The BLZ Systems for the 2011 NIST Language Recognition Evaluation

Luis Javier Rodríguez-Fuentes[1], Mikel Penagarikano[1], Amparo Varona[1], Mireia Díez[1], Germán Bordel[1], Alberto Abad[2], David Martínez[3], Jesus Villalba[3], Alfonso Ortega[3], Eduardo Lleida[3]

[1] *GTTS, Department of Electricity and Electronics, University of the Basque Country, Spain*
[2] *L^2F - Spoken Language Systems Lab, INESC-ID Lisboa, Portugal*
[3] *Aragon Institute for Engineering Research (I3A), University of Zaragoza, Spain*

luisjavier.rodriguez@ehu.es

Abstract

This paper briefly describes the language recognition systems developed for the 2011 NIST Language Recognition Evaluation (LRE) by the BLZ (Bilbao-Lisboa-Zaragoza) team, a three-site joint including GTTS from the University of the Basque Country (Spain), L^2F (Spoken Language Systems Lab) from INESC-ID Lisboa (Portugal) and I3A from the University of Zaragoza (Spain). The primary system fuses 8 (3 acoustic + 5 phonotactic) subsystems: a Linearized Eigenchannel GMM (LE-GMM) subsystem, a JFA subsystem, an iVector subsystem, three Phone-SVM subsystems using the Brno University of Technology phone decoders for Czech, Hungarian and Russian, and two Phone-SVM subsystems using the L^2F phone decoders for European Portuguese and Brazilian Portuguese. Gaussian backends and multiclass fusion have been applied to get the final scores. Three contrastive systems have been also submitted, featuring: (1) the fusion of the whole set of 13 (6 acoustic + 7 phonotactic) subsystems; (2) the fusion of 3 subsystems, for the combination of one subsystem per site yielding the best performance on development data; and (3) the fusion of the same 8 subsystems used in the primary system under a different configuration.

1. Introduction

BLZ (Bilbao-Lisboa-Zaragoza) is a three-site team including GTTS from the University of the Basque Country (Spain), L^2F (Spoken Language Systems Lab) from INESC-ID Lisboa (Portugal) and I3A from the University of Zaragoza (Spain). The three sites have made their own submissions to 2011 NIST LRE, including a larger set of subsystems [1, 2, 3]. The motivation for a joint submission was the potential of fusing subsystems based on different approaches or trained on different datasets. Sometimes, two identical systems trained on different data complement each other surprisingly well. The collaboration has focused on collecting and sharing data for the 9 newly added target languages, and on defining a common development set to allow the estimation of backend and fusion parameters on independent data (i.e. not used to train models). The FoCal and Bosaris toolkits [4, 5] have been used to estimate and apply common backend and fusion approaches, and to evaluate recognition performance for the development of systems.

The BLZ submission to the 2011 NIST LRE includes one primary and three contrastive systems, combining multi-site acoustic and phonotactic subsystems under four different configurations. The primary system fuses 8 (3 acoustic + 5 phonotactic) subsystems: a Linearized Eigenchannel GMM (LE-GMM) subsystem, a JFA subsystem, an iVector subsystem, three Phone-SVM subsystems using the Brno University of Technology (BUT) phone decoders for Czech, Hungarian and Russian, and two Phone-SVM subsystems using the L^2F phone decoders for European Portuguese and Brazilian Portuguese. Gaussian backends and multiclass fusion have been applied to get the final scores. Three contrastive systems have been also submitted, featuring: (1) the fusion of the whole set of 13 (6 acoustic + 7 phonotactic) subsystems contributed by BLZ partners; (2) the fusion of 3 subsystems, for the combination of one subsystem per site yielding the best performance on development data; and (3) the fusion of the same 8 subsystems used in the primary system under a different backend configuration (including a *zt-norm* and using different datasets to estimate backend and fusion parameters).

