ALIZE, A FREE TOOLKIT FOR SPEAKER RECOGNITION

Jean-François Bonastre, Frédéric Wils

LIA/CNRS, Université d'Avignon
Agroparc, BP 1228
84911 Avignon CEDEX 9, France
jean-francois.bonastre@lia.univ-avignon.fr
frederic.wils@lia.univ-avignon.fr

Sylvain Meignier

LIUM/CNRS, Université du Maine
Avenue Laennec
72085 LE MANS CEDEX 9, France
sylvain.meignier@univ-lemans.fr

ABSTRACT

This paper presents the ALIZE free speaker recognition toolkit. ALIZE is designed and developed in the framework of the ALIZE project, a part of the French research Ministry Technolangue program. The paper focuses on the innovative aspects of ALIZE and illustrates them by some examples. An experimental validation of the toolkit during the NIST 2004 Speaker Recognition Evaluation campaign is also proposed.

1. INTRODUCTION

The ELISA consortium [5] groups several speaker recognition research teams (mainly from France). The main objective of ELISA is to share the engineering effort needed for participating in international evaluation campaigns, like NIST Speaker Recognition Evaluation (NIST SRE) campaigns. In this sense, the ELISA consortium has participated in NIST SRE since 1998 [1][2]. During this time, the consortium has developed and maintained a common software, shared along the different partners. The different tools cover speaker detection tasks as well as speaker segmentation ones. The common platform was updated regularly in order to integrate the different proposals and to match the state-of-the-art.

In order to save and to spread the ELISA knowledge, the consortium decided to rewrite all the software following strong software engineering methods and to distribute it freely (it also facilitates the user support and the software updating). This effort is sponsored by the French Research Ministry, in the framework of ALIZE project (formally AGILE/ALIZE), a part of Technolangue program [4].

This paper firstly presents ALIZE project. The section 3 illustrates some characteristics of the ALIZE software. Section 4 presents an experimental validation of ALIZE in the framework of NIST SRE 2004 evaluation. Finally, the last section proposes some comments and future directions.

2. ALIZE PROJECT

The ALIZE project [6] groups several partners. Firstly, the LIA is responsible for both the project management and the toolkit development. Secondly, several academic research teams - mainly former partners of ELISA consortium (IRISA/INRIA, CLIPS/GEOD, Dep. TSI ENST, DDL and IRIT) constitute the scientific guaranty of the project. They have participated in different international evaluation campaigns before and during the project. They validate the software performance and they propose to add new up-to-date functionalities. Thirdly, three companies are also ALIZE partners (Thales Communication, ATLOG and CALISTEL). They have in charge the industrial aspects of the project (to crosscheck the software on different hardware, to verify the quality regarding the software engineering constraints, to propose some specific functionalities or constraints for building industrial demonstrators). Lastly, some sponsors (like the ETCA/DGA and the COST275 project) and non French ELISA partners complete the ALIZE partner list.

The main objectives of ALIZE are:

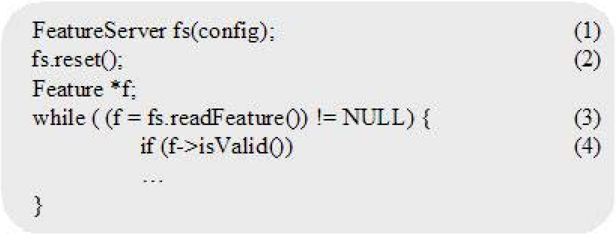
- to propose a toolkit making faster the development of new ideas, with a (proved) state-of-the-art level of performance;
- to encourage the laboratories to evaluate new proposals using the toolkit and both standard databases and protocols like NIST SRE ones;
- to help the understanding of speaker recognition algorithms (like EM training, MAP adaptation or Viterbi algorithm), parameters and limits;
- to test new commercial applications;
- to facilitate the exchanges and the knowledge transfer between the academic labs and between academic labs and companies.

3. AN ILLUSTRATION OF ALIZE SPECIFICITIES

ALIZE is developed in C++ following an object oriented UML method. The general architecture of the toolkit is based on a split of the functionalities between several software servers. The main servers are the feature server, which manages the acoustic data, the mixture server which deals with the models (storage, modification, tying of components, saving/reading...), and the statistic server which implements all the statistical computations (EM-based statistic estimations, likelihood computation, viterbi alignment, etc.)

