The GTH-LID System for the Albayzin LRE12 Evaluation

Luis Fernando D'Haro, Ricardo Córdoba

Speech Technology Group – Dpto. de Ing. Electrónica – E.T.S.I. de Telecomunicación - Universidad Politécnica de Madrid. Ciudad Universitaria s/n 28040, Madrid, Spain
{lfdharo,cordoba}@die.upm.es

Abstract.
This paper contains a description of the data-sets, systems and fusion alternatives developed by the Speech Technology Group (GTH) for the Albayzin 2012 Language Recognition Evaluation for the 4 conditions: plenty-closed condition (core), plenty-open, empty-closed, and empty-open. In all cases, the primary system is the fusion of three different i-vector based systems: one acoustic system, a phonotactic system using trigrams of phone-posteriorgram counts, and another acoustic system based on RPLP features instead of the traditional MFCC features. For each plenty condition, a contrastive system was also included where the RPLP features or MFCC features where replaced by a different system based on using glottal source features. We provide results for the plenty conditions using the proposed metrics for the evaluation (i.e. Fact, Fdis, and Fcal), as well the known Cavg metric used on NIST evaluations.

Keywords: Language Recognition, Phonotactic system, iVectors, RPLP, GlottHMM

1 Introduction

The goal of this paper is to describe the GTH system for the Albayzin 2012 LRE task. Our primary submission includes three systems:

1. Acoustic system based on MFCC + SDC features and RASTA, iVectors
2. Phonotactic system based on trigram Posteriorgram Counts, iVectors
3. Acoustic system based on RPLP + SDC features and RASTA, iVectors

Fig. 1 shows a block diagram of the submitted system. Detailed information about each system, as well as the calibration and fusion process will be provided throughout the paper. We also submitted a contrastive system for the plenty conditions based on using glottal source features (called GlottHMM-iVector, see section 6). In this case, we replaced system 3 by a system based on using glottal source information and iVectors.

As many of current state-of-the-art systems, our three systems make extensive use of sub-space projections in the form of iVectors [1] combining different, but complementary, kind of information.

The paper is organized as follows: Section 2 describes the data-sets used for training, development and test prior to the final evaluation. Section 2.2 explains the acoustic system, section 4 the phonotactic system, section 5 describes the RPLP system, and section 6 the Glottal source based system. Finally, section 7 covers the fusion and calibration results.

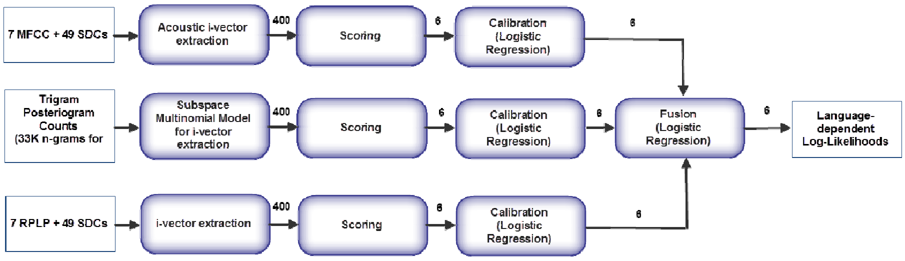

7 MFCC + 49 SDCs
Acoustic i-vector extraction
400
Scoring
6
Calibration (Logistic Regression)
6
Trigram Posteriogram Counts (33K n-grams for
Subspace Multinomial Model for i-vector extraction
400
Scoring
6
Calibration (Logistic Regression)
6
Fusion (Logistic Regression)
6
Language-dependent Log-Likelihoods
7 RPLP + 49 SDCs
i-vector extraction
400
Scoring
6
Calibration (Logistic Regression)
6

Fig. 1. Block diagram of the primary submitted system.

2 Data description

2.1 Plenty conditions

Table 1 shows the statistics of the number of files used in our setup for training, development and test in both plenty conditions. For the three systems reported in this paper we have always used the same file sets.

