University of the Basque Country (EHU) Systems for the NIST 2015 Language Recognition Evaluation

Mireia Diez, Mikel Penagarikano, Amparo Varona,
Luis J. Rodríguez-Fuentes, Germán Bordel

GTTS (http://gtts.ehu.es), Department of Electricity and Electronics
University of the Basque Country UPV/EHU, 48940, Leioa, Spain
e-mail: mireia.diez@ehu.es

1 Introduction

This paper briefly describes the language recognition systems developed by the Software Technology Working Group (http://gtts.ehu.es) of the University of the Basque Country (EHU) for the NIST 2015 Language Recognition Evaluation. The submitted systems follow the Total Variability Factor Analysis (*i-Vector*) approach. The systems use either Mel-Frequency Cepstral Coefficients (MFCC) or Phone Log-Likelihood Ratio (PLLR) as features [1]. Different modelings and scorings were applied on top of the ivectors, and several combinations of them were fused for the final submissions.

2 Dataset partition

First, Speech Activity Detection (SAD) was applied to all the signals provided for system development in the NIST 2015 LRE. Signals were then cut into 30-second segments. The resulting set of segments was partitioned into training, development and test subsets, according to the following criteria:

- For languages with more than 800 30s segments, 150 segments were chosen for development and 150 segments for test, the remaining ones being used for training.
- For languages containing between 300 and 800 30s segments, 150 segments were chosen for development and the remaining ones were used for training. Note that these languages didn't contribute data to the test subset.

This work has been supported by the University of the Basque Country UPV/EHU, under grant GIU13/28.

- The remaining languages were handled as follows: in the case of *zho-yue*, 100 segments were chosen for development and the remaining ones were used for training; in the case of *eng-gbr* and *por-brz*, 30 segments were chosen for development (in each case) and the remaining ones were used for training.

In all cases, segments extracted from a given signal were all allocated to the same set (either training, development or test). Therefore, the numbers shown above denote the minimum number of segments posted for development and test (actual numbers are slightly higher). Finally, the development and test sets were balanced according to the nature of the speech whenever CTS and BN training data were available for a language.

3 i-vector Extraction

3.1 PLLR features

Phone Posterior Extraction

Kaldi [2] was used to train a NNet-based acoustic model. Training was only based on LDC97S62 (Switchboard-1 Release 2) and NIST provided Mississippi State University transcripts. The NNet model was used to extract frame-level phone posteriors from audio signals.

PLLR computation

Given a phone decoder that outputs an N-dimensional vector of phone posteriors at each frame: $\mathbf{p} = (p_1, p_2, \ldots, p_N)$, such that $\sum_{i=1}^{N} p_i = 1$ and $p_i \in [0, 1]$ for $i = 1, 2, \ldots, N$, PLLRs are computed as follows:

$$r_i = \text{logit}(p_i) = \log \frac{p_i}{(1 - p_i)} \qquad i = 1, ..., N \tag{1}$$

Besides phonetic units, the phone decoder provides special non-phonetic units representing silence, noises and other non-linguistic events. These units were integrated into a single non-phonetic unit. The PLLR systems developed in this work performed SAD by removing the feature vectors whose highest PLLR value was obtained for the integrated non-phonetic unit.

3.2 MFCC Features

The concatenation of MFCC and Shifted Delta Cepstrum coefficients under a 7-2-3-7 configuration was used as acoustic representation in MFCC-iVector systems. SAD was performed by removing the feature vectors whose highest PLLR value was obtained for the integrated non-phonetic unit, using the Brno University of Technology decoder for Hungarian [3], as done in [4].

3.3 i-vector configuration

For each set of features (PLLRs and MFCCs), a gender independent 1024-mixture GMM was used as UBM, and estimated by Maximum Likelihood using a partition of the LDC English data. The total variability matrix T was estimated as in [5], using the training set defined in Section 2. The i-vector dimensionality was set to 500.