The paper is organized as follows. Section 2 describes the datasets used for system development, which have been partly shared among sites and partly defined specifically by each site. The main features of the subsystems developed at each site are briefly outlined in Section 3 (detailed descriptions of the EHU, L^2F and I3A submissions can be found in [1, 2, 3]). Finally, Section 4 describes the systems included in the BLZ submission, along with the backend and fusion approaches on which the submission relies and the procedure followed to choose the combination of subsystems for the primary and the three contrastive systems.

2. Train and development data

2.1. Data collection for the newly added target languages

NIST has provided a development dataset specifically collected for this evaluation, including 100 30-second segments for each of the newly added target languages, except for Lao, for which only 93 segments have been provided. We have augmented the dataset with 10- and 3-second segments extracted from the original 30-second segments. Hereafter, we will refer to this dataset as *lre11*. We have randomly split *lre11* into two disjoint subsets, each having approximately half the segments for each language:

- *lre11-train*. This subset is intended to train specific models for the newly added languages.
- *lre11-dev*. This subset is intended for the estimation of backend and fusion parameters for the joint submission. It has been also used to evaluate the performance of subsystems and fused systems during development (see Section 4 for details). In any case, sites should never use this subset to train language models.

Each of these subsets have around 13 minutes of speech per target language, which may be enough to estimate backend and fusion parameters, but not to train robust models. So, we decided to collect speech data for the newly added languages.

In some cases we collected telephone speech directly from the source (that was the case of CTS databases and BN databases including telephone speech). When this was not possible, we collected broadcast news speech, downsampled it to 8 kHz and applied the *Filtering and Noise Adding Tool* (FANT)[1] to filter speech data with a frequency characteristic as defined by ITU for telephone equipment.

VOA data provided for the 2009 NIST LRE were explored in first place, starting from the labels provided by NIST. Music and fragments in English were automatically detected and filtered out, retaining only telephone-channel speech fragments. Around two hours of Lao were extracted this way. Then we used databases distributed by the LDC, some of them containing conversational telephone speech (LDC2006S45 for Arabic Iraqi and LDC2006S29 for Arabic Levantine) and others broadcast news with fragments of telephone speech (LDC2000S89 and LDC2009S02 for Czech). In both cases, segments containing telephone speech were extracted with no further processing.

The remaining materials were extracted from wideband broadcast news recordings, dowsampling them to 8 kHz and applying FANT to simulate a telephone channel. The COST278 Broadcast News database [6] was used to get speech segments for Czech and Slovak. Arabic MSA was extracted from Al Jazeera broadcasts included in the Kalaka-2 database created for the Albayzin 2010 LRE [7]. Finally, broadcasts were also *captured* from video archives in TV websites to get speech segments in Arabic Maghrebi (Arrabia TV, http://www.arrabia.ma) and Polish (Telewizja Polska, TVP INFO, http://tvp.info). TV broadcasts were fully audited, so that only reasonably clean speech segments were selected for training.

We were not able to collect by any means additional training materials for Panjabi. Hereafter, we wil refer to the data collected for the newly added target languages as *BLZ-train*.

2.2. Train data

The collaboration for a joint submission partly focused on sharing and collecting training data for the newly added target languages, but each site was free to use any other data for system development, which is interesting for fusion. In the following paragrahs, we simply give an account of the data used at each site (see [1, 2, 3] for details).

2.2.1. EHU train data

EHU defined 66 training subsets, very heterogeneous in size and composition, corresponding to different languages/dialects, including target and non-target languages and different sources. A different model was trained on each subset, which means that EHU subsystems output 66 scores. The EHU training dataset comprised the following subsets:

- Conversational telephone speech (CTS) from previous LRE: (1) the Call-Friend Corpus; (2) the OHSU Corpus provided by NIST for the 2005 LRE; and (3) the development corpus provided by NIST for the 2007 LRE.
- Narrowband speech segments extracted from VOA broadcasts, which were provided by NIST for the 2009 LRE [8][9].
- The lre11-train corpus, which amounts to half of the segments provided by NIST for the newly added target languages in the 2011 LRE.
- The BLZ-train corpus, shared and collected by BLZ partners specifically for the 2011 NIST LRE, covering eight of the newly added target languages.