This architecture presents several advantages:

The work was done at the LIA with the support of the French Research Ministry Technolangue program



```
FeatureServer fs(config);                              (1)
fs.reset();                                            (2)
Feature *f;
while ( (f = fs.readFeature()) != NULL) {              (3)
            if (f->isValid())                          (4)
            ...
}
```

Fig. 1. *Acoustic data management in ALIZE. (1) Server init., (2) Server reset, (3) Reading call in the reading loop, (4) Verify if a frame is available*

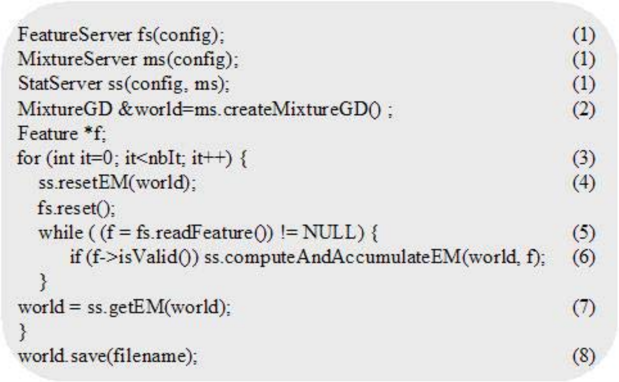
- Each server is accessible thanks to a very short list of high level functions and the low level functions like memory management are usually masked to the user;
- Each server is optimized and updated separately (thanks to this relative independence, developing new functionalities inside a new server is also very easy);
- Several instances of the same server could be launched at the same time (particularly useful for multi-streams or multimedia systems);
- The user-code presents the same structure, organized between the main servers, helping the source code development and understanding;
- Finally, the software architecture allows easily to distribute the servers on different computers: several instances of the same server could be launched on different computers for increasing the computational power (these functionalities are included in the design but are not yet available).

3.1. Data reading and synchronizing

All the acoustic data management is delegated to the feature server. It relies on a two step procedure: feature server initialization and a reading call for accessing to each feature vector. The reading of an acoustic stream is done frame by frame following a feature loop as shown in figure 1. The user synchronizes the reading process thanks to the reading calls. Therefore, the same source code is employed for off-line (file based) processing and for on-line (open microphone) processing (the only difference occurs in the server configuration). Accessing at different time instants in the audio stream (go backward or forward) is also allowed. The memory management is done using a simple rule: the user defines the size of the data buffer, in time, from unlimited to one frame. If the buffer shows a limited size, while the user requests a frame out of the buffer, the server will return a ”frame non available” message (the same message is sent if no frame is available during an open microphone mode).

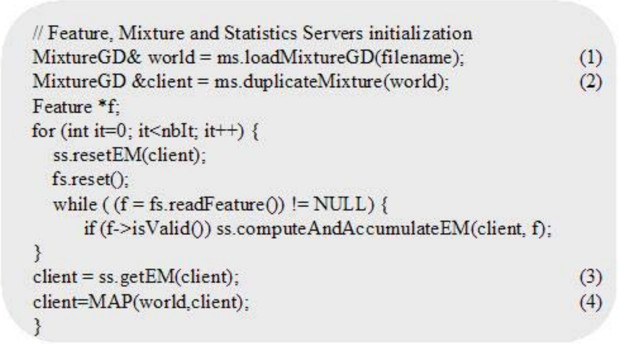
3.2. EM/ML world model training

For training a GMM world model, the user has to initialize three servers, the feature server for the data, the mixture server for managing the Gaussian components (and mixture models) and the statistic server for estimating the statistics. Figure 2 shows the skeleton of an example of EM training, beginning from scratch (the model is randomly initialized by default but the initialization could be easily controlled).



```
FeatureServer fs(config);                                            (1)
MixtureServer ms(config);                                            (1)
StatServer ss(config, ms);                                           (1)
MixtureGD &world=ms.createMixtureGD() ;                              (2)
Feature *f;
for (int it=0; it<nbIt; it++) {                                      (3)
   ss.resetEM(world);                                                (4)
   fs.reset();
   while ( (f = fs.readFeature()) != NULL) {                         (5)
       if (f->isValid()) ss.computeAndAccumulateEM(world, f);        (6)
   }
world = ss.getEM(world);                                             (7)
}
world.save(filename);                                                (8)
```

