The L^{2}F Language Recognition System for NIST LRE 2011

Alberto Abad

L^{2}F - Spoken Language Systems Lab, INESC-ID

alberto.abad@l2f.inesc-id.pt

Abstract

This document presents a description of INESC-ID's Spoken Language Systems Laboratory (L^{2}F) Language Recognition systems submitted to the 2011 NIST Language Recognition evaluation. The L^{2}F *primary* system consists of the fusion of six individual sub-systems: four phonotactic sub-systems and two acoustic based sub-systems. The major differences of the submitted LR system with respect to previous L^{2}F system submitted to the NIST LRE 2009 campaign are: a) use of SVM discriminative modelling of expected phone counts extracted from lattices in contrast to generative n-gram modelling of phoneme sequences in phonotactic systems, b) development of a single kernel based system of Gaussian supervectors with support vector machine modelling, and c) incorporation of a new i-vector based system with linear generative classifiers. Additionally, two contrastive systems have been submitted. One fundamental particularity of the L^{2}F submission is that a relatively small training data set was defined and used for building the several sub-systems. Thus, the "small" training data set permitted fast development and comparison of algorithms and new subsystems.

1. Introduction

The National Institute of Standards and Technology (NIST) has organized in the last years a series of evaluations in some relevant speech processing topics devoted to encourage language research activities.

In the 2011 NIST Language Recognition Evaluation (LRE11) the objective is to decide which of the two languages of a specified language pair is spoken in a given segment of speech. The number of possible target languages is 24, resulting in 276 possible language pairs. There are three segment duration test conditions, corresponding to nominal durations of 30, 10 and 3 seconds. Detailed information on the LRE11 campaign can be found in the evaluation plan document [1].

Language recognition (LR) approaches can generally be classified according to the kind of source of information that they rely on. The most successful systems are based on the exploitation of the acoustic phonetics, that is the acoustic characteristics of each language, or the phonotactics which are the rules that govern the phone combinations in a language. Usually, the combination of different sources of knowledge and systems of different characteristics tends to provide increased language recognition performances [2].

This document presents the LR systems developed by the INESC-ID's Spoken Language Systems Laboratory (L^{2}F) for the LRE11 campaign. The *primary* system is composed by the fusion of six individual LR sub-systems: four phonotactic and two acoustic-based. The four phonotactic systems are phone recognizers followed by support vector machine modelling (PRSVM) [3] that exploits the phonotactic information extracted by four multi-layer perceptron tokenizers. The first acoustic system is the well-known Gaussian SuperVector (GSV) approach with linear SVM kernel of [4] based on the Kullback-Leibler (KL) divergence modelling. The second acoustic system is an i-vector [5] based language recognition system similar to the one in [6] that makes use of single mixture Gaussian distributions for language modelling. Additionally, two contrastive systems have been submitted, both also composed by the fusion of the six individual sub-systems. The *contrastive1* system uses a language-pair dependent logistic regression for fusion, in contrast to the multi-class fusion used by the *primary* system. The *contrastive2* system uses the new data provided by NIST for this year evaluation only for back-end training and calibration, in contrast to the *primary* system that uses a sub-set of this new data set for language modelling also.

In next section 2 a brief description of the data used for language model training and for back-end development is provided. Section 3 provides details of each one of the six individual sub-systems: the PRSVM-LR, the GSV-LR and the iVECTOR-LR sub-systems are described in sections 3.1, 3.2 and 3.3, respectively. Finally, the three submitted systems are described in section 4.

2. Training and development data

2.1. Data for acoustic and phonotactic modelling

The training data set used for the evaluation is 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.

Three fundamental design criteria have been followed for the definition of the training data set: a) Try to have training data from all NIST LRE11 target languages, b) Use separate language data sets for training data obtained from different sources, i.e.: conversational telephone speech (CTS), Voice of America (VOA), etc., and c) Keep a reduced training data set.

2.1.1. Data sources

With respect to languages present in the training data set, we first identified the following languages that were present in VOA3 corpus from previous NIST LRE09: Bengali, Dari, English American, Farsi, Hindi, Mandarin, Pashto, Russian, Spanish, Thai, Lao, Turkish, Ukrainian and Urdu. We extracted speech segments from these languages based on labels provided by NIST and we further purified them based on in-house speech segmentation, music detection, telephone detection and English language detection. These data sets belong to the "voa" type of data.

