Language Recognition for Dialects and Closely Related Languages

G. Gelly[1,2]*, J.L. Gauvain*[1]*, L. Lamel*[1]*, A. Laurent*[3]*, V.B. Le*[3]*, A. Messaoudi*[3]

[1]LIMSI, CNRS, Université Paris-Saclay, 91405 Orsay, France
[2]Univ. Paris-Sud, Orsay, France
[3]Vocapia Research, Orsay, France

Abstract

This paper describes our development work to design a language recognition system that can discriminate closely related languages and dialects of the same language. The work was a joint effort by LIMSI and Vocapia Research in preparation for the NIST 2015 Language Recognition Evaluation (LRE). The language recognition system results from a fusion of four core classifiers: a phonotactic component using DNN acoustic models, two purely acoustic components using a RNN model and and i-vector model, and a lexical component. Each component generates language posterior probabilities optimized to maximize the LID NCE, making their combination simple and robust. The motivation for using multiple components representing different speech knowledge is that some dialect distinctions may not be manifest at the acoustic level. We report experiments on the NIST LRE15 data and provide an analysis of the results and some post-evaluation contrasts. The 2015 LRE task focused on the identification of 20 languages clustered in 6 groups (Arabic, Chinese, English, French, Slavic and Iberic) of similar languages. Results are reported using the NIST Cavg metric which served as the primary metric for the OpenLRE15 evaluation. Results are also reported for the EER and the LER.

1. Introduction

Automatic spoken language recognition is the task of automatically determining the language spoken in a given speech segment using the characteristics of the speech signal. This paper describes recent developments and studies we carried out in preparation for the National Institute of Science and Technology (NIST) 2015 Language Recognition Evaluation (LRE). LIMSI has been developing systems for language recognition since the early 1990s, when the use of phone-based acoustic likelihoods was proposed for language identification as well as a general framework to identify non-linguistic information in the speech signal [1, 2]. The basic approach was extended to use parallel phone recognizers with phonotactic characteristics [3], lexical information [4, 5] and phone lattices [6, 7]. One of the main innovations for such types of systems was proposed by Zissman (1996), who demonstrated that phone recognizers were not required for each targeted language in order to characterize their phonotactic constraints.

The phonotactic approach relies on the assumption that the way sequences of phones are arranged is language specific [8] meaning that even if two languages share the same phonemes, their phonotactic characteristics are different. Various approaches based on phone decoding with phonotactic constraints have been explored for many years and have been shown to provide state-of-the-art results [9, 10, 11]. This papers also investigates other methods which have been used for language recognition (I-vectors and RNNs) and their combination with phonotactic and lexical approaches.

These methods were explored in the context of the NIST 2015 Language Recognition Evaluation (OpenLRE15) which aimed to distinguish closely related languages and dialects of the same language [12]. As such, the OpenLRE15 task is more challenging than previous LRE evaluations which focused on discriminating among languages. In accordance with the evaluation plan, the core testing condition was based on the use of only limited and specified training data to develop the models for each of the target languages. Data augmentation [13] was applied in order to mitigate the effect of the limited training data for some languages.

This development work presented was a joint effort between LIMSI and Vocapia Research. We use a similar architecture to the one described in [10, 11], but with HMM-DNN instead of HMM-GMM. This phonotactic system was combined with up to three other language recognition components: two purely acoustic components, an RNN model and and I-vector model, and a lexical component. System combination is just a simple averaging of the component LID posteriors which were all optimized to maximize the LID normalized cross entropy (NCE). [1]

After a description of the training and evaluation data, the component language recognizers are described, followed by extensive results and their analysis.

2. Corpora

This section describes the training data distributed for the OpenLRE15 evaluation, the selection of a set of development data from the distributed data (without prior knowledge of the characteristics of the evaluation data), and the distribution of the evaluation data set.

2.1. Training data

Table 1 gives a descriptions of the available training data, providing for each cluster the associated target languages, the number of speech files for each language and the speech duration in hours.