Plenty	Closed			Open		
	Train	Dev.	Test	Train	Dev.	Test
No. Files	4656	458	457	5265	725	725
No. Langs	6			7		
No. of clean files	3060	N.A	N.A	3060	N.A	N.A
No. of Noisy files	1596	N.A	N.A	1596	N.A	N.A

Table 1. Statistics for the training, development and test set for the plenty conditions

We have divided the original development set into two subsets with a similar language distribution. We did not apply any k-fold strategy. The first one is the "Dev." set used to calibrate the system, and the second one is the "Test" set, which we have used to obtain the results presented in the paper.

For the final evaluation, we have added the "Test" set to the Train set to have more training data, calibrating the system with the "Dev." set.

2.2 Empty conditions

Table 2 shows the statistics of the number of files used in our setup for training, development and test in both empty conditions.

Empty	Closed			Open		
	Train	Dev.	Test	Train	Dev.	Test
No. Files	-	304	305	-	571	571
Our experiments	7400 (1)	304	305	10141 (2)	571	571
No. Langs	4			5		

Table 2. Statistics for the training, development and test set for the empty conditions

As we did not have any training data for the 4 new languages, we have reused the training set from the plenty conditions and merged them with the development data available duplicated three times to give it more relevance. We did not apply any adaptation technique to the models from the plenty conditions.

In detail, for (1) we have merged the data from the plenty closed (PC) training data with the PC Development data and 3 times the empty closed (EC) Development data, giving a total of 7400 training examples. In the same way, for (2), we have merged the data from plenty open (PO) training data with the PO Development data and 3 times the empty open (EO) Development data, giving a total of 10141 training examples.

As for the plenty conditions, for the final evaluation we have also added the “Test” set to the Train set to have more training data, calibrating the system with the “Dev.” set. For training the logistic regression classifier for the EC condition, we have unified the “Dev.” and “Test” sets but the calibration was done only on the “Dev.” Set. For the EO condition we have used 10141 files for training the LR classifier and the calibration was done on the 571 files.

3 Acoustic system based on MFCC + SDC features and RASTA, iVectors

In this section, we provide a brief summary of the acoustic feature extraction and UBM training used for training our acoustic system.

The first step in our system is to extract the feature vectors from variable duration segments of recorded speech. In order to do this, we first parameterize each file using SPRO5[1] extracting 12 MFCC coefficients (including C0) from 24 Mel filter banks plus the energy for each frame. Finally, Cepstral mean and variance normalization is applied to the feature vectors for each file.

[1] https://gforge.inria.fr/projects/spro

The Voice Activity Detector (VAD) used for all the systems is the output from the the BUT Hungarian phone recognizer[2]. Then, we suppressed all segments marked as silence or noise in the output. After discarding the silence segments, every 10 ms speech frame was mapped to a 56-dimensional feature vector. The feature vector is the concatenation of SDC features [3] using the common 7-1-3-7 configuration and stacking them with the first 7 MFCC coefficients out from the 12 MFCCs. Finally, a RASTA filter was applied in order to reduce short-term noise variations in each frequency subband. No Vocal-Tract Length Normalization (VTLN) was applied.

Then, we train a language-independent GMM, a.k.a universal background model (UBM), through five iterations of the EM-algorithm and using all the acoustic feature vectors coming from all the 6 (or 7 for the open condition) languages that appeared in the training set.

3.1 Acoustic iVectors

Currently one of the main techniques used for speaker recognition [2] and language recognition [4][5] is the iVectors technique. In this framework, the language and channel-dependent GMM supervector M can be modeled as:

$$M = m + Tw \tag{1}$$

Where m is the UBM GMM mean supervector, T is the total variability matrix (i.e. the iVector extractor) and w is a standard normal distributed vector of size M (i.e. iVector). The main advantage of w is that it maps most of the relevant information from the variable-length audio file to a fixed-length and small dimensional vector.

Finally, the iVectors are normalized by first subtracting the mean of all the training iVectors and then dividing them by its corresponding norm.

3.2 Results

Table 3 shows the results for the systems presented.