[1] http://dnt.kr.hs-niederrhein.de/download.html

2.2.2. L^2F train data

The L^2F training dataset was composed by several sources of data, including data provided by NIST for previous evaluations and other external sources like LDC corpora and captured TV broadcast data. The training dataset was designed under three fundamental criteria: (a) to have data from all the NIST LRE11 target languages; (b) to consider separate datasets for training data obtained from different sources, i.e. conversational telephone speech (CTS), Voice of America (VOA), etc.; and (c) to keep the size of the training dataset as small as possible.

Attending to the language and source, the L^2F training dataset comprised 43 independent subsets, extracted from various sources:

- The VOA3 corpus from NIST LRE09 was used to extract data in 14 languages: Bengali, Dari, English American, Farsi, Hindi, Mandarin, Pashto, Russian, Spanish, Thai, Lao, Turkish, Ukrainian and Urdu. Speech segments were extracted based on labels provided by NIST, further refined by detecting and discarding segments in English or with music, and then detecting and retaining telephone-channel speech.
- Corpora from NIST 1996, 2003 and 2005 LRE were used to extract conversational telephone speech (CTS) for 7 languages: English American, English Indian, Farsi, Hindi, Mandarin, Spanish and Tamil.
- The NIST 2007 LRE corpus was used to extract speech segments for Bengali, Russian, Thai and Urdu.
- The lre11-train and BLZ-train corpora, as described in Section 2.1, were used to get speech segments for the 9 newly added target languages. Note that the Lao segments included in BLZ-train have been already counted above in the VOA3 corpus.

In order to keep a reduced and balanced training dataset, only a fraction of the available data was selected. For the 34 subsets not coming from lre11-train, 200 30-second segments and 100 10-second segments were randomly selected (if possible). For the 9 subsets coming from lre11-train, all the available speech segments were selected except for the 3-second segments for Panjabi. The whole L^2F training dataset consisted of 8128 segments, lasting almost 60 hours (less than 2.5 hours per target language, on average).

2.2.3. I3A train data

The I3A training dataset contained speech from 61 different languages, extracted from the following sources:

- NIST 2003 and 2005 LRE,
- CallFriend,
- OHSU,
- NIST 2004, 2006, 2008 and 2010 SRE,
- Switchboard,
- the VOA3 development data provided by NIST for LRE09,
- the lre11-train dataset for the 9 newly added target languages in LRE11, as defined in Section 2.1,

- the BLZ-train dataset, as defined in Section 2.1, including data from the COST278 BN database for Czech and Slovak and additional data for Arabic MSA, Arabic Maghrebi and Polish, specifically captured from TV broadcast stations for this evaluation.

For the VOA3 data, only the segments marked as telephone by NIST were used. Then, segments in English were automatically detected and removed. Finally, repeated speakers within the same language were detected and the amount of data from them were limited, to avoid undesired biases.

2.3. Development data

The criterion applied to define the development set was making the process of tuning systems as robust and reliable as possible, so we decided to use only segments audited by NIST. To cover all the target languages, the evaluation sets of the NIST 2007 and 2009 LREs (only the segments corresponding to NIST 2011 LRE target languages), together with the lre11-dev subset, as defined in Section 2.1, were used. We defined three development subsets: *dev30*, *dev10* and *dev03*, corresponding to nominal durations of 30, 10 and 3 seconds, containing 8539, 8343 and 8290 segments, respectively. Target languages showed large differences in the number of segments amongst each other. In particular, the newly added target languages were the less populated, with around 50 segments each, and thereby, they were the most likely to suffer from overtraining and/or robustness issues.

3. The BLZ subsystems

3.1. EHU subsystems

3.1.1. The EHU Dot Scoring subsystem

Acoustic features consisted of the concatenation of 7 Mel-Frequency Cepstral Coefficients (MFCC) and the Shifted Delta Cepstrum (SDC) coefficients under a 7-2-3-7 configuration. A gender independent 1024-mixture GMM was estimated on the training dataset. Then, for each input utterance, MAP adaptation was applied and the centered zero-order and first-order Baum-Welch statistics were used as features.