Fig. 2. *EM based world model training. (1) Servers init., (2) Init. the world model, (3) EM it. loop, (4) Reset the stat. accumulator, (5) Feature reading loop, (6) Accumulate the stat., (7) Get the stat and copy it in the world model, (8) Save the model*

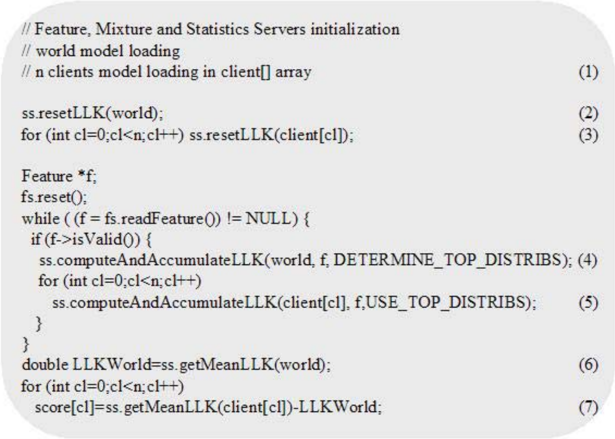


```
// Feature, Mixture and Statistics Servers initialization
MixtureGD& world = ms.loadMixtureGD(filename);                       (1)
MixtureGD &client = ms.duplicateMixture(world);                      (2)
Feature *f;
for (int it=0; it<nbIt; it++) {
   ss.resetEM(client);
   fs.reset();
   while ( (f = fs.readFeature()) != NULL) {
       if (f->isValid()) ss.computeAndAccumulateEM(client, f);
}
client = ss.getEM(client);                                           (3)
client=MAP(world,client);                                            (4)
}
```

Fig. 3. *Client model estimation by MAP algorithm. (1) Load the world model, (2) Create the client model by world duplication, (3) Get the stat on the client training data, (4) Estimate the resulting model as a function between a priori knowledge (world) and the current stat, copy it in client model*

3.3. EM/MAP speaker model estimation

For deriving a speaker model from the world model using a MAP adaptation algorithm, the user builds exactly the same program as the previous one. Only two differences have to be highlighted: the client model is initialized as a copy of the world model and the final model (at each iteration) is the result of MAP(), a function involving both the world model (the a priori knowledge) and the statistic estimated on the client training data. Implementing some variants of MAP algorithm will only take place in this MAP() function. This process is illustrated figure 3.



```
// Feature, Mixture and Statistics Servers initialization
// world model loading
// n clients model loading in client[] array                                  (1)

ss.resetLLK(world);                                                           (2)
for (int cl=0;cl<n;cl++) ss.resetLLK(client[cl]);                             (3)

Feature *f;
fs.reset();
while ( (f = fs.readFeature()) != NULL) {
  if (f->isValid()) {
    ss.computeAndAccumulateLLK(world, f, DETERMINE_TOP_DISTRIBS); (4)
    for (int cl=0;cl<n;cl++)
      ss.computeAndAccumulateLLK(client[cl], f,USE_TOP_DISTRIBS);             (5)
  }
}
double LLKWorld=ss.getMeanLLK(world);                                         (6)
for (int cl=0;cl<n;cl++)
  score[cl]=ss.getMeanLLK(client[cl])-LLKWorld;                               (7)
```

Fig. 4. *LLR score computation with n top gaussian computing. (1) Load the models, (2) and (3) Reset the LLK accumulators for the world model and for each client model, (4) For a given frame and using the world model, determine the top gaussian, memorize the individual component likelihood and compute/accumulate the world likelihood, (5) For the same frame, using the top components and the memorized values, compute and accumulate the likelihood for each client, (6) and (7) Get the mean log likelihood and compute the LLR per client.*

3.4. Score computation

The score computation follows the same structure as the two previous programs. Figure 4 shows an example of the (log) likelihood computation for several client models, using a n top gaussian computing. For top gaussian computing, the user needs only to set an optional flag (DETERMINE_TOP_DISTRIBS) during the corresponding call for memorizing the winning components and to set the flag to a different value (USES_TOP_DISTRIBS) for using its for the other calls. An implicit component tying is also implemented and helps to save computational time and memory.