Condition	System	Closed		Open	
		Fact	**Cavgx100**	**Fact**	**Cavgx100**
Plenty	300iv, 512 Gauss.	0.176646	9.08	0.202484	10.54
	400iv, 512 Gauss.	0.172519	9.25	-	-
	400iv, 1024Gauss.	0.172484	9.19	0.196182	10.37
Empty	400iv, 64 Gauss.	0.075818	0.43	0.092105	3.32
	400iv, 128 Gauss.	0.092308	0.83	0.115086	4.24
	400iv, 256 Gauss.	0.109475	1.57	0.149377	5.70
	400iv, 512 Gauss.	0.120738	2.10	-	-

Table 3. Reported results for the acoustic MFCC system on the test set

[2] http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context

As we can see, in the plenty conditions, 400 iVectors provide better results than 300, and there is little difference between 512 and 1024 Gaussians, so we will use the 512 Gaussians, as it is faster and probably more robust for unseen test examples.

For the empty conditions, as there is little data, best results are obtained for 64 Gaussians, so this is the system used in the evaluation.

4 Phonotactic system based on trigram Posteriorgram Counts, iVectors

4.1 System description

In this case we have used a novel approach to phonotactic LID reported in [10], where instead of using soft-counts based on phoneme lattices, we use posteriorgrams to obtain n-gram counts. In this approach, the high-dimensional vectors of counts are reduced to low-dimensional units for which we adapted the commonly used technique iVectors. The reduction is based on multinomial subspace modeling and is designed to work in the total-variability space. In comparison with the other techniques based on soft-counts, the new technique provides better results, reduces the problems due to sparse counts, and avoids the process of using pruning techniques when creating the lattices. Previous reported results for the NIST 2009 LRE data-set showed better results compared to a system based on using soft-counts, and with very good results when fused with an acoustic i-vector LID system. For this reason, we decided to use this technique in this evaluation and check its behavior on a different database. Next, we briefly describe the proposed technique.

Feature extraction.

Fig. 2 shows the process of creating the vector of posteriorgram-based n-gram counts. In the figure, we consider the bigram counts for simplicity, but in our system we used trigrams. The process can be divided into four main steps:

1. The first step is to tokenize speech by the means of running a phone-recognizer that, for each frame, provides the posterior probabilities of the phone occurrences. In our experiments, we used the BUT Hungarian phone recognizer.
2. The second step is to sum up and average the posterior probabilities for the frames that are considered to be within the same phoneme unit (A, B, C in the Figure). The phone boundaries are obtained by running Viterbi decoding on the posteriorgram. The averaged posterior provides a good de-correlation and smoothness for the resulting matrix that we call *averaged posteriorgram*.
3. The third step is to create the *joint-posteriorgram* – a sequence of matrices of joint probabilities for the *n* consecutive frames. Here, we take the averaged posteriorgram of each frame and we do the outer product with the posteriorgram of the previous frame. Then, the process is repeated for all the phone-grams considering the *n*-1 phone-gram history.

4. The final step is to sum up all frames (matrices) of the joint-posteriorgram. This way, we create a matrix of *n*-gram counts that is converted into a 1xD vector (where D is the total number of possible *n*-grams, in the case of trigrams is 35937, 33^3, 33 phonemes and order 3) and then used as a feature file for training the iVectors using Subspace Multinomial Models.