The Linearized Eigenchannel GMM (LE-GMM) approach, also known as *Dot-Scoring*, makes use of a linearized procedure to score test segments against target models (see [10] for details). Channel compensation was performed by using Niko Brümer's recipe [11]. The channel matrix was estimated using only data from target languages.

3.1.2. The EHU iVector subsystem

In the EHU approach, the estimation of the total variability matrix T and the computation of iVectors started from the channel-compensated sufficient statistics computed for the EHU Dot-Scoring subsystem. Except for that, the approach described in [12] was followed. The total variability matrix was estimated using only data from target languages.

3.1.3. The EHU Phonotactic subsystems

Three phonotactic sub-systems were developed under a phone-lattice-SVM approach, using the Temporal Patterns Neural Network (TRAPs/NN) phone decoders developed by the Brno University of Technology (BUT) for Czech (CZ), Hungarian (HU) and Russian (RU) [13]. Regarding channel compensation, noise reduction, etc. the three subsystems relied on the acoustic front-end provided by BUT decoders. Lattices produced by BUT phone decoders were used to produce expected counts of phone n-grams, by means of HTK [14]. Finally, a Support Vector Machine classifier was applied, SVM vectors consisting of counts of phone n-grams up to $n = 3$, weighted as in [15]. A Crammer and Singer solver for multiclass SVMs with linear kernels was applied, by means of LIBLINEAR [16].

3.2. L^2F subsystems

3.2.1. The L^2F PRSVM subsystems

Four Phone Recognition followed by Support Vector Machine Modelling (PRSVM) subsystems were developed, exploiting the phonotactic information extracted by phone recognizers for European Portuguese (*pt*), Brazilian Portuguese (*br*), European Spanish (*es*) and American English (*en*). Phone recognition was performed by means of the hybrid ASR system AUDIMUS [17], based on MultiLayer Perceptrons (MLP) trained on different sets of acoustic features and providing the posterior probabilities of phones for a given speech segment. Monophone units were used, resulting in MLP networks of 39, 40, 30 and 41 softmax outputs, respectively.

For each input speech signal, the recognized phone lattice was used to compute posterior expected n-gram counts (up to trigrams), by means of the 'lattice-tool' program from the SRILM toolkit [18]. Finally, n-gram counts were stacked in a single vector. The high-dimensionality of the n-gram vectors (between 30000 and 75000 components) motivated the use of dimensionality reduction techniques. In particular, frequency ranking was applied to get reduced vectors including only the counts of the 10000 most frequent n-grams.

For each target language and for each phone recognizer, an L2-regularized support vector classifier was trained on the corresponding set of training vectors, using the LibLinear implementation of the libSVM tool [19]. In order to avoid problems with discriminative training, data coming from subsets with the same target language than that of the subset being trained were taken neither as positive nor negative samples.

3.2.2. The L^2F GSV subsystem

This subsystem followed the approach known as GSV [20], which has been successfully applied to both speaker and language verification. The acoustic features were SDC of Perceptual Linear Prediction features with log-RelAtive SpecTrAl speech processing (PLP-RASTA), under a 7-1-3-7 configuration. Low-energy frames were detected and removed by applying a simple bi-Gaussian model of the log energy distribution.

A 1024-mixture GMM-UBM was trained on approximately 150 randomly selected speech segments per training subset (which amounted to 5200 speech segments, almost 24 hours of speech). Training subsets coming from lre11-train were excluded, except for those in Panjabi. One single iteration MAP adaptation with relevance factor 16 was performed for each speech segment to obtain a high-dimensional feature vector of size 56×1024.

SVM language models were trained on MAP-adapted GMM supervectors by means of the LibLinear implementation of the libSVM tool [19], using a linear kernel based on the Kullback-Leibler (KL) divergence. Again, data coming from subsets with the same target language than that of the subset being trained were not considered.