From *lid96d1*, *lid96e1*, *lid03e1*, *lid05d1* and *lid05e1* corpora we extracted conversational telephone speech (CTS) data for the following languages: English American, English Indian,

Lang	source	#seg	minutes
Arabic Iraqi	ldc.cts	300	74.5
Arabic Iraqi	lid11d1	156	41.9
Arabic Levantine	ldc.cts	300	97.5
Arabic Levantine	lid11d1	153	41.1
Arabic Maghrebi	bn	160	107.3
Arabic Maghrebi	lid11d1	138	37.3
Arabic MSA	bn	300	122.6
Arabic MSA	lid11d1	144	38.0
Bengali	voa	237	142.8
Bengali	cts	300	68.2
Czech	ldc.bn	300	130.2
Czech	bn	300	88.6
Czech	ldc.voa	171	66.6
Czech	lid11d1	132	35.6873
Dari	voa	300	136.3
English American	voa	293	180.7
English American	cts	300	169.8
English Indian	cts	278	180.4
Farsi	voa	212	139.8
Farsi	cts	300	121.6
Hindi	voa	221	141.4
Hindi	cts	300	149.5
Lao	voa	253	133.2
Lao	lid11d1	156	42.3
Mandarin	voa	300	131.4
Mandarin	cts	300	170.8
Panjabi	lid11d1	110	40.4
Pashto	voa	300	145.2
Polish	bn	281	107.5
Polish	lid11d1	162	44.3
Russian	voa	227	141.9
Russian	cts	300	86.6
Slovak	bn	300	101.1
Slovak	lid11d1	132	35.8
Spanish	voa	180	138.1
Spanish	cts	300	168.8
Tamil	cts	300	164.9
Thai	voa	300	132.1
Thai	cts	300	56.0
Turkish	voa	300	159.1
Ukrainian	voa	300	130.6
Urdu	voa	254	139.6
Urdu	cts	300	63.9
Total	–	8128	3554.3

Table 1: *Language training data sets with their corresponding target language, source type, number of speech segments and total duration in minutes.*

Farsi, Hindi, Mandarin, Spanish and Tamil. Also from previous *lid07tr1* corpus, we obtained speech segments for Bengali, Russian, Thai and Urdu. For these four language sets, English language detection was also applied to reject English segments. These data sets belong to the "cts" type of data.

In addition to these data extracted from previous LRE campaigns, we obtained CTS data for Arabic Iraqi and Arabic Levantine from LDC corpora LDC2006S45 and LDC2006S29 respectively. Long conversations were automatically segmented. These data sets belong to the "ldc.cts" type of data.

Also from LDC, VOA Czech Broadcast News audio (LDC2000S89) and Czech Broadcast Conversation Speech (LDC2009S02) was used. In both cases, telephone segments were automatically detected and further processed to automatically detect and reject segments with music and English speech. We identify these sets as "ldc.voa" and "ldc.bn" types, respectively.

Wide-band broadcast news data for Czech and Slovak was extracted from the COST278 pan-European Broadcast News Database [7]. Additional wide-band data for Arabic Maghrebi, Arabic MSA and Polish was captured from TV broadcast[1]. For these five wide-band broadcast language sets, the Filtering and Noise Adding Tool (FANT)[2] was used to filter speech data with a frequency characteristic as defined by ITU for telephone equipment. These data sets belong to the "bn" type of data.

Lang	30 sec	10 sec	3 sec	Tot
Arabic Iraqi	48	48	48	144
Arabic Levantine	49	49	49	147
Arabic Maghrebi	54	54	54	162
Arabic MSA	51	51	51	153
Bengali	123	117	123	363
Czech	56	56	56	168
Dari	389	389	389	1167
English American	976	943	936	2855
English Indian	734	699	671	2104
Farsi	470	465	464	1399
Hindi	827	778	797	2402
Lao	41	41	41	123
Mandarin	1173	1149	1128	3450
Panjabi	86	84	83	253
Pashto	395	395	395	1185
Polish	46	46	46	138
Russian	671	652	643	1966
Slovak	56	56	56	168
Spanish	625	625	625	1875
Tamil	160	160	160	480
Thai	268	247	236	751
Turkish	394	394	394	1182
Ukrainian	388	388	388	1164
Urdu	459	457	457	1373
Total	8539	8343	8290	25172

Table 2: *Development data set: Number of speech segments for each target language and nominal duration.*

Finally, new data provided by NIST for new LRE11 languages (Arabic Iraqi, Arabic Levantine, Arabic Maghrebi, Arabic MSA, , Czech , Lao, Panjabi, Polish and Slovak) was split in two balanced sub-sets that we named *lid11d1* and *lid11d2*. In both sub-sets, 10 seconds and 3 seconds sub-segments were extracted from each original segment. The *lid11d1* sub-set was used for training, while *lid11d2* was kept for development. Notice that the only new language included in *lid11d1* set for which we were not able to obtain data from other sources is Panjabi.

Considering the different sources and languages, a total of 43 independent training data sets have been defined: 9 obtained from the *lid11d1* data set and 34 from the other sources described.