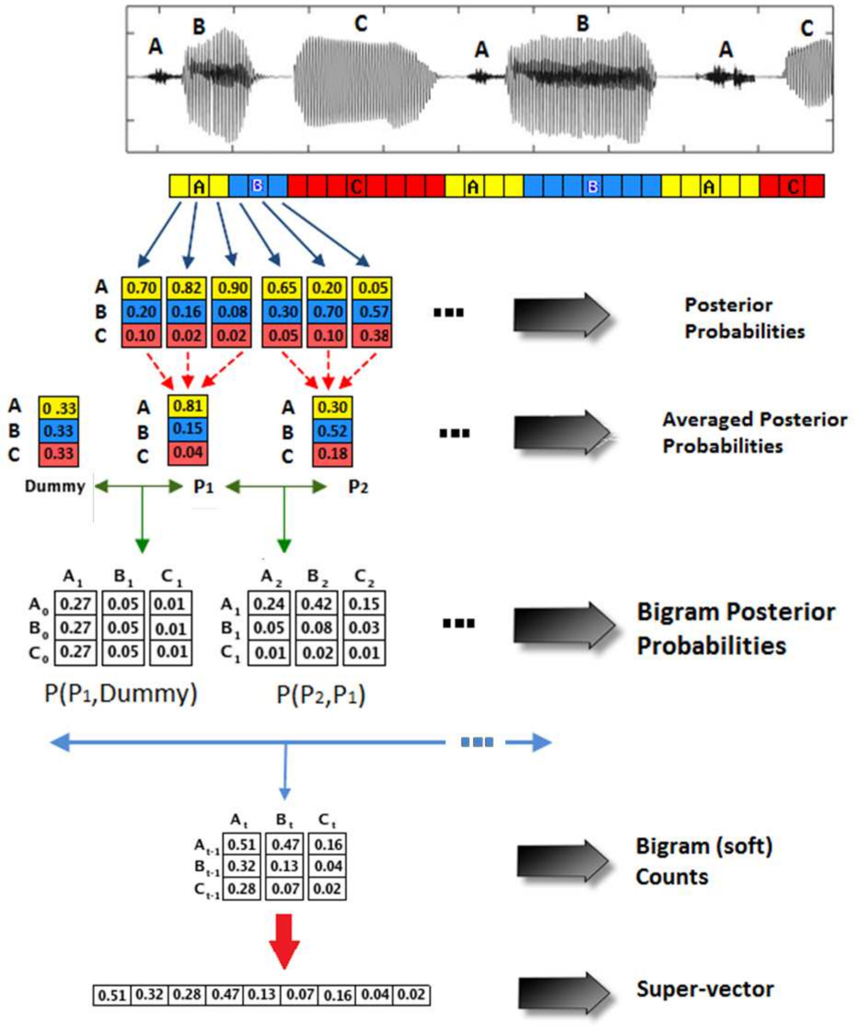


Fig. 2. Procedure to generate posteriorgram counts features

Subspace Multinomial Models.

The goal of the Subspace Multinomial Model is to model the discrete representation of the posteriorgram counts created in the previous step. Thanks to the Subspace Multinomial Models we can train low dimensional vectors of coordinates in the total variability subspace, i.e. iVectors, and then use these iVectors as a feature vector input for training a discriminative LID classifier. Next, we briefly describe the process for training the subspace multinomial models. For further details please refer to [11] and [12].

The log-likelihood of data D for a multinomial model with C discrete classes is determined by model parameters φ and sufficient statistics γ, representing the occupation counts of classes for all N utterances in D:

$$\log p(D) = \sum_{n=1}^{N} \sum_{c=1}^{C} \gamma_{nc} \log \varphi_{nc} \qquad (2)$$

Where γ_{nc} is the occupation count for class c and utterance n and φ_{nc} are probabilities of (utterance dependent) multinomial distribution, defined by a subspace model according:

$$\varphi_{nc} = \frac{\exp(\mathbf{m}_c + \mathbf{t}_c \mathbf{w}_n)}{\sum_i^C \exp(\mathbf{m}_i + \mathbf{t}_i \mathbf{w}_n)} \qquad (3)$$

Where t_c is the c-th row of subspace matrix T and w_n is an r dimensional column vector (i-vector) representing language and channel of utterance n.

For training the iVectors, we have followed the algorithm reported in [11] with slight modifications in order to iterate several times the estimation and maximization steps (further details can be found in [10]). Finally, in our experiments, we have considered a set of 1089 multinomial models when using trigrams (i.e. considering all the possible number of bigram histories, 33x33, using 33 phones for the Hungarian recognizer).

Finally, it is important to mention that for the empty condition we have implemented the following algorithm. First, we obtain the T matrix by using the created training set described in section 2.2 together with the development set and applying two epochs and two iterations for the EM iVectors extraction process. Then, the new T matrix is used to extract the final iVectors for all sets.

4.2 Results

Condition	**System**	**Closed**		**Open**	
		Fact	Cavg x100	Fact	Cavg x100
Plenty	400iv_Trigrams	0.138718	9.43	0.163411	10.37
Empty	400iv_Trigrams	0.037714	0.17	0.047180	2.40