3.2.3. The L^2F iVector subsystem

The L^2F iVector subsystem follows the approach in [12] (where the distribution of iVectors for each language is modelled with a single Gaussian), using PLP-RASTA SDC features and a 1024-mixture GMM-UBM, as described for the L^2F GSV subsystem. The total variability matrix (**T**) was estimated according to [21], the dimension of the total variability subspace being fixed to 400. **T** was trained on zero and first-order sufficient statistics estimated on the training dataset (including Panjabi data from lre11-train). The **T** matrix was used to extract the total variability factors as in [21], the resulting vectors, normalized to unity, being referred to as iVectors. Like in [12], the iVectors extracted for a given training subset were used to estimate a single mixture Gaussian distribution with full covariance matrix shared across different training subsets.

3.3. I3A subsystems

3.3.1. The I3A iVector subsystem

The I3A iVector subsystem was also based on the approach presented in [12]. Acoustic vectors included 7 static MFCC and 49 SDC coefficients computed under a 7-1-3-7 configuration. Vocal Tract Length Normalization (VTLN) and Cepstral Mean and Variance Normalization were applied in MFCC computation. A 2048-mixture GMM-UBM was used. Both the GMM-UBM and the total variability matrix **T** were trained on the whole training dataset, including 61 different languages. Once the iVectors were obtained, a linear generative classifier was trained as in [12]. The distributions of iVectors for individual languages were modeled by Gaussian distributions with a single within-class full covariance matrix shared by all the languages. Only target languages were modeled in this step, using 500 speech segments per language (when possible).

3.3.2. The I3A Joint Factor Analysis (JFA) subsystem

This subsystem followed the principles in [22], using the same 56-dimensional acoustic features and the same 2048-mixture GMM-UBM as for the I3A iVector subsystem. Two factors were defined, one for the language and one for the channel. Thus, a channel compensated model for each language was obtained. The whole training dataset (including 61 different languages) was used to estimate model parameters. Finally, each utterance was scored via linear scoring, as proposed in [23] (see [3] for details).

4. The BLZ submission

The BLZ submission consists of one primary and three contrastive systems, whose features are summarized in Table 1. Backend and fusion were estimated and applied separately for each nominal duration, under two different configurations:

1. Gaussian backend. Training datasets for backend and fusion: dev10+dev30 for 10- and 30-second segments, dev03+dev10+dev30 for 3-second segments.
2. *zt-norm* + Gaussian backend. Training datasets for backend and fusion: dev30 for 30-second segments, dev10 for 10-second segments and dev03 for 3-second segments.

The EHU, L^2F and I3A subsystems produce 66, 43 and 24 scores, respectively (one score per trained model). These scores are taken as input by the backend, which outputs 24 log-likelihoods, one per target language. A Gaussian backend (preceded by a *zt-norm* [24] for the contrastive system 3) has been applied in all cases. Finally, the resulting $N \times 24$ log-likelihood values (N: number of subsystems) are fused to get 24 calibrated scores for which a minimum expected cost Bayes decision is made, according to application-dependent language priors and costs. The *FoCal* toolkit has been used to estimate the backend and calibration/fusion models, applying linear logistic regression under a multiclass paradigm [25, 26].

Table 1: BLZ primary and contrastive systems: configuration and fused subsystems.

		Subsystems		
System	**Config**	**EHU**	L^2F	**I3A**
Pri	(1)	Phone-CZ Phone-HU Phone-RU DotScoring	PRSVM-*pt* PRSVM-*br*	iVector JFA
Con1	(1)	Phone-CZ Phone-HU Phone-RU DotScoring iVector	PRSVM-*pt* PRSVM-*br* PRSVM-*en* PRSVM-*es* GSV iVector	iVector JFA
Con2	(1)	Phone-RU	PRSVM-*es*	JFA
Con3	(2)	Phone-CZ Phone-HU Phone-RU DotScoring	PRSVM-*pt* PRSVM-*br*	iVector JFA

4.1. Selection of subsystems

To select the best combinations of subsystems, the development set was split in two halves, the first being used to estimate backend and fusion parameters and the second to generate a set of trials, on which the performance measure, as defined in the Evaluation Plan, was computed, using the Bosaris toolkit [5]. In fact, to have a more robust measure of system performance, 10 random partitions (always the same) were defined and the average performance was computed on them. This strategy pursues (via random subset selection) the same goal than a 2-fold cross-validation strategy, but providing a better balance between the size of the evaluation subset (large enough for the results to be reliable) and the number of partitions considered in the average (for statistical significance).