Since only very limited resources were provided for some of the language variants (e.g. British English), we decided to augment the training data set as proposed in [13]. Modified copies of the audio files at different speeds were created by resampling the data using the speed function of Sox. For each audio file in the training data two additional copies were created by modifying the speed to 90% and 110% of the original

[1] Since our phone decoders had been pre-trained, they did not adhere to a strict application of the evaluation rules. However, all phonotactic models used only the provided training data.

code	cluster - target language	# files	# hours
ara-arz	Arabic - Egyptian	220	95.4
ara-acm	Arabic - Iraqi	210	37.2
ara-apc	Arabic - Levantine	225	41.1
ara-ary	Arabic - Maghrebi	207	38.6
ara-arb	Arabic - Modern Standard	406	3.7
zho-yue	Chinese - Cantonese	17	3.4
zho-cmn	Chinese - Mandarin	219	71.8
zho-cdo	Chinese - Min Dong	37	8.1
zho-wuu	Chinese - Wu	36	7.7
eng-gbr	English - British	47	0.5
eng-usg	English - American	214	100.0
eng-sas	English - Indian	392	8.1
fre-waf	French - West African	34	7.7
fre-hat	French - Haitian	323	2.7
qsl-pol	Slavic - Polish	363	30.0
qsl-rus	Slavic - Russian	386	18.0
spa-car	Iberian - Caribbean Spanish	60	26.9
spa-eur	Iberian - European Spanish	38	8.1
spa-lac	Iberian - Latin American Spanish	30	6.9
por-brz	Iberian - Brazilian Portuguese	47	0.8

Table 1: Training data repartition by language cluster and target language.

code	cluster - target language	# files	# hours
ara-arz	Arabic - Egyptian	8023	41.8
ara-acm	Arabic - Iraqi	8994	47.7
ara-apc	Arabic - Levantine	6802	38.4
ara-ary	Arabic - Maghrebi	8264	46.3
ara-arb	Arabic - Modern Standard	2447	8.2
zho-yue	Chinese - Cantonese	22532	151.9
zho-cmn	Chinese - Mandarin	6026	30.4
zho-cdo	Chinese - Min Dong	8542	47.1
zho-wuu	Chinese - Wu	7496	41.2
eng-gbr	English - British	7998	27.1
eng-usg	English - American	6980	33.8
eng-sas	English - Indian	6932	35.4
fre-waf	French - West African	6935	38.3
fre-hat	French - Haitian	28741	168.6
qsl-pol	Slavic - Polish	4818	17.0
qsl-rus	Slavic - Russian	3051	11.4
spa-car	Iberian - Caribbean Spanish	2332	12.2
spa-eur	Iberian - European Spanish	5803	25.3
spa-lac	Iberian - Latin American Spanish	6973	30.3
por-brz	Iberian - Brazilian Portuguese	4645	15.3

Table 2: Repartition of language clusters and target languages and in the official LRE15 evaluation data set.

speaking rate. In order to meet the evaluation conditions, all LID models were trained on the restricted data after data augmentation.

2.2. Development data set

As no common development data set was specified in the evaluation plan, we selected a development data set from the training data. To do so, 10% of the training files were randomly selected prior to data augmentation. From these files two development sets were defined: one with 1180 long cuts containing segments ranging from 10s to 50s of speech and another with 7500 short cuts containing speech segments of 3s to 10s. Even after data augmentation, some of languages, in particular, British English and Brazilian Portuguese remained poorly represented in the development data set.

2.3. Test data

Table 2 provides information about the test data used in the official NIST evaluation. The test segments cover a wide range of speech durations from 1s to 82s with more than a third of the files containing less than 5s of speech (cf. Figure 1). The test segment durations are more varied than our internal development set and have a larger number of short segments.