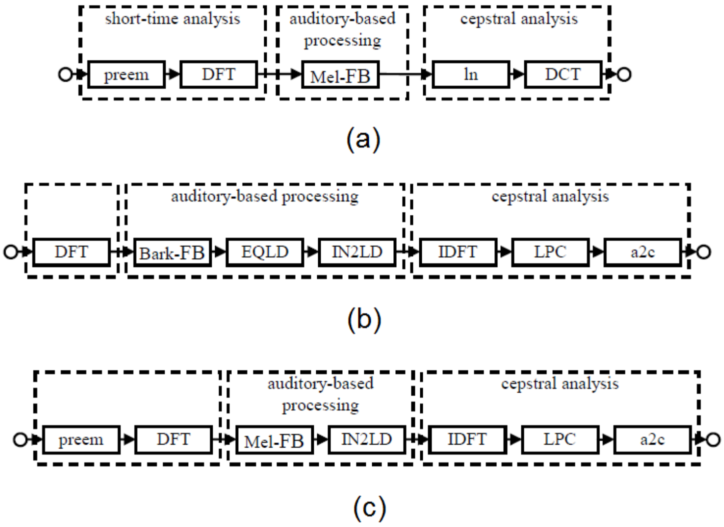
Table 4. Reported results for the phonotactic system on the test set

5 Acoustic system based on RPLP + SDC features and RASTA, iVectors

5.1 System description

The goal of this system was to introduce a new set of features which could be more robust to noise. In this case, we decided to use the RPLP (Revised PLP) features used in [13] and proposed in [14]. These features can be seen as a hybrid approach between MFCC and PLP, combining the advantages of both. **Fig. 3** shows the modules needed to calculate the traditional features (MFCC and PLP) compared to these new RPLP. As we can see, the procedure to calculate them is very similar to the MFCC computation but the DCT-based transformation is replaced by an auto-regressive (AR) modeling with additional decreasing of spectral dynamics using the INtensity-TO-LouDness (IN2LD) factor (i.e. the power-law) introduced during the PLP calculation. In [13] good improvements were found for ASR recognition in comparison with the standard features. One important contribution of this method is that it performs a double suppression of spectral dynamics before calculating the cepstral coefficients (LPC), and with less effect on the accuracy when modifying the number of FB bands, shape, and non-linearity scaling.

Finally, we apply a RASTA filter to these coefficients and then we calculate the SDC parameters for the first 7 RPLP, obtaining a final vector of dimension 56, using the common 7-1-3-7 configuration.



(a)

(b)

(c)

Fig. 3. Module sequence for calculating (a) MFCC features, (b) PLP features, and (c) RPLP features

5.2 Results

Table 5 shows the results for the systems presented.

Condition		Closed		Open	
		Fact	Cavgx100	Fact	Cavgx100
Plenty	400iv, 512 Gauss.	0.159279	7.62	0.173536	10.20
	400iv, 1024Gauss.	0.156287	7.60	0.172224	10.27
Empty	400iv, 64 Gauss.	0.040305	0.099	0.054902	1.84
	400iv, 128 Gauss.	0.041599	0.099	0.053973	1.39
	400iv, 256 Gauss.	0.038545	0.047	0.050978	1.27

Table 5. Reported results for the acoustic RPLP system on the test set

As we can see, in the plenty conditions, there is little difference between 512 and 1024 Gaussians, so we will use the 512 Gaussians, as it is faster and probably more robust for unseen test examples. It is important to see that results are better than those obtained with MFCC (see **Table 3**). For example, for the Plenty-closed condition the relative improvement is 6.42%.

For the empty conditions, we have found much better results than using MFCC (see Section 3), and we could even use 256 Gaussians.

6 GlottHMM-iVector: Glottal source based system

6.1 System description

The goal of this system was to check the viability of using glottal source features for language recognition based on the good results reported by [6] on speaking style identification in the Ircam database, and by [7] for classifying expressive speech. In the former, the use of only prosodic information (i.e pitch and rhythm) provided identification rates of about 74%; for the later, a 95% for styled speech and 82% for emotional speech on a different database.

GlottHMM [8] is a vocoding technique that was recently developed for parametric speech synthesis. It is based on decomposing speech into the glottal source and vocal tract through glottal inverse filtering. In our system we have used GlottHMM to extract the F0 and the Harmonics to Noise Ratio (HNR) of the glottal source. For this system, the F0 information is used as VAD to separate between voiced and unvoiced frames. HNR is evaluated based on the ratio between the upper and lower smoothed spectral envelopes (defined by the harmonic peaks and inter-harmonic valleys) and averaged across five frequency bands according to the equivalent rectangular bandwidth (ERB) scale. Choosing the same selected parameters reported in [7], our feature vector was made by concatenating the: F0 and the five HNR coefficients, and then calculating the SDCs coefficients on the same way as for the acoustic systems in our primary submission. Then, we used the iVectors technique following the same approach as for the Acoustic system.