2.1.2. Data selection

In order to keep a reduced and balanced training data set we decided to select only a sub-set of all the training data available described above. Particularly, for the 34 data sets that were not extracted from *lid11d1*, 200 segments of 30 seconds nominal duration and 100 segments of 10 seconds nominal duration were randomly selected whenever it was possible. Anyway, we selected at most 300 segments for every training sub-set. For the 9 data sub-sets extracted from *lid11d1*, we selected all the available speech segments except the 3-seconds Panjabi segments that were not included (see section 4.4). Table 1 summarizes the data used for training the L^{2}F language recognition system, which consists of 8128 segments with a total duration of almost 60 hours (in average, less than 2.5 hours per target language).

[1] Thanks to the EHU team for data recording.

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

2.2. Data for back-end development

Every LRE11 target language segment present in previous LRE07 and LRE09 evaluation data sets (*lid07e1* and *lid09e1*) together with the *lid11d2* data sub-set described above compose the development data used for calibration and fusion of the L^2F language recognition systems submitted to LRE11. It consists of a total of 25172 speech segments, split in 30 seconds, 10 seconds and 3 seconds nominal duration. Table 2 summarizes the development data set.

3. LR sub-system description

Six sub-systems form the core of the L^2F LR system: 4 PRSVM phonotatic systems, one GSV based detector and an i-vector based classifier. For each sub-system, a separate target language model is trained with the data of each one of training data subsets of Table 1. Consequently, for every test segment a vector of 43 scores $\mathbf{x}_i$ is produced by every individual sub-system i.

3.1. PRSVM-LR sub-systems

Phone Recognition followed by Support Vector Machine Modelling (PRSVM) systems used for LRE11 exploit the phonotactic information extracted by four individual tokenizers: European Portuguese (*pt*), Brazilian Portuguese (*br*), European Spanish (*es*) and American English (*en*). The key aspect of this type of system is the need for robust phonetic recognizers that generally need to be trained with word-level or phonetic level transcriptions. In this case, the tokenizers are MultiLayer Perceptrons (MLP) trained to estimate the posterior probabilities of the different phonemes for a given input speech frame (and its context). A recognition lattice is generated for every processed segment, from which the posterior expected n-gram counts are computed. For each target language and for each tokenizer a different phonotactic SVM language model is trained with the counts vectors. During test, vectors of n-gram counts of a given speech signal are computed from the lattices obtained with the automatic phoneme recognizers and used to evaluate each language SVM model.

3.1.1. Phoneme Recognizers

Vectors of expected n-gram counts are obtained for each speech segment based on the recognition results of our hybrid Automatic Speech Recognition (ASR) system named AUDIMUS [8]. The recognizers combine four MLP outputs trained with Perceptual Linear Prediction features (PLP, 13 static + first derivative), PLP with log-RelAtive SpecTrAl speech processing features (PLP-RASTA, 13 static + first derivative), Modulation SpectroGram features (MSG, 28 static) and Advanced Font-End from ETSI features (ETSI, 13 static + first and second derivatives). A phone-loop grammar with phoneme minimum duration of three frames is used for lattice generation.

The language-dependent MLP networks were trained with different amounts of annotated data. For the *pt* acoustic models, 57 hours of BN downsampled data and 58 hours of mixed fixed-telephone and mobile-telephone data were used. The *br* models were trained with around 13 hours of BN down-sampled data. The *es* networks used 36 hours of BN down-sampled data and 21 hours of fixed-telephone data. The *en* system was trained with the HUB-4 96 and HUB-4 97 down-sampled data sets, that contain around 142 hours of TV and Radio Broadcast data.

Each MLP network is characterized by the size of its input layer that depends on the particular parametrization and the frame context size (13 for PLP, PLP-RASTA and ETSI; 15 for MSG), the number of units of the two hidden layers (500), and the size of the output layer. In this case, only monophone units are modelled, resulting in MLP networks of 41 (39 phonemes +1 silence + 1 respiration) soft-max outputs in the case of *en*, 39 for *pt* (38 phonemes + 1 silence), 40 for *br* (39 phonemes + 1 silence) and 30 for *es* (29 phonemes + 1 silence).

3.1.2. N-gram vector extraction and dimensionality reduction

The 'lattice-tool' program from the SRILM toolkit[3] is used to compute the expected n-gram counts (up to 3-grams) of each recognition lattice. This resulting n-gram counts vector is converted to a vector of probabilities (sum 1) and it is normalized by the square root of the average probability vector computed over the whole training data set.