3. LID System Components

The language recognition system resulting from this work fuses the results produced by four component systems, relying on acoustic-based and token-based classifiers. The token based components aim to capture the phonotactic and lexical properties of the language. This section provides brief descriptions of the four component LID systems and the score fusion.

3.1. Phonotactic component (PHO)

Phonotactic systems for language identification have been popular since the mid-1990s [1, 2, 3]. Such systems rely on the assumption that the phonotactic characteristics, that is the way phonemes make up words and sentences, differ across languages. Reliable estimation of phonotactic constraints requires highly consistent phone recognizers across varied acoustic environments in order to differentiate between languages [10, 11]. Phonotactic constraints are inferred from the speech signal using one or several phone recognizers to decode speech signals into phone sequences or phone lattices from which n-gram phonotactic statistics (representing phonotactic constraints) are estimated. Compared to the best phone hypothesis, phone lattices contain more information captured by the alternative phone sequences. Phone decoding is carried out without any grammar.

The phonotactic component of this language recognition system makes use of the Parallel Phone Recognizer followed by Language Modeling (PPRLM) approach [3]. Pre-trained phone decoders using HMM-DNN acoustic models for three languages (English, Italian and Russian) were used to decode all of the training data. Phone n-gram statistics were estimated from the resulting phone lattices. The n-gram statistics are then used to compute the expectation of the phone log-likelihood for each target language [6]. The posteriors of the three phone decoders are averaged, and used as the score for the phonotactic LID component.

3.2. I-vector component (IVC)

The i-vector framework [14] has been successfully applied to Speaker Verification [15, 16] and Language Identification [17].

The i-vector system characterizes speakers and utterances with vectors obtained by projecting their speech data onto a *total variability space* T where speaker and channel information is dense. It is generally expressed as:

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

where w is called an *i-vector* and m and S are the GMM super-vector of the speaker independent UBM and speaker adapted model, respectively.

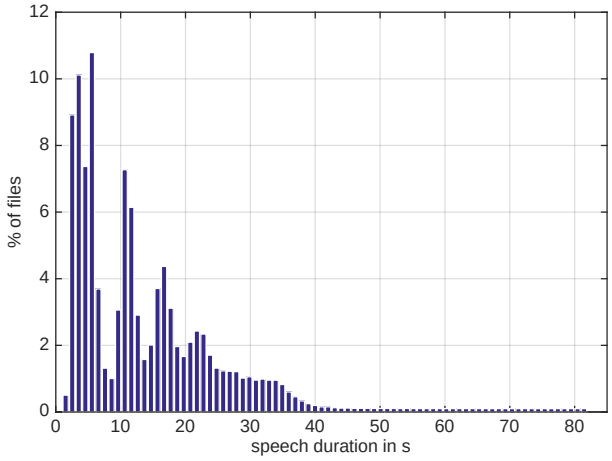

Figure 1: Distribution of the evaluation data: number of test segments as a function of speech duration.

During the test phase, the i-vector of the test utterance is scored against the claimant (hypothesized language) specific vector obtained in the training phase, after post-processing the vectors for session variability compensation. The PLDA (Probabilistic Linear Discriminant Analysis) technique [18], which is also commonly used for speaker verification [15, 16] or gender identification [19]), was used. PLDA is a generative modeling technique which decomposes the i-vector w into several components as:

$$w = \mu_w + \Phi y_s + \Gamma z + \epsilon \tag{2}$$

where μ is the mean of i-vectors obtained by averaging over the training set and Φ and Γ are rectangular matrices representing the *eigen voice* and *eigen channel* subspace respectively. y_s and z are respectively called the speaker and channel factor, and have a prior normal distribution. ϵ denotes the residual noise. In the test phase, the score between the i-vector of the claimant w_{cl} and test utterance w_{tst} is calculated as:

$$score(w_{cl}, w_{tst}) = \log \frac{p(w_{cl}, w_{tst}|\theta_{tar})}{p(w_{cl}, w_{tst}|\theta_{non})} \tag{3}$$

with hypothesis θ_{tar} that w_{cl} and w_{tst} are from the same (target) speaker and hypothesis θ_{non} that they are from different speakers. For details see [18].