4 Classifiers

4.1 Generative Gaussian (G)

In the generative multiclass Gaussian classiffier, the distribution of language ivectors is modeled by a multivariate normal distribution $\mathcal{N}(\mu_l, \mathbf{\Sigma})$ for each target language $l \in L$, where the full covariance matrix $\mathbf{\Sigma}$ is shared across all target languages. Maximum Likelihood (ML) estimates of the language dependent means μ_l and the covariance matrix $\mathbf{\Sigma}$ are computed. For each target language l, the i-vector scores are given by:

$$score(f, l) = \log(N(x_f; \mu_l, \Sigma)) \tag{2}$$

where x_f is the feature vector (i-vector) for target signal f. The score vector

$$\mathbf{s}_f = [score(f, 1), \ldots, score(f, L)] \tag{3}$$

can be obtained by:

$$\mathbf{s}_f = \mathbf{A}x_f + \mathbf{b} + \mathbf{c}_f \tag{4}$$

where the rows of $\mathbf{A}$ are:

$$\mathbf{a}_l = \mu_l^T \mathbf{\Sigma}^{-1} \tag{5}$$

and the elements of $\mathbf{b}$ and $\mathbf{c}_f$ are:

$$b_l = -\frac{1}{2}\mu_l^T \mathbf{\Sigma}^{-1} \mu_l \tag{6}$$

$$c_{fl} = -\frac{K}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{\Sigma}| - \frac{1}{2}x_f^T \mathbf{\Sigma}^{-1} x_f \tag{7}$$

4.2 Fully Bayesian Generative Gaussian (FBG)

Under the multiclass, fully Bayesian, generative Gaussian classifier paradigm [6], the distribution of language i-vectors is modeled by a multivariate normal distribution $\mathcal{N}(\mu_l, \mathbf{\Sigma})$, but instead of using a maximum likelihood estimate of the model parameters (as in the generative Gaussian classifier), an integration is made over all possible parameters, according to their proper prior distributions.

4.3 Logistic Regression (LR)

In a multiclass logistic regression classifier, the score vector $\mathbf{s}_f$ corresponding to an i-vector x_f is given by:

$$\mathbf{s}_f = \mathbf{Q}x_f + \mathbf{u} \tag{8}$$

where $\mathbf{Q}$ and $\mathbf{u}$ are estimated in order to minimize the multi-class cross entropy [7, 8].

4.4 Neural Network (NN)

A neural network consisting of three hidden layers, each of size 512, with rectifier activations, was trained using the PDNN toolkit [9]. Dropout factors of 0.4 and $(0.3, 0.2, 0.1)$ were applied to the input and hidden layers respectively. All the parameters were heuristically tuned on the development set.

5 Backend and Fusion

The backend is just a classifier applied to the score vector space. It is usually trained on a development dataset, and it serves as a pre-calibration step of the system. Two different backends were applied on top of the scores obtained by the previously presented classifiers:

- **Fully Bayesian Generative Gaussian Backend (FBG).** It is equivalent to the fully Bayesian generative Gaussian classifier described in Section 4.2.

- **Discriminative Gaussian Backend (DG).** In this case, Maximum Likelihood estimates of the means and the common covariance matrix are used initially, but further reestimates of the means are iteratively computed in order to maximize the Maximum Mutual Information (MMI) criterion:

  $$F_{\mathbf{MMI}}(\lambda) = \sum_{\forall \mathbf{s}} \log \frac{p_\lambda\left(\mathbf{s}|l_{true}\left(\mathbf{s}\right)\right)^C}{\sum_{\forall l} p_\lambda\left(\mathbf{s}|l\right)^C p\left(l\right)} \tag{9}$$

  where $l_{true}(\mathbf{s})$ is the *true* target language, $p_\lambda\left(\mathbf{s}|l\right) = \mathcal{N}(\mathbf{s}; \mu_l, \mathbf{\Sigma})$ is the likelihood of the score vector $\mathbf{s}$ given the language l, $p(l)$ is the probability of language l and C is a heuristic factor.

5.1 Fusion

Linear logistic regression fusion parameters were estimated on the development dataset using the FoCal Toolkit [10].