During the development of the systems, the high-dimensionality of the n-gram vectors –63971 for *pt*, 74102 for *en*, 29799 for *es* and 68893 for *br*– motivated the study of some dimensionality reduction methods. In practice, we compared the baseline to simple frequency selection [9] and PCA vector reduction [10] . In both cases we could only observe very modest performance improvements for some particular dimension with respect to the baseline. We believe that the modest improvements can be partially explained by the SVM ability to ignore dimensions that are low discriminant. Anyway, for shake of efficiency, we decided to apply frequency selection dimensionality reduction with new dimensionality of 10000 elements in the four PRSVM sub-systems (this size was experimentally verified to provide good performance).

3.1.3. Phonotactics Modelling

For every phoneme recognizer, phonotactic realations of each training data sub-set are modelled with an L2-regularized support vector classifier using the LibLinear implementation of the libSVM tool[4]. Notice that there are several training data sets that correspond to a same target language. In order to avoid problems with discriminative training, for every trained model only the data of the corresponding data sub-set is considered as positive examples for SVM training, while data from all the other sub-sets is used as negative examples, except the data of the sub-sets with the same target language than the trained one that is ignored.

3.2. GSV-LR sub-system

A method generally known as GSV [4] is known to be a successful approach for both speaker and language verification tasks. GSV-based approaches map each speech utterance to a high-dimensional vector space. Support Vector Machines (SVMs) are used for classification of test vectors within this space. The mapping to the high-dimensional space is achieved by stacking all parameters (usually the means) of an adapted GMM in a single supervector by means of a Bayesian adaptation of a universal background model (GMM-UBM) to the characteristics of a given speech segment. In language recognition, a binary SVM classifier is trained for each target language with supervectors of the target language as positive examples and supervectors of other non-target languages as negative examples. During test, the supervector of the testing speech utterance is used by the binary classifier to generate a score for each target language.

[3] http://www-speech.sri.com/projects/srilm/

[4] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

3.2.1. Feature Extraction

The extracted features are shifted delta cepstra (SDC) [11] of Perceptual Linear Prediction features with log-RelAtive SpecTrAl speech processing (PLP-RASTA). First, 7 PLP-RASTA static features are obtained and mean and variance normalization is applied in a per segment basis. Then, SDC features (with a 7-1-3-7 configuration) are computed, resulting in a feature vector of 56 components. Finally, low-energy frames detected with the alignment generated by a simple bi-Gaussian model of the log energy distribution computed for each speech segment are removed.

3.2.2. Supervector Extraction and SVM Language Modelling

A GMM-UBM of 1024 mixtures was trained with approximately 150 speech segments per training data sub-set randomly selected among the training data of Table 1. In this random selection, data coming form *lid11d1* sources was excluded, except data for Panjabi language (see section 4.4). The resulting amount of data used for UBM training consisted of 5200 speech segments totalling almost 24 hours of speech.

One single iteration of Maximum a Posteriori (MAP) adaptation with relevance factor 16 is performed for each speech segment to obtain the high-dimensional vector of size 56x1024.

The linear SVM kernel of [4] based on the Kullback-Leibler (KL) divergence is used to train the target language models with the LibLinear implementation of the libSVM tool. For each training language sub-set, all the training segments of that subset are used as positive examples and all the segments from the other training sub-sets that have a different target language are used as negative background set (the other sub-sets with same target language are excluded).

During the development of this sub-system, we experimented applying nuisance attribute projection (NAP) [12]. Unfortunately, we did not observe significant performance improvements even for different NAP dimensionality. We do not have a definite explanation for this behaviour. It is likely that in language recognition applications, the channel variability may be captured by the model, since the training data consists of several (hundreds and even thousands) of positive training examples that represent each target language in different recording conditions. On the other hand, in the speaker verification case, there is often a single segment for training and testing. Consequently, channel mismatched conditions may have an higher performance impact and channel compensation may become more necessary. Finally, we decided not to use NAP in the submitted system.

3.3. iVECTOR-LR sub-system

Total-variability modelling [5] has rapidly emerged as one of the most powerful approaches to the problem of speaker verification. In this approach, closely related to the Joint Factor Analysis [13], the speaker and the channel variabilities of the high-dimensional GMM supervector are jointly modelled as a single low-rank total-variability space. The low-dimensionality total variability factors extracted from a given speech segment form a vector, named i-vector, which represents the speech segment in a very compact and efficient way. Thus, the total-variability modelling is used as a factor analysis based front-end extractor. In practice, since the i-vector comprises both speaker and channel variabilities, in the i-vector framework for speaker verification some sort of channel compensation or channel modelling technique usually follows the i-vector extraction process.

The success of i-vector based speaker recognition has motivated the investigation of its application to other related fields, including language recognition [6, 14]. For LRE11, we have developed an i-vector based language recognition sub-system very similar to the one in [6], where the distribution of i-vectors for each language is modelled with a single Gaussian.