4. EXPERIMENTAL VALIDATION DURING NIST04

In order to evaluate the ALIZE toolkit, the LIA built a new system based on it. This system, referenced in this paper as LIA04, was presented during the NIST 2004 evaluation campaign and compared to the previous ELISA system.

4.1. The LIA04 system

LIA04 is a GMM/UBM based system [9]. The world model training, the speaker model estimation and the score computation are close to the examples explained in 3. The system is more detailed in [8]. The front-end processing relies on a LFCC analysis. The signal is characterized by 16 linear frequency cepstral coefficients (LFCC) and their first derivative coefficients (a 300-3400Hz bandwidth filtering is applied). As for ELISA systems, the LFCC computation is performed using the (free) SPRO software [3]. A frame removal process based on multi-gaussian modeling of the energy is used in order to remove non-informative frames. Finally, the parameter vectors are normalized to fit a 0-mean and 1-variance distribution. The world models are 2048 component gender dependent GMM with diagonal covariance matrices (an experiment with 128 component models is also proposed). The target models follow the same structure and are derived from the corresponding world model using a MAP Adaptation (only means of each gaussian are adapted) [9]. The score computation relies on a log-likelihood ratio computed on the n top components (n=10).

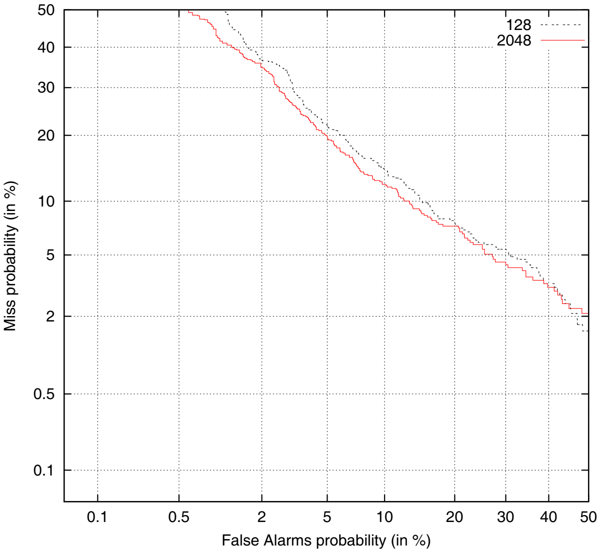


Fig. 5. *Influence of model size on SRE'04 (ztnorm).*

4.2. Experimental protocol

The experiments reported in this paper were done in the framework of NIST SRE 2004 evaluation campaign and followed the official NIST protocol. The different DET curves proposed in the next section correspond to the primary 1side-1side condition (see the NIST SRE'04 evaluation plan for more details on the different proposed tasks [10]). The target speaker set is composed of 370 female and 246 male speakers with a training segment duration of about five minutes of conversation (giving 2,5 minutes of speech from the speaker). The number of test trials is 26224 and the test duration is close to the train segment duration. For the world model training and the score normalization (Znorm, Tnorm, ZTnorm) [7], we used some data issued from previous years NIST SRE databases.

4.3. Results

Figure 5 shows the performance of LIA04 (ALIZE based) system during NIST04 evaluation. The results are provided for two different model sizes, 128 and 2048 components.

As noticed in 1, ALIZE is a project issued from the ELISA consortium. The next experiment aims at comparing the performance of the LIA 2004 system based on ALIZE toolkit with last ELISA system, ELISA02 (based on LIA AMIRAL software) [2].

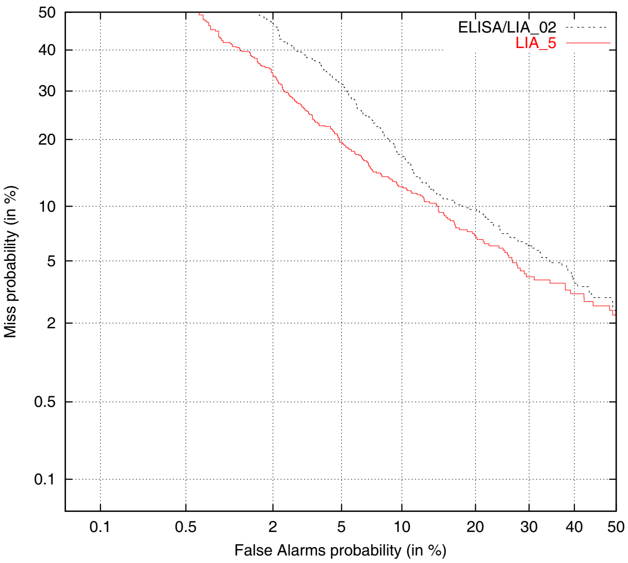


Fig. 6. *LIA 2004 ALIZE system versus ELISA/LIA AMIRAL 2002 system on SRE'04 (using ztnorm).*