In addition, and in an effort to try to select the same parameters reported in [7], we also included into our feature vector the selected parameters of the vocal tract obtained using Line Spectral Frequency (LSF) with a vector of length 30, the selected parameters of the spectral tilt of the glottal source modeled using LSFs (with 10 LSFs), and the Normalized Amplitude Quotient (NAQ) [9] and the magnitude differences between the 10 first harmonics of the voice source.

Finally, it is important to mention that in the process of extracting the features using GlottHMM, it uses a high-pass filter whose purpose is to reduce low-frequency components that may cause large fluctuations in the estimated glottal flow signal. The tool includes two files with the coefficients required to process files sampled at 16KHz or 44KHz, that can be specified through the configuration file. In addition, files with other sampling rates can be processed thanks to a simple Matlab script for designing FIR filters that is provided with the package. In our case, since the original files provided by the organizers from the Kalaka3 database are sampled at 16 KHz we used the default corresponding coefficients file provided by the tool.

6.2 Results

Table 6 shows the results for the systems presented.

Condition	System	Closed		Open	
		Fact	Cavgx100	Fact	Cavgx100
Plenty	400iv, 64 Gauss.	0.720873	33.11	0.740958	34.02
	400iv, 128 Gauss.	0.633388	28.61	0.715139	32.26
	400iv, 256 Gauss.	0.673944	30.67	0.710156	32.94
	400iv, 512 Gauss.	0.668717	30.81	0.718101	32.64
Empty	400iv, 8 Gauss.	0.075131	2.38	0.163136	7.99
	400iv, 16 Gauss.	0.082595	2.29	0.162338	7.96
	400iv, 32 Gauss.	0.113325	4.19	0.210829	11.33
	400iv, 64 Gauss	0.160465	5.06	0.304118	15.28
	400iv, 128 Gauss.	0.211179	8.18	0.472886	23.32

Table 6. Reported results for the GlottHMM system on the test set

As this system provided worse results it was only used in the contrastive systems for all conditions. In the plenty conditions, best results are obtained using 128 Gaussians. In the empty conditions, results clearly improved using very few Gaussians. We finally decided to use only 16 Gaussians for the contrastive system using this features.

7 Classifier and Calibration Back-end

As classifier for our three iVectors systems, we have used a Multiclass logistic regression which generates a different classifier for each language to recognize. Then, these classifiers are used to generate scores for the files in our development and test sets. For calibration and fusion, a Gaussian Back-end followed by a Discriminative

Multi-Class Logistic Regression is used to post-process the scores obtained before. Previously, the input vectors were conditioned by within-class covariance normalization (WCCN).

Regarding our calibration and fusion module, it is mainly based on the FoCal toolkit[3], so we did not use the tools provided by the Albayzin organizers.

Table 7 shows the results for the systems presented.

System 1	System 2	System 3	Closed		Open	
			Fact	Cavgx100	Fact	Cavgx100
MFCC-512G	Phon-1089G	RPLP-512G	0.069258	4.16	0.080184	5.77
MFCC-512G	Phon-1089G	Glot-128G	0.071393	4.16	-	-
RPLP-512G	Phon-1089G	Glot-512G	-	-	0.079517	5.37

Table 7. Reported results for the fusion of all systems for the plenty condition

The first line corresponds to the primary system submitted to the evaluation and the second and third lines correspond to the contrastive systems using GlotHMM. We can see that the contrastive system in the Open condition is slightly better than the primary one, even though the results for GlotHMM and clearly worse than for the MFCC system. Also, the GlotHMM with 512 Gaussians provided slightly better results than with 128 Gaussians. However, we decided to keep the primary system using MFCC.