3.3.1. Feature extraction and UBM modelling

We have used the same PLP-RASTA with SDC feature extraction process described in section 3.2.1 and the same GMM-UBM of 1024 mixtures used by the GSV-LR sub-system described in section 3.2.2.

3.3.2. Total variability and i-vector extraction

The total variability factor matrix (**T**) was estimated according to [15]. The dimension of the total variability sub-space was fixed to 400. Zero and first-order sufficient statistics of the training sub-sets described in Table 1 were used for training **T** (only Panjabi data was included from the *lid11d1* sub-sets, see section 4.4). 10 EM iterations were applied, in the first 7 iterations only ML estimation updates were applied, while in the last 3 EM iterations both ML and minimum divergence update were applied. The covariance matrix was not updated in any of the EM iterations.

The estimated **T** matrix is used for extraction of the total variability factors of the processing speech segments as described in [15]. Finally, the resulting factor vectors are normalized to be of unit length, which we will refer as i-vectors.

3.3.3. Language modelling and scoring

Like in [6], all the extracted i-vectors from a data sub-set of Table 1 are used to train a single mixture Gaussian distribution with full covariance matrix shared across different training subsets. For a given test i-vector, each Gaussian model is evaluated and log-likelihood scores are obtained.

4. The L^{2}F Submissions

4.1. Commonalities

4.1.1. Linear Gaussian Back-End

A linear Gaussian Back-End (GBE) follows every single subsystem to transform the 43 elements score-vector $\mathbf{x}_i$ to a 24 elements log-likelihood vector $\mathbf{s}_i$ (the 24 target languages):

$$\mathbf{s}_i = \mathbf{A}_i \mathbf{x}_i + \mathbf{o}_i \tag{1}$$

where $\mathbf{A}_i$ is the transformation matrix for system i and $\mathbf{o}_i$ is the offset vector.

4.1.2. Linear logistic regression fusion

The L^{2}F submitted systems consist of the fusion of the subsystems described in previous section 3. Linear logistic regression (LLR) has been used to fuse the log-likelihood outputs generated by the linear GBEs of the individual sub-systems to produce fused likelihoods **l**:

$$\mathbf{l} = \sum_i \alpha_i \mathbf{s}_i + \mathbf{b} \tag{2}$$

where α_i is the weight for sub-system i and **b** is the language-dependent shift.

During the development of the L^{2}F systems, the GBEs and the LLR fusion parameters were trained and tested with the development data set using a jack-knifing strategy. For the final submission, no partition of the data was made and all the development data was used to simultaneously calibrate the GBEs and the LLR fusion. Calibration was carried out using the FoCal Multiclass Toolkit[5].

4.1.3. Segment duration based score normalization

We have investigated some score segment duration normalization strategies with the aim of developing duration independent back-ends. Thus, a segment length normalization strategy similar to the one described in [16] was considered, where the scores of each individual system are augmented with multiplied versions of the duration d^p ($\mathbf{x}_i d^p$). In our attempt, we did not use duration-information in the fusion as side-information [17].

On the one hand, we observed that for some of the subsystems there exits scaled versions of the scores that obtain even better LR performance than the original scores. On the other hand, we did not observe very significant improvements by fusing several duration-normalized versions of the same scores with respect to the best (normalized or not) one. Since including more score-normalized based sub-systems implies an increase of the number of parameters of the back-end, we finally decided just to keep one set of scores per sub-system. Concretely, for the PRSVM-LR sub-systems we used the original scores without any duration normalization, while for the GSV-LR and the iVECTOR-LR sub-systems we normalize the scores by the square root of the duration ($\bar{\mathbf{x}}_i = \mathbf{x}_i/\sqrt{d}$) and by the natural logarithm of the duration ($\bar{\mathbf{x}}_i = \mathbf{x}_i/\log(d)$), respectively. Duration d is the number of high-energy frames determined by the bi-Gaussian alignment process of section 3.2.1.

4.1.4. Time dependent back-end

In addition to the segment duration normalization approach described previously, we were still able to observe LR improvements when specific back-end and fusion parameters were calibrated for different test segment durations. Particularly, we trained one specific back-end for test segments of 30 seconds nominal duration using 30 and 10 seconds duration development data, and another back-end for 10 and 3 seconds test segments using all the development data (30, 10 and 3 seconds)[6].

4.1.5. Processing times

The evaluation tests were run in a cluster of computers under the Condor framework for parallelization of tasks. In order to approximately estimate the computational time, we have separated a reduced set of data from the evaluation test set: 300 files amounting 4520 seconds (100 segments from each time duration condition). Then, we have run the language recognition tests in a computer with 2 Intel Xeon E5530 CPUs (x 4 cores x 2 turbo HT) running at 2.40GHz with 24GB of memory. Table 3 shows the real-time factors for each sub-system.