We used the best configuration for each of the systems (128 component models for ELISA 02/AMIRAL, 2048 component models for LIA04/ALIZE). The experimental protocol is the NIST SRE 2004 one (described in the previous section) and the same data sets were used for training the world models and for the score normalization. The parameterization and score normalization are common for both systems (using a ZTnorm score normalization). This comparison is illustrated on figure 6. A significative gain can be observed, especially around the operating points defined by the NIST protocol.

During the NIST SRE 2004 evaluation, the LIA primary system (LIA04_5) obtained a good performance level, one of the best for acoustic only systems (high level information are not used).

5. CONCLUSION

The ALIZE project was launched by the ELISA consortium in order to save and to distribute the consortium knowledge. The ALIZE project wants to encourage private and public labs to take part in speaker recognition research and development and particularly to international evaluation campaigns like NIST campaigns. In order to achieve these objective, the ALIZE project proposes a free open source software (GNU/LGPL licence), the ALIZE toolkit. The toolkit follows strong software engineering rules, is easy to understand and easy to use. It is also maintained and an user support is provided.

LIA04, a system developed by the LIA using ALIZE, was evaluated during NIST SRE 2004 campaigns. LIA04 system outperforms the ELISA02 system (the previous ELISA consortium common software) and achieved good results especially compared to equivalent acoustic only systems (LIA04 doesn't take advantage of high level information). These results demonstrate that ALIZE reaches the state-of-the-art. The LIA04 system is also available freely, following a GNU/GPL licence and could be used as a reference system.

ALIZE also includes HMM/Viterbi functionalities in order to cover text-dependent speaker recognition as well as speech/speaker segmentation tasks.

One of the hope of ALIZE project is to create a large community of ALIZE developers. This last point is mandatory for enforcing the quality of the software and for maintaining the performance level of the toolkit. ALIZE project was launched thanks to the French Research Ministry Technolangue project. The future of the project will now depend on the ALIZE developer community and sponsors are welcomed.

6. REFERENCES

[1] The ELISA Consortium, "The ELISA Systems for the NIST'99 Evaluation in Speaker Detection and Tracking", *Digital Signal Processing*, Vol. 10, No. 1-3, pp. 143-153, January/April/July 2000.

[2] The ELISA consortium, "Overview of the ELISA consortium research activities", *2001 a speaker Odyssey: the speaker recognition workshop*, Chania (Crete), June 2001.

[3] http://www.irisa.fr/metiss/guig/software.html.

[4] http://www.technolangue.net/

[5] http://elisa.ddl.ish-lyon.cnrs.fr/

[6] ALIZE project web site, http://www.lia.univ-avignon.fr/heberges/ALIZE/

[7] R. Auckenthaler, M. Carey, H. Lloyd-Thomas, "Score normalization for text-independent speaker verification system", *Digital Signal Processing (DSP), a review journal - Special issue on NIST 1999 speaker recognition workshop*, vol. 10(1-3), 2000.

[8] J.F. Bonastre, N. Scheffer, C. Fredouille, D. Matrouf, "NIST04 speaker recognition evaluation campaign: new LIA speaker detection platform based on ALIZE toolkit", *proceeding of NIST 2004 speaker recognition workshop*, 2004.

[9] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, "Speaker verification using adapted Gaussian mixture models", *Digital Signal Processing (DSP), a review journal Special issue on NIST 1999 speaker recognition workshop*, vol. 10(1-3), pp 19-41, 2000.

[10] The evaluation plan of NIST 2004 Speaker Recognition evaluation campaign. http://www.nist.gov/speech/tests/spk/2004/SRE-04_evalplan-v1a.pdf.