The i-vector LID component uses 7 MFCC features including C0. Similar to [17], vocal tract length normalization (VTLN) and cepstral mean and variance normalization (CMVN) are applied to both the training and test data. Then the Shifted Delta Coefficients (SDC) [20] are computed and concatenated to the MFCC vector. The final feature vectors have 56 dimensions. The system is implemented using the Kaldi toolkit [21].

A full covariance GMM with 2048 components and an i-vector extractor are estimated on the training data. The i-vector dimensionality is 600. This i-vector length is normalized to unity [16]. The PLDA model is estimated on all training utterances with 10 EM iterations. Training the PLDA model on a subset of the training data (max 200 utterances per language), as proposed in [17], gave less good results than training it on the full data set.

3.3. Lexical component (LEX)

For the lexical system, the basic model is similar to the PPRLM approach, but in this case the decoding is done at the lexical level [4, 5]. Seven word-based speech recognizers, 5 monolingual and 2 multi-lingual systems, were used in the lexical component. The five monolingual systems are conversational telephone speech recognition systems for the French, Spanish, Italian, Arabic and English languages. The multi-lingual systems were trained using data from 5 languages (Arabic, Chinese, English, Spanish and French). The lexicon of the multi-lingual system is comprised of all words in all languages and contains 311k words and is represented with a set of 221 phones.

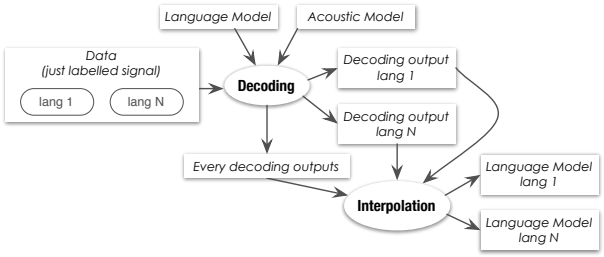

Figure 2: Schema for training the lexical LID component.

Figure 2 shows how the lexical system is trained. Each speech recognition system is used to decode the training data pool. The output of the decoding of the data corresponding to each target language is used to build a language model for that language. A multi-lingual language model is also built using the decoding output for the entire training corpus. The final model for each target language results from an interpolation of these two LMs.

To identify the language of an audio segment, the data is decoded with the seven systems. Then for each language, the likelihood of each hypothesis for each LM is computed, followed by the estimation of the posterior probabilities. Finally, the geometric mean is performed to obtain a single output vector.

3.4. RNN component (RNN)

The modified version of the BLSTM neural network introduced in [22] was used in this work. The input to the system are 8 PLP coefficients and their first and second derivatives, computed every 10 ms after VTLN is applied. Then, cepstral mean and variance normalization is performed, producing features with 24 dimensions. One recurrent neural network (RNN) based on BLSTM with a softmax layer for the output was used to produce a sequence of vectors with 20 dimensions (one for each variant). Finally, to obtain a single output vector, the geometric mean of all the vectors in the output sequence was computed.

3.4.1. Augmented BLSTM

To make use of the context around each audio frame, an RNN based on Long Short-Term Memory cells as shown in Figure 3 was used. The LSTM cells were introduced to overcome some of the shortcomings of classical RNNs [23] and were popularized after Graves demonstrated their good performance on optical character recognition and speech sequence labeling tasks [24, 25].