Decisions were made based on system performance on the subset of 30-second segments, applying a Gaussian backend and a discriminative multiclass fusion by means of FoCal. An exhaustive search was tried in first place, by computing system performance for all the k-combinations of 13 subsystems. However, this was a very costly process so we aborted it for $k = 5$. Instead, we tried a much faster greedy process: the best k-combination was determined by extending the best $(k-1)$-combination with each of the available subsystems and that yielding the best performance was selected. Though this should generally lead to suboptimal solutions, we found that the best greedy k-combinations for k = 1, 2, 3 and 4 matched the optimal ones (those previously found with an exhaustive search). The actual and minimum C_{avg} for the optimal combinations under the greedy approach are graphically shown in Figure 1. For the primary system, the best overall combination was selected according to the evolution of the actual cost. The combination involving eight subsystems was chosen because the actual cost, which had monotonically decreased to that point, began to in-

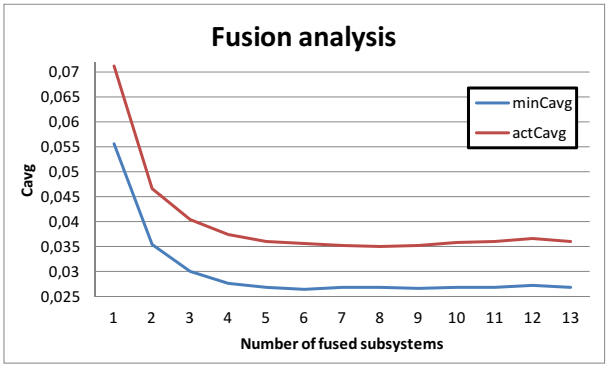

Figure 1: Actual and minimum average costs on the development set (30-second segments) for the optimal combinations of k subsystems according to a greedy algorithm.

crease for combinations of higher order. The fusion of *all* the available subsystems was selected for the first contrastive system. The best combination of three subsystems involving one subsystem per site was selected for the second contrastive system. Finally, for the third contrastive system, the same combination used for the primary system was selected, but under a different backend/fusion configuration, which provided a slight performance improvement on the development dataset.

4.2. Processing times

At EHU, a computer with 2 Intel Xeon 5550 CPUs (x4 cores, x2 turbo HT) running at 2.66GHz and 32GB of memory was used for processing the evaluation data. Processing times reported by EHU were all measured on this machine. At L^2F, a cluster of computers under the *Condor* framework for parallelization of tasks, was used. Thus, to approximately estimate computation times at L^2F, 300 speech files (amounting to 4520 seconds) from the evaluation set were chosen and language recognition tests were run in a computer with 2 Intel Xeon E5530 CPUs (x4 cores, x2 turbo HT) running at 2.40GHz with 24GB of memory. At I3A, a high-performance computing cluster was used to process evaluation data, including 1534 processors of various types and 3.5 TB of total memory, under *Condor* software for parallelization. Processing times reported by I3A have been calculated by processing ten 30-second speech files in a Dual Nehalem processor at 2.33 GHz with 24 GB of memory (one of the processor types included in the I3A cluster). Real-time factors for all the subsystems, the sub-processes and the fused systems are shown in Table 2.