For the PRSVM based sub-systems, feature extraction processing is shared among the different phonetic decoders. The most demanding operations are lattice decoding and expected counts computation, while scoring is negligible since vector counts are reduced to sparse vectors of 10000 dimensions, which results in very efficient computation. For the GSV subsystem, the feature extraction process is highly computational demanding, partially due to the low-energy frame detection method applied, but also because of the inefficient implementation of some of the intermediate steps. In contrast to the PRSVM sub-systems, the scoring step is more significant since in this case supervectors are dense vectors of 56x1024 dimensions. The overall processing time of the GSV sub-system is penalized by the reading/writing speed of large size files in distributed file systems. For the i-VECTOR sub-system, the feature extraction process is shared with the GSV sub-system. Sufficient statistics computation and i-vector extraction are quite efficient operations, while i-vector scoring has been neglected since it is extremely fast. Finally, fusion operations have been also omitted, since they are very fast. The overall six subsystems fused system runs at 1.25 times real time.

PRSVM		**0.6928**
	Feature Extraction	0.0518
	Lattice Decod. + Counts *pt*	0.1383
	Lattice Decod. + Counts *br*	0.1648
	Lattice Decod. + Counts *es*	0.1330
	Lattice Decod. + 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
Fusion		**1.2503***

Table 3: *Real-time factors of the sub-systems and their corresponding sub-processes. The real time factor of the fused system is the sum of all the individual sub-systems factors (GSV and i-VECTOR feature extraction is only computed once).*

4.2. Primary System (primary)

The L^{2}F *primary* system consists of multi-class fusion of the six sub-systems. For a given test segment, the outcome of the fusion is a likelihood vector $\mathbf{l}$ of 24-elements, one for each target language. For every target language pair, the log-likelihood ratios between the corresponding elements of the $\mathbf{l}$ vector are the detection scores. The decision threshold is set to 0.

4.3. First Contrastive System (contrastive1)

The objective of the L^{2}F *contrastive1* system is to investigate an alternative language pair-dependent back-end method that may be more appropriate for LRE11 task. In practice, it consists of a language pair-dependent fusion of the six individual sub-systems. In contrast to the *primary* system the fusion parameters of the logistic regression are estimated separately for every language pair with the development data of the pair of languages involved. The GBE is the same as for the *primary* system, since it is generative. For a given test segment and language pair, the outcome of the fusion is now a likelihood vector $\mathbf{l}$ of 2 elements, one for each target language of the language pair, that is computed with the LLR parameters learnt for that specific language pair. Then, the log-likelihood ratio for that pair of languages is the difference between the two likelihoods of $\mathbf{l}$. The decision threshold is again set to 0.

[5] http://niko.brummer.googlepages.com/focalmulticlass

[6] Thanks to the EHU team for providing estimated speech durations for the evaluation data set