Given an input sequence $\boldsymbol{p} = (\boldsymbol{p}^1, ..., \boldsymbol{p}^T)$, a standard RNN computes the output vector sequence $\boldsymbol{z} = (\boldsymbol{z}^1, ..., \boldsymbol{z}^T)$ by iter-

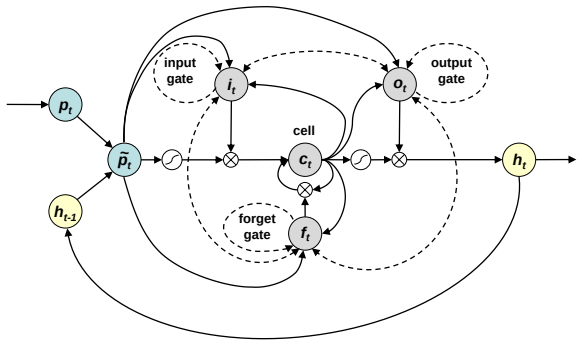

Figure 3: *LSTM cell. The dashed lines correspond to the added links between the gates for the augmented LSTM cell.*

ating the following equations from $t = 1 \rightarrow T$:

$$\boldsymbol{h}^t = \sigma_1\left(\boldsymbol{W}_1 \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{b}_1\right) \quad with \quad \tilde{\boldsymbol{p}}^t = \begin{bmatrix} \boldsymbol{p}^t \\ \boldsymbol{h}^{t-1} \end{bmatrix} \tag{4}$$

$$\boldsymbol{z}^t = \sigma_z\left(\boldsymbol{W}_z \cdot \boldsymbol{h}^t + \boldsymbol{b}_z\right) \tag{5}$$

The use of LSTM cells instead of the classic summation units modifies the computation of $\boldsymbol{h}^t$ as follows:

$$\boldsymbol{i}^t = \sigma_i\left(\boldsymbol{W}_i \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_i^c \cdot \boldsymbol{c}^{t-1} + \boldsymbol{b}_i\right) \tag{6}$$

$$\boldsymbol{f}^t = \sigma_f\left(\boldsymbol{W}_f \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_f^c \cdot \boldsymbol{c}^{t-1} + \boldsymbol{b}_f\right) \tag{7}$$

$$\boldsymbol{c}^t = diag\left(\boldsymbol{f}^t\right) \cdot \boldsymbol{c}^{t-1} + diag\left(\boldsymbol{i}^t\right) \cdot \sigma_c\left(\boldsymbol{W}_c \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{b}_c\right) \tag{8}$$

$$\boldsymbol{o}^t = \sigma_o\left(\boldsymbol{W}_o \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_o^c \cdot \boldsymbol{c}^t + \boldsymbol{b}_o\right) \tag{9}$$

$$\boldsymbol{h}^t = diag\left(\boldsymbol{o}^t\right) \cdot \sigma_h\left(\boldsymbol{c}^t\right) \tag{10}$$

where $\boldsymbol{i}^t$, $\boldsymbol{f}^t$, $\boldsymbol{c}^t$ and $\boldsymbol{o}^t$ are respectively the *input gate*, the *forget gate*, the *cell* and the *output gate* activation vectors. They are all the same size as the hidden vector $\boldsymbol{h}^t$. $\boldsymbol{W}_i^c$, $\boldsymbol{W}_f^c$, and $\boldsymbol{W}_o^c$ are diagonal matrices so that each heart of a cell is only visible to the gates of the same cell.

One shortcoming of conventional RNNs is that they are only able to make use of the left context. For LID purposes there is no reason not to exploit right context as well. Bidirectional LSTM neural networks (*BLSTM*) were developed to do just that: 2 distinct LSTM networks process the sequence both forward and backward, and then the output of both networks are combined and fed into the output layer. This way, we can fully exploit the long range capabilities of LSTM cells. In the literature (e.g. [24, 25]) BLSTM networks always outperform unidirectional ones, so only BLSTM networks were explored in this study.

In [22], a modified version of the BLSTM neural network was proposed in which direct links are added between the three gates of a LSTM cell as shown by the dashed lines in Figure 3. This modification aims to prevent that some of the LSTM cells get stuck in a saturated state when trained on long sequences.

Equations (6), (7) and (9) are thus modified into (12), (14) and (16):

$$\tilde{\boldsymbol{i