Note that the computation time reported for the fused systems is lower than the addition of times for the corresponding subsystems, since some of the sub-processes were shared by two or more subsystems. For instance, the EHU iVector subsystem relies on the compensated statistics computed for the EHU Dot-Scoring subsystem. The same applies for the I3A iVector subsystem, which relies on the feature extraction and statistics computation performed for the I3A JFA subsystem. In some cases, sub-processes with relatively small (negligible) run times (such as iVector scoring and SVM vector scoring for the EHU subsystems) have not been taken into account. Processing times for the backend and fusion operations have been also omitted, since they are extremely fast. The primary system ran at 1.5745 times real time (the same as the contrastive system 3). The contrastive system 1, which fused all the available subsystems, ran at 2.4951 times real time, whereas the contrastive system 2 (including three subsystems, one per site) was the only system able to operate in real time ($0.7898 \times RT$).

Table 2: Real-time factors of subsystems, sub-processes and fused systems. The real-time factor of fused systems is computed by adding the real-time factors of the involved subsystems, the shared sub-processes being counted just once.

EHU	**Dot-Scoring**		**0.0467**
		Acoustic Parameterization	0.0020
		Sufficient Statistics	0.0187
		Channel Compensation	0.0260
	iVector		**0.0717**
		Acoustic Parameterization	0.0020
		Sufficient Statistics	0.0187
		Channel Compensation	0.0260
		iVector Extraction	0.0250
	Phone-SVM-CZ		**0.2114**
		Lattice Decoding	0.1267
		Expected Counts	0.0847
	Phone-SVM-HU		**0.2300**
		Lattice Decoding	0.1517
		Expected Counts	0.0783
	Phone-SVM-RU		**0.2164**
		Lattice Decoding	0.1327
		Expected Counts	0.0837
L^2F	**PRSVM**		**0.6928**
		Feature Extraction	0.0518
		Lattice Decoding + Counts *pt*	0.1383
		Lattice Decoding + Counts *br*	0.1648
		Lattice Decoding + Counts *es*	0.1330
		Lattice Decoding + Counts *en*	0.2049
	GSV		**0.4588**
		Feature Extraction	0.1631
		Supervector Computation	0.1925
		Scoring	0.1031
	i-Vector		**0.2620**
		Feature Extraction	0.1631
		Sufficient Statistics	0.0425
		i-Vector Extraction	0.0564
I3A	**JFA**		**0.3886**
		Feature Extraction	0.2777
		Statistics Computation	0.0748
		Channel Factors + Scoring	0.0361
	i-Vector		**0.4790**
		Feature Extraction	0.2777
		Statistics Computation	0.0748
		iVector Extraction	0.1143
		iVector Classification	0.0122
BLZ	**Primary**		**1.5745**
	Contrastive 1		**2.4951**
	Contrastive 2		**0.7898**
	Contrastive 3		**1.5745**

5. Acknowledgements

The work by the EHU team has been supported by the University of the Basque Country (EHU) under grant GIU10/18, by the Government of the Basque Country under program SAIOTEK (project S-PE10UN87) and by the Spanish MICINN under Plan Nacional de I+D+i (project TIN2009-07446, partially financed by FEDER funds). The work by Alberto Abad has been partially suported by FCT (INESC-ID multi-annual funding) through the PIDDAC Program funds, and also through the EU-funded project *EUTV Adaptive Media Channels*. The work of I3A was funded by the Spanish Ministry of Science and Innovation under project TIN2008-06856-C05-04.

6. References

[1] M. Penagarikano, A. Varona, L. J. Rodriguez-Fuentes, M. Diez, and G. Bordel, "University of the Basque Country (EHU) Systems for the 2011 NIST Language Recognition Evaluation," in *The 2011 NIST Language Recognition Evaluation (LRE11) Workshop Booklet*, Atlanta, USA, December 6-7 2011.

[2] A. Abad, "The L^2F Language Recognition System for NIST LRE 2011," in *The 2011 NIST Language Recognition Evaluation (LRE11) Workshop Booklet*, Atlanta, USA, December 6-7 2011.

[3] D. Martinez, J. Villalba, A. Ortega, and E. Lleida, "I3A Language Recognition System Description for NIST LRE 2011," in *The 2011 NIST Language Recognition Evaluation (LRE11) Workshop Booklet*, Atlanta, USA, December 6-7 2011.

[4] *FoCal: Toolkit for Evaluation, Fusion and Calibration of statistical pattern recognizers*, http://sites.google.com/site/ nikobrummer/focal.