6 LLR computation

Log-Likelihood Ratios (LLR) were computed from the calibrated and fused scores $\mathbf{s} = [s_1, s_2, \ldots, s_L]$ as follows:

$$LLR_i = \log \left(\frac{e^{s_i}}{\frac{1}{N_i - 1} \sum\limits_{\substack{j \in C_i \\ j \neq i}} e^{s_j}} \right) \tag{10}$$

where i is the target language, C_i is the cluster where the target language i belongs to and N_i is the number of languages in C_i.

7 EHU submission

The EHU submission was built on 12 different subsystems, using different combinations of features, classifiers and backends:

(1) PLLR - FBG classifier - DG backend

(2) PLLR - FBG classifier - FBG backend

(3) PLLR - LR classifier - DG backend

(4) PLLR - LR classifier - FBG backend

(5) PLLR - NN classifier - DG backend

(6) PLLR - NN classifier - FBG backend

(7) MFCC - FBG classifier - DG backend

(8) MFCC - FBG classifier - FBG backend

(9) MFCC - LR classifier- DG backend

(10) MFCC - LR classifier- FBG backend

(11) MFCC - NN classifier - DG backend

(12) MFCC - NN classifier - FBG backend

The 8 systems submitted by EHU within the established deadline consisted of different fusions of the subsystems (1-12) listed above. Details are shown in Table 1.

Table 1: Systems submitted by EHU to NIST 2015 LRE. Each system consisted of the fusion of a different choice of subsystems.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
Primary	×		×		×						×	
Alternative 1			×									
Alternative 2			×								×	
Alternative 3		×						×				
Alternative 4			×		×						×	
Alternative 5		×		×		×						×
Alternative 6	×		×		×				×		×	
Alternative 7	×		×		×		×		×		×	

8 Processing speed

The processing speed of EHU systems was measured on a dual Xeon E52450 2.16 GHz processor (featuring 16 cores), with 64 GB of RAM. As shown in Table 2, the CPU time required to process the input signals was extremely low. In particular, the time corresponding to scorings was almost negligible, so it has been omitted. Note also that the UBM and the Total Variability matrix are estimated beforehand.

Table 2: CPU time, in terms of Real-Time factors (×RT), for i-Vector systems based on MFCC and PLLR features.

	Features	CPU time (×RT)
Parameterization	MFCC	0.001
	PLLR	0.012
UBM	MFCC	0.006
	PLLR	0.036
T matrix	MFCC	0.060
	PLLR	0.067
i-Vector extraction	MFCC	0.003
	PLLR	0.004

References

[1] Mireia Diez, Amparo Varona, Mikel Penagarikano, Luis Javier Rodriguez-Fuentes, and Germán Bordel, "On the Use of Log-Likelihood Ratios as Features in Spoken Language Recognition," in *IEEE Workshop on Spoken Language Technology (SLT 2012)*, Miami, Florida, USA, December 2012.

[2] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

[3] Petr Schwarz, *Phoneme recognition based on long temporal context*, Ph.D. thesis, Faculty of Information Technology, Brno University of Technology, http://www.fit.vutbr.cz/, Brno, Czech Republic, 2008.

[4] Mireia Diez, "Frame-Level Features Conveying Phonetic Information for Language and Speaker Recognition," in *PhD Thesis*, University of the Basque Country, Leioa, Spain, September 2015.

[5] N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[6] N. Brummer, *Generative, Fully Bayesian, Gaussian Pattern Classifier*, https://sites.google.com/site/nikobrummer/.

[7] David A Van Leeuwen and N Brummer, "Channel-dependent gmm and multi-class logistic regression models for language recognition," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*. IEEE, 2006, pp. 1–8.

[8] N. Brümmer and J. du Preez, "Application-Independent Evaluation of Speaker Detection," *Computer, Speech and Language*, vol. 20, no. 2-3, pp. 230–275, April-July 2006.

[9] Yajie Miao, "Kaldi+pdnn: Building dnn-based ASR systems with kaldi and PDNN," *CoRR*, vol. abs/1401.6984, 2014.

[10] *FoCal Multi-class: Tools for evaluation, calibration and fusion of, and decision-making with, multi-class statistical pattern recognition scores*, https://sites.google.com/site/nikobrummer/focalmulticlass.