THE CAVE-WP4 GENERIC SPEAKER VERIFICATION SYSTEM

Cédric JABOULET [2,5] *Johan KOOLWAAIJ* [4] *Johan LINDBERG* [3]
Jean-Benoît PIERROT [1] *Frédéric BIMBOT* [1]

(1) ENST / CNRS (2) IDIAP (3) KTH (4) KUN (5) Ubilab-UBS

cedric.jaboulet@ubs.com koolwaaij@let.kun.nl lindberg@speech.kth.se
pierrot@sig.enst.fr bimbot@sig.enst.fr

http://www.PTT-Telecom.nl/cave

1. CONTEXT

The CAVE project (CAller VErification in Banking and Telecommunications) was a 2 year project supported by the Language Engineering Sector of the Telematics Applications Programme of the European Union, and for the Swiss partners, by the Office Fédéral de l'Education et de la Science (Bundesamt für Bildung und Wissenschaft). The partners were Dutch PTT Telecom, KUN, KTH, ENST, UBILAB, IDIAP, VOCALIS, TELIA and SWISSCOM. The CAVE project terminated on November 30th, 1997.

This papers describes the generic speaker verification system that was developed within the *Research and Technology improvements for Speaker Verification* Work package, to focus on research and development aspects of speaker verification.

2. BASICS

The CAVE-WP4 generic speaker verification system is a software system developed to simulate different speaker verification algorithms. These approaches can be Left-Right or Ergodic HMMs, Vector Quantization, Gaussian Mixture Modeling and some kind of Dynamic Time Warping under certain constraints. Text Dependent, Text Prompted and Text Independent strategies can be tested with the generic system.

3. GOAL

The aim of the generic system is to provide a common framework for speaker verification experimentation. Thus the system has to be flexible in such a way that it allows many configurations of the algorithms, but also portability across different hardware platforms. Thanks to these features, each improvement made in one research lab was easily transfered to the other sites, allowing fast progresses of the speaker verification performances.

[1]ENST - Dépt Signal, CNRS - URA 820, 46 Rue Barrault, 75634 Paris cedex 13, FRANCE-EU

[2]IDIAP, Rue du Simplon 4, Case Postale 592, CH-1920 Martigny, SWITZERLAND

[3]KTH, Department of Speech, Music and Hearing, Drottning Kristinas Väg 31, S-100 44 Stockholm, SWEDEN-EU

[4]KUN, Dept of Language & Speech, Erasmusplein 1, NL-6525 HT Nijmegen, THE NETHERLANDS-EU

[5]Ubilab, Union Bank of Switzerland, Bahnhofstrasse 45, CH-8021, Zürich, SWITZERLAND

4. STRUCTURE OF THE GENERIC SYSTEM

The system is based on two elements : a large set of shell scripts/programs to automate all steps of the experimentations, and the HTK (Hidden Markov Modeling Toolkit, version 2.x)[1] as core of the speaker verification engine to simulate the various algorithms. These two distinct elements were chosen due to their availability across most Unix platforms : without any changes, the generic system was extensively used on Hewlett Packard, Silicon Graphics, Digital and Sun workstations. Only very small numerical differences were noticed between these platforms. They were due to different processor architectures, but they never influenced the results of our experiments.

In addition to that software structure, a common formalism that we describe later was developed to codify all the experiments : this provides a completely unambiguous definition of each experiment, ensuring an easy deployment of the technology improvements, but also reducing potential risks for mistakes. The generic system was kept as flexible as possible, so that moving to a new test database requires only very few changes.

An experiment can be briefly described by the following statements. Each potential speaker (client) is modeled by a set of HMM models trained on his own speech data. A World Model is used for likelihood normalization. The verification process is performed using a two-step alignment, one using the 'Claimed Identity' speaker models, and a second using the World models. This provides a log likelihood ratio that can be compared to a decision threshold for acceptance or rejection. More details are given in a companion paper[2].

5. COMMON FORMALISM TO DESCRIBE SPEAKER VERIFICATION EXPERIMENTS

The following points describe all the characteristics of an experiments.

5.1. Database

This provides different information on the used database (language, available training and test material specific to this database, list of speakers, ...)

5.2. Algorithms

This defines whether the experiment is performed with Left-Right or Ergodic HMMs, Vector Quantization,Gaussian Mixture Modeling, Dynamic Time Warping.

5.3. Enrollment material

For each speaker, databases generally provide data for one or more training sessions. This element defines one particular configuration (number of training sessions, and amount of speech available per session) between all possibilities for the current database.

5.4. Speech material used for the World modeling

Since the currently available Speaker Verification databases are quite small and do not contain a large number of speakers, the World models are generally trained on other databases, commonly available for Automatic Speech Recognition.

5.5. Mode for the verification

This defines if the generic system will be used in Text Dependent, Text Prompted or Text Independent mode.

5.6. Size of the HMM models

Each word/sub-word unit, of each registered speaker will be modeled as a simple HMM. The size (number of states) of the HMM chain can be fixed for all HMMs, or can be relative to the length of each word/sub-word (eg. number of states per phoneme). The number of Gaussian mixtures for each state has also to be defined. Additionally, the generic system can deal with a different model structure/size for the HMMs of the World model.