8 Summary

As a summary, here is a list of the systems presented for the evaluation:

Condition	System 1	System 2	System 3
Plenty- closed-primary	MFCC-512G	Phon-1089G	RPLP-512G
Plenty-closed-contrastive1	MFCC-512G	Phon-1089G	Glot-128G
Plenty-open-primary	MFCC-512G	Phon-1089G	RPLP-512G
Plenty-open-contrastive1	RPLP-512G	Phon-1089G	Glot-512G
Empty-primary	MFCC-64G	Phon-1089G	RPLP-256G
Empty-contrastive1	RPLP-256G	Phon-1089G	Glot-16G
Empty-open-primary	MFCC-64G	Phon-1089G	RPLP-256G
Empty-open-contrastive1	RPLP-256G	Phon-1089G	Glot-16G

Table 8. Summary of all systems presented to the evaluation

[3] https://sites.google.com/site/nikobrummer/focal

9 Acknowledgment

This work has been supported by TIMPANO (TIN2011-28169-C05-03), INAPRA (MICINN, DPI2010-21247-C02-02), MA2VICMR (Comunidad Autonoma de Madrid, S2009/TIC-1542) projects, and the European project Simple4All (under grant agreement No. 287678).

10 Bibliography

1. Najim Dehak, Réda Dehak, Patrick Kenny, Niko Brümmer, Pierre Ouellet, and Pierre Dumouchel, “Support vector machines versus fast scoring in the low dimensional total variability space for speaker verification,” in Proc. of Interspeech 2009, Brighton, UK.
2. Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-End Factor Analysis For Speaker Verification", IEEE Transactions on Audio, Speech and Language Processing, Vol. 19 (4), May 2011.
3. P.A. Torres-Carrasquillo, E. Singer, M.A. Kohler, R.J. Greene, D.A. Reynolds, and J.R. Deller Jr., “Approaches to language identification using Gaussian mixture models and shifted delta cepstral features,” in Proc. International Conferences on Spoken Language Processing (ICSLP), Sept. 2002, pp. 89–92.
4. D. Martínez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language Recognition in iVectors Space", in Proc. of Interspeech 2011.
5. N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language Recognition via Ivectors and Dimensionality Reduction", in Proc. of Interspeech 2011.
6. Nicolas Obin, “MeLos: Analysis and Modelling of Speech Prosody and Speaking Style”, PhD, Ircam-UPMC, Paris, 2011.
7. Jaime Lorenzo-Trueba, Roberto Barra-Chicote, Tuomo Raitio, Nicolas Obin, Paavo Alku, Junichi Yamagishi, Juan M Montero. "Towards Glottal Source Controllability in Expressive Speech Synthesis", in Proc. of Interspeech 2012.
8. Raitio, T. and Suni, A. and Yamagishi, J. and Pulakka, H. and Nurminen, J. and Vainio, M. and Alku, P., ”HMM-based speech synthesis utilizing glottal inverse filtering”, Audio, Speech, and Language Processing, IEEE Transactions on, 9:153-165, IEEE, 2011.
9. P. Alku, T. Bäckström, and E. Vilkman, “Normalized amplitude quotient for parameterization of the glottal flow”, J. Acoust. Soc. Amer., vol. 112, no. 2, pp. 701-710, 2002.
10. Luis Fernando D’Haro, Ondrej Glembek, Oldrich Plchot, Pavel Matejka, Mehdi Soufifar, Ricardo Cordoba, Jan Cernocký. “Phonotactic Language Recognition using i-vectors and Phoneme Posteriorgram Counts”, in Proc. of Interspeech 2012.
11. Kockmann, et al, 2010. “Prosodic speaker verification using subspace multinomial models with intersession compensation,” in Proc. of ICSPL, Makuhari, Chiba, Japan, 2010.
12. D. Povey, Lukas Burget et. al, 2011. "The Subspace Gaussian Mixture Model– a Structured Model for Speech Recognition", Computer Speech and Language, 25(2), pp. 404-439
13. Rajnoha, J., and Pollák, P. 2011. “ASR systems in Noisy Environment: Analysis and Solutions for Increasing Noise Robustness”. Radionegineering, Vol. 20, No. 1, April 2011, pp. 74-84.
14. Hönig, F., Stemmer, G., Hacker, C., Brugnara, F. "Revising Perceptual Linear Prediction (PLP)". In Eurospeech 2005, p. 2997-3000.