[5] *Bosaris Toolkit: MATLAB tools useful for processing of speaker recognition scores*, http://sites.google.com/site/ bosaristoolkit.

[6] A. Vandecatseye, J.-P. Martens, J. Neto, H. Meinedo, C. Garcia-Mateo, J. Dieguez, F. Mihelic, J. Zibert, J. Nouza, P. David, M. Pleva, A. Cizmar, H. Papageorgiou, and C. Alexandris, "The COST278 pan-European Broadcast News Database," in *Proceedings of the LREC 2004*, Lisbon, Portugal, 2004, pp. 873–876.

[7] L. J. Rodriguez-Fuentes, M. Penagarikano, A. Varona, M. Diez, and G. Bordel, "The Albayzin 2010 Language Recognition Evaluation," in *Proceedings of Interspeech*, Firenze, Italia, August 28-31 2011, pp. 1529–1532.

[8] *The 2009 NIST Language Recognition Evaluation Plan (LRE09)*, http://www.itl.nist.gov/iad/mig/tests/lre/2009/ LRE09_EvalPlan_v6.pdf.

[9] A. Martin and C. Greenberg, "The 2009 NIST Language Recognition Evaluation," in *Proceedings of Odyssey 2010: The Speaker and Language Recognition Workshop*, Brno, Czech Republic, 2010, pp. 165–171.

[10] A. Strasheim and N. Brümmer, "SUNSDV system description: NIST SRE 2008," in *NIST Speaker Recognition Evaluation Workshop Booklet*, 2008.

[11] N. Brümmer, A. Strasheim, V. Hubeika, P. Matejka, L. Burget, and O. Glembek, "Discriminative Acoustic Language Recognition via Channel-Compensated GMM Statistics," in *Proceedings of Interspeech*, Brighton, UK, 2009, pp. 2187–2190.

[12] D. Martínez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language Recognition in iVectors Space," in *Proceedings of Interspeech*, Firenze, Italy, 2011, pp. 861–864.

[13] P. Schwarz, "Phoneme recognition based on long temporal context," Ph.D. dissertation, Faculty of Information Technology, Brno University of Technology, http://www.fit.vutbr.cz/ schwarzp/publi/thesis.pdf, 2008.

[14] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Lui, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department, UK, 2006.

[15] F. Richardson and W. Campbell, "Language recognition with discriminative keyword selection," in *Proceedings of ICASSP*, 2008, pp. 4145–4148.

[16] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008, software available at http://www.csie.ntu.edu.tw/∼cjlin/liblinear.

[17] H. Meinedo, A. Abad, T. Pellegrini, I. Trancoso, and J. Neto, "The L2F Broadcast News Speech Recognition System," in *Proceedings of FALA 2010: VI Jornadas en Tecnologia del Habla and II Iberian SLTech Workshop*, Vigo (Spain), 10-12 November 2010.

[18] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proceedings of Interspeech*, November 2002, pp. 257–286.

[19] C. Chang and C. Lin, *LIBSVM: a library for support vector machines*, 2001, http://www.csie.ntu.edu.tw/ ∼cjlin/libsvm.

[20] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[21] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A Study of Inter-Speaker Variability in Speaker Verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 5, pp. 980–988, July 2008.

[22] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," CRIM, Tech. Rep. Technical Report CRIM-06/08-13, 2005, [Online]. Available: http://www.crim.ca/perso/patrick.kenny/.

[23] O. Glembek, L. Burget, N. Dehak, N. Brümmer, and P. Kenny, "Comparison of Scoring Methods Used in Speaker Recognition with Joint Factor Analysis," in *Proceedings of IEEE ICASSP*, Taipei, April 2009, pp. 4057–4060.

[24] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42–54, January 2000.

[25] N. Brümmer and D. van Leeuwen, "On calibration of language recognition scores," in *Proceedings of Odyssey - The Speaker and Language Recognition Workshop*, 2006, pp. 1–8.

[26] N. Brümmer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.