5.7. Structure of the Covariance matrices

Depending on the available enrollment material and the enrollment strategy, the covariance matrices of the HMMs can be diagonal or full, fixed or learnable.

5.8. Parametrisation - Acoustic analysis

This defines the front-end to be computed from the speech samples. It includes the kind of parameters (FTT- or LPC-derived cepstrum coefficients), sampling frequency, frame rate, frame shift, windowing, pre-emphasis on the signal, vector size, ... All combinations defined in the built-in HTK signal processing libraries are allowed. Additionally, external user defined front-ends are possible, as soon as they are supplied into the standard HTK binary format.

5.9. Computation of the likelihood

There are several ways of computing the log likelihood of each verification attempt. We mainly use a forced alignment strategy, knowing the acoustic content of the speech utterance, since it does not require a segmentation of the speech material.

5.10. Flooring for variance estimation

As described in [2], it has proven to be very efficient to use adaptive flooring when estimating some covariances matrices of the Gaussian distributions. Parameters have to be fixed for this adaptive flooring[6].

6. SOFTWARE ARCHITECTURE

As mentioned earlier, the core of the speaker verification engine is based on HTK. The generic system was originally built for HTK 2.0. when newer versions (2.0.2 and 2.1) were available, we adapted our system to take advantage of the bug fixes and the new HTK features. Around this package, a lot of shell-scripts/programs allow the following tasks.

- Preparation of a new database : each time a new database is used, tasks common to many experiments have to be run (preparation of the enrollment data in a suitable format, definition of the verification protocol, ...)
- Setup of the generic system for each new parametrisation : computation of a flooring vector for the Variance Flooring strategy, training of silence/non-speech HMM model, ...
- Enrollment: training of HMM models for the registered speakers and creation of the World models.
- Management of all untrainable models
- Speaker Verification for all attempts defined in the verification protocol
- Decision threshold computing (for a-priori/a-posteriori strategies)[4]
- Scoring of the verification performance : Equal Error Rate, False Acceptance Rate, False Rejection Rate

Since the parameters of each Speaker Verification experiment were clearly defined, we designed a Graphical User Interface, based on Tcl/Tk. This interface aims at integrating all parameters and all tasks in a user friendly application. It is based on a client/server architecture. There is only one client which is the graphical interface (see Figure 1). Its role is to define all the parameters of a given task/experiment, and to run it through a process server. Thus, there are several process servers, distributed over a network of workstations. This allows to very easily distribute the computations across many workstations when running experiments in parallel.

7. PROCESSING OF AN EXPERIMENT

To give a clear picture of the generic system, we describe hereafter all the steps of a speaker verification experiment. All theoretical justifications are described in [2].

7.1. Enrollment

The enrollment of the speakers/clients is made by the following procedure :