	C_{avg}^{act}			C_{avg}^{min}		
	30s	10s	3s	30s	10s	3s
Primary	**0.058** ($\pm$0.005)	**0.084** ($\pm$0.007)	**0.154** ($\pm$0.007)	**0.046** ($\pm$0.003)	**0.069** ($\pm$0.004)	**0.132** ($\pm$0.005)
PRSVM-pt	0.104 ($\pm$0.007)	0.151 ($\pm$0.008)	0.254 ($\pm$0.014)	0.087 ($\pm$0.006)	0.129 ($\pm$0.006)	0.225 ($\pm$0.010)
PRSVM-br	0.104 ($\pm$0.004)	0.145 ($\pm$0.007)	0.249 ($\pm$0.007)	0.085 ($\pm$0.004)	0.125 ($\pm$0.005)	0.223 ($\pm$0.006)
PRSVM-es	0.110 ($\pm$0.005)	0.151 ($\pm$0.007)	0.244 ($\pm$0.011)	0.088 ($\pm$0.005)	0.131 ($\pm$0.005)	0.219 ($\pm$0.008)
PRSVM-en	0.136 ($\pm$0.007)	0.172 ($\pm$0.005)	0.261 ($\pm$0.007)	0.113 ($\pm$0.005)	0.149 ($\pm$0.005)	0.230 ($\pm$0.005)
GSV	0.095 ($\pm$0.007)	0.144 ($\pm$0.005)	0.236 ($\pm$0.005)	0.080 ($\pm$0.004)	0.126 ($\pm$0.004)	0.209 ($\pm$0.005)
i-VECTOR	0.095 ($\pm$0.005)	0.132 ($\pm$0.006)	0.211 ($\pm$0.008)	0.078 ($\pm$0.004)	0.115 ($\pm$0.006)	0.189 ($\pm$0.008)
Contrastive1	**0.060** ($\pm$0.006)	**0.085** ($\pm$0.006)	**0.162** ($\pm$0.007)	**0.048** ($\pm$0.004)	**0.070** ($\pm$0.004)	**0.138** ($\pm$0.005)
PRSVM-pt	0.103 ($\pm$0.008)	0.151 ($\pm$0.008)	0.255 ($\pm$0.014)	0.087 ($\pm$0.006)	0.129 ($\pm$0.006)	0.225 ($\pm$0.010)
PRSVM-br	0.104 ($\pm$0.006)	0.145 ($\pm$0.007)	0.249 ($\pm$0.007)	0.085 ($\pm$0.004)	0.125 ($\pm$0.005)	0.223 ($\pm$0.006)
PRSVM-es	0.108 ($\pm$0.005)	0.152 ($\pm$0.006)	0.245 ($\pm$0.011)	0.088 ($\pm$0.005)	0.131 ($\pm$0.005)	0.219 ($\pm$0.008)
PRSVM-en	0.135 ($\pm$0.008)	0.173 ($\pm$0.005)	0.260 ($\pm$0.007)	0.113 ($\pm$0.005)	0.149 ($\pm$0.005)	0.230 ($\pm$0.005)
GSV	0.095 ($\pm$0.006)	0.144 ($\pm$0.004)	0.236 ($\pm$0.006)	0.080 ($\pm$0.004)	0.126 ($\pm$0.004)	0.209 ($\pm$0.005)
i-VECTOR	0.094 ($\pm$0.004)	0.132 ($\pm$0.006)	0.213 ($\pm$0.009)	0.078 ($\pm$0.004)	0.115 ($\pm$0.006)	0.189 ($\pm$0.008)
Contrastive2*	**0.077** ($\pm$0.004)	**0.107** ($\pm$0.005)	**0.184** ($\pm$0.006)	**0.063** ($\pm$0.004)	**0.092** ($\pm$0.004)	**0.163** ($\pm$0.006)
PRSVM-pt	0.132 ($\pm$0.005)	0.173 ($\pm$0.007)	0.277 ($\pm$0.008)	0.115 ($\pm$0.004)	0.156 ($\pm$0.007)	0.250 ($\pm$0.008)
PRSVM-br	0.121 ($\pm$0.004)	0.171 ($\pm$0.007)	0.264 ($\pm$0.009)	0.102 ($\pm$0.004)	0.153 ($\pm$0.006)	0.240 ($\pm$0.008)
PRSVM-es	0.131 ($\pm$0.008)	0.178 ($\pm$0.006)	0.273 ($\pm$0.007)	0.114 ($\pm$0.006)	0.158 ($\pm$0.005)	0.247 ($\pm$0.008)
PRSVM-en	0.155 ($\pm$0.006)	0.193 ($\pm$0.005)	0.287 ($\pm$0.007)	0.138 ($\pm$0.006)	0.172 ($\pm$0.004)	0.256 ($\pm$0.006)
GSV	0.129 ($\pm$0.006)	0.168 ($\pm$0.004)	0.271 ($\pm$0.005)	0.112 ($\pm$0.004)	0.151 ($\pm$0.004)	0.241 ($\pm$0.004)
i-VECTOR	0.116 ($\pm$0.006)	0.155 ($\pm$0.005)	0.246 ($\pm$0.008)	0.101 ($\pm$0.004)	0.136 ($\pm$0.004)	0.220 ($\pm$0.008)

Table 4: *Actual and minimum average cost for the L^{2}F submitted systems and for the individual sub-systems of the fusion.*

4.4. Second Contrastive System (contrastive2)

The L^{2}F *contrastive2* compares the use of additional data for language model training in contrast to its use for back-end development. It was in fact the first one to be built. In our early experiments, it was first decided to use the *lid11d1* data set partition as additional development data, except the Panjabi subset. The Panjabi *lid11d1* sub-set was in fact kept in the training data set in order to cover all the LRE11 target languages in the training corpus. Consequently, in addition to have a larger and hopefully better development data set, each sub-system i of the *contrastive2* system provides a 35-element scores vector $\mathbf{x}_i$ in contrast to the 43 scores of *primary* sub-systems. The GBEs and LLR fusion scheme and the language pair log-likelihood ratios computation are identical to the one of the *primary* system, but of course more data was used for calibrating the back-end and fusion parameters.

Notice that some components of the "primary" system were also affected by this initial decision of not using *lid11d1* data for training, namely the GMM-UBM of the GSV-LR and iVECTOR-LR sub-systems, and the total variability matrix **T** of the iVECTOR-LR sub-system. For shake of efficiency and time constraints, these components were not re-estimated for the "primary" system with additional *lid11d1* data.

