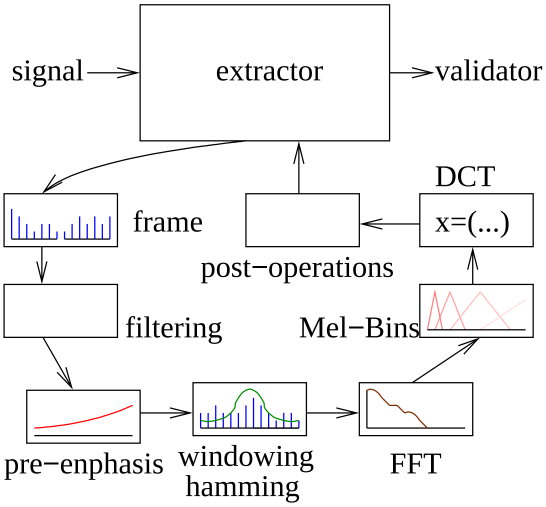

Figure 4: FE flow chart

3.2.1 Frame

As we are working in real time, complete or large speech signals can't be manipulated by our system so the incoming signal will be fragmented into more manageable signals (frames). We call this process *Framing* and we'll use 256 sample's frames. There is a special particularity in the contruction method of the frames. We follow the next process:

- For the 1^{st} frame: $f_1 := read(S, 256);$
- For the i^{th} frame, $(i > 1)$:
  $f_i := f_{i-1}[128 - 256] * read(S, 128);$

With this mechanism a more smoothed signal is obtained without cuts or discontinuity.

3.2.2 Filtering: CMM

A noise substraction technique[**?**] has been implemented using a window based algorithm.

For each sample we calculate

$$\bar{y} = y\frac{1}{T}\sum_{t=1}^{T} y(t) \tag{2}$$

where $\bar{y}$ is the noise added signal.

3.2.3 Pre-emphasis

This filter will enhance the higher frequencies to reduce the large difference between lower and higher frequencies inherent to vocal tract excitation. The effect of this is a reduction in the spectral dynamic range and an improvement in the following numeric processing.

3.2.4 Windowing Hamming

The frame is multiplied by a window function that smoothens the effects of splitting the speech signal into frames.

3.2.5 FFT

The Fast Fourier Transform is calculated in order to have an easier calculation of the Mel Bins.

3.2.6 Mel Bins

To calculate the MFCCs, a fixed number of triangular bandpass filters with large overlapping are used.

3.2.7 DCT

The *Discret Fourier Transformation* is calculated for continuingthe post operations.

3.2.8 *Post* Operations

In this section we normalise the system energy using a window based algorithm.

3.3 Validation and Decision

The second diagram ($validation$) will take care of the validations. Once the individual features are obtained we have to compare with the real individual model of the speaker stored in the platform. The iteration means that the system may need more frames to process and it will be sent by the FE. The ASV[4] software was used in order to simulate the validation procedure as it has a low computational time cost and a flexible parameter configuration despite it has more results obtention orientation than time efficiency orientation. The $decisor$ will decide if the results obtained can be considered as good.

4 Results

In this section we present some of the obtained results using the XM2VTS with the FE developed at the UC3M and some of the work done with the BancaDB.

4.1 Experimentation with XM2VTS

Four experiments have been tested with four different configurations presented in the following table

Protocol	Cep	Bin	Size	Gauss	EVA	DEV
I	12	16	256	64	3.58	4.53
II	12	16	256	64	5.211	5.042

where

- Cep: the number of Cepstral coefficients to calculate,
- Bin: the number of filter Bins used in the calculation,
- $Size$: Frame size,
- $Gauss$: Number of gaussians for the validator
- EVA: Results obtained in the system tunning phase.
- $TEST$: Results with the system tunned ”a priori”.

For the results we have also used a posteriori energy normalisation and CMM filtering.

[4] ASV: Autimatic Speech Validation, developed at IDIAP