[6] A patent describing this strategy is currently under review

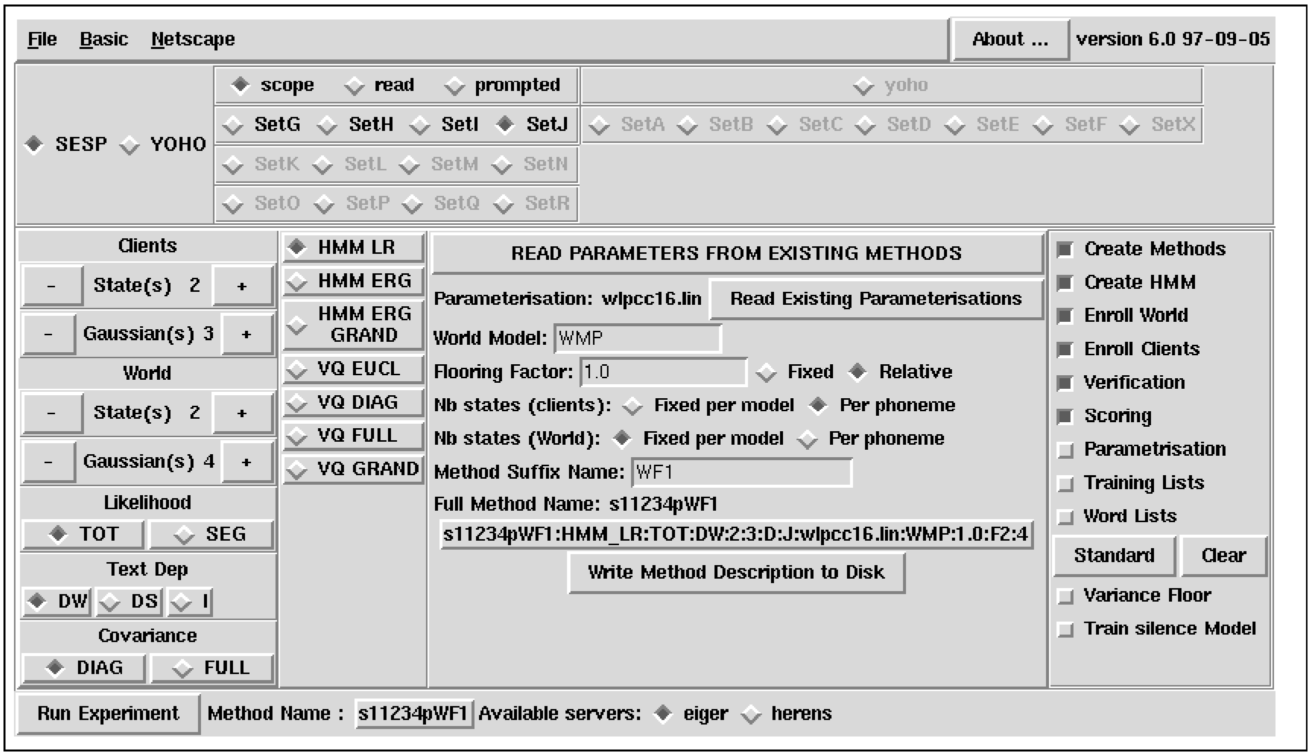

Figure 1. Client part of the GUI

```
For each Speaker
  For each Word/Sub-word in the available vocabulary
    Initialize HMM model from scratch,
    using speaker training data.
    Use the same training data in a more robust
    re-estimation process.
  End
End
```

During each iteration of the training process, the variance flooring strategy is used, to prevent over-fitting of the variance.

The procedure for the training of the World models is very similar. The only difference is the availability of a big amount of speech data when training the World models.

A post-enrollment procedure was implemented, to deal with untrainable models for the clients. Sometimes, due to the lack of available enrollment material, the generic system is not able to create a model for a given word. In that case, the model is replaced by the model of the same word, from the world model. This approach is only viable if a limited number of words are untrainable.

7.2. Access

The verification process is performed by doing two forced alignments. The first uses the *claimed identity* speaker models and the verification speech utterance. The second one is made with the World models. These two alignments provide a *client log likelihood* and a *world log likelihood*. These two values will be used to compute the *log likelihood ratio*.

7.3. Scoring procedure - Threshold setting

Most of the time, following the EAGLES recommendations [5] we evaluate the performance of the experiments in term of Gender-Balanced Sex-Independent Equal Error Rate. Thus, we perform an a posteriori threshold setting, to find the Equal Error Rate for each registered speaker, using both the genuine and the impostors verification attempts. The scoring procedure is then computing the Gender-Balanced Sex-Independent Equal Error Rate.

The CAVE project also studied a priori threshold setting procedures. A comparison of different methods is described in [4].

8. NEEDED RESOURCES

As we intend to make the generic system available to the research community, we give hereafter a list of resources needed to perform speaker verification research using the CAVE-WP4 generic system.

- the CAVE-WP4 generic system: it will provided as a set of tools including shell-scripts and programs.
- A standard Unix development environment: shell scripts, C compiler, ...
- The HTK package (version 2.1 or later).
- A database for speaker verification. Good examples are the SESP[7] or the YOHO[6] databases.
- A database to train the World models. The Polyphone-like databases are suitable for this task.

[7] property of PTT-Telecom

9. CONCLUSION

The strategy we adopted to design the CAVE generic speaker verification system has proven to be very efficient to share the knowledge between all the CAVE partners. We were able to conduct more than 400 speaker verification experiments, and we obtained very good results on real life telephone speech databases. Theses results are described in [2] and [3].
As mentioned earlier, we intend to make the CAVE generic speaker verification system available to the research community. Details are given on the CAVE web page :
`http://www.PTT-Telecom.nl/cave`

REFERENCES

[1] S. YOUNG, J. ODELL, D. OLLASON, V. Valtchev, P. WOODLAND *The HTK BOOK*, HTK 2.1 Manual. 1997.

[2] F. BIMBOT, H-P. HUTTER, C. JABOULET, J. KOOLWAAIJ, J. LINDBERG, J-B. PIERROT : *An overview of the CAVE project research activities in Speaker Verification*, Submitted to RLA2C workshop, Avignon, 1998.

[3] F. BIMBOT, H-P. HUTTER, C. JABOULET, J. KOOLWAAIJ, J. LINDBERG, J-B. PIERROT : *Speaker Verification in the Telephone Network : research activities in the CAVE project*, Eurospeech-97, v.2 p971-974, Rhodes, 1997.

[4] J.-B. PIERROT, J. LINDBERG, J. KOOLWAAIJ, H.-P. HUTTER, D. GENOUD, M. BLOMBERG, F. BIMBOT : *A comparison of a priori threshold setting procedures for speaker verification in the CAVE project*. Submitted to RLA2C workshop, Avignon, 1998.

[5] F. BIMBOT, G. CHOLLET, *Assessment of speaker verifica-
tion systems.*, In Spoken Language Resources and Assessment EAGLES Handbook. 1995.

[6] CAMPBELLF *Testing with the YOHO CD-ROM Voice Verification Corpus*, Proc. ICASSP 95, vol. 1, pp.341-344, Detroit, 1995