4.5. Systems performance in the development set

Table 4 shows actual and minimum average cost for the L^{2}F submitted systems in the development dataset described in section 2.2. Performance costs of each individual sub-system involved in the fusion systems are also provided. In order to assess the systems, the development dataset is randomly partitioned in two halves, one used for back-end and fusion parameters estimation and the other for system scoring. This process is repeated up to 10 times and the final mean and standard deviation results over the 10 iterations are computed. Notice that a slightly different calibration/evaluation dataset is used for *contrastive2* system (see section 4.4).

5. Acknowledgements

The author would like to thank to the other colleagues of the BLZ team for the valuable discussions and the fun time shared during this evaluation. I also would like to thank to David Matos for his support with the processing machines and data management issues and to the 2011 NIST LRE organizers for their availability for solving doubts and problems. This work was 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*.

6. References

[1] "The 2011 NIST Language Recognition Evaluation Plan (LRE11)", URL: http://www.nist.gov/itl/iad/mig/upload/LRE11_EvalPlan_releasev1.pdf.

[2] Rodríguez-Fuentes, L. J., et al., "Multi-site heterogeneous system fusions for the Albayzin 2010 language recognition evaluation", IEEE 2011 Automatic Speech Recognition and Understanding Workshop (ASRU), 2011.

[3] Li, H., Ma, B. and Lee, C.-H., "A vector space modeling approach to spoken language identification", IEEE Transactions on ASLP, vol. 15, no. 1, pp. 271 284, 2007.

[4] Campbell, W. M., Sturim, D. E. and Reynolds, D. A., "Support vector machines using GMM supervectors for speaker verification", IEEE Signal Processing Letters, vol. 13(5), pp. 308-311, 2006.

[5] Dehak, N., Dehak, R., Kenny, P., Brummer, N., Ouellet, P. and Dumouchel, P., "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification", in Proc. Interspeech 2009, pp. 1559-1562, 2009.

[6] Martínez, D., Plchot, O., Burget, L., Glembek, O. and Matejka, P., "Language Recognition in iVectors Space", in Proc. Interspeech 2011, Firenze, Italy, 2011.

[7] Vandecatseye, A., et al., "The COST278 pan-European Broadcast News Database", in Proc. LREC 2004, pp. 873-876, Lisbon, Portugal, 2004.

[8] Meinedo, H., Abad, A., Pellegrini, T., Trancoso, I. and Neto, J. "The L2F Broadcast News Speech Recognition System", in Proc. Fala2010, Vigo, Spain, 2010.

[9] Tong, R., Ma, B., Li, H. and Chng, E. S., "Selecting phonotactic features for language recognition", in Proc. Interspeech 2010, pp. 737-740, September 2010.

[10] Mikolov, T., Plchot, O., Glembek, O., Matejka, P., Burget, L. and Cernocky, J., "PCA-based feature extraction for phonotactic language recognition", in Proc. Odyssey 2010 - The Speaker and Language Recognition Workshop, pp. 251-255, 2010.

[11] Torres-Carrasquillo, P. A. et al., "Approaches to Language Identification using Gaussian Mixture Models and Shifted Delta Cepstral Features", in Proceedings of ICSLP 2002, pp. 89-92, Denver, Colorado, Sep 2002.

[12] Campbell, W.M. et al, "SVM based Speaker Verification using GMM Supervector Kernel and NAP Variability Compensation", in Proc. ICASSP 2006, Toluse, France, May 2006.

[13] Kenny, P., Boulianne, G., Ouellet, P. and Dumouchel, P. "Joint factor analysis versus eigenchannels in speaker recognition", IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, no. 4, pp. 1435-1447, 2007.

[14] Dehak, N., Torres-Carrasquillo, P.A., Reynolds, D. and Dehak, R., "Language Recognition via Ivectors and Dimensionality Reduction", in Proc. Interspeech 2011, Firenze, Italy, 2011.

[15] Kenny, P., Ouellet, P., Dehak, N., Gupta, V. and Dumouchel, P., "A Study of Inter-Speaker Variability in Speaker Verification", IEEE Transactions on Audio, Speech and Language Processing, 16(5), pp. 980-988, July 2008.

[16] van Leeuwen, D. and Gonzalez-Dominguez, J., "The TNO system for LRE-2009", The 2009 NIST Language Recognition Evaluation (LRE09) Workshop, Baltimore, US, Jun 2009.

[17] Abad, A., Koller, O. and Trancoso, I., "The L2F Language Verification Systems for Albayzin-2010 Evaluation", in Proc. Fala2010, Vigo, Spain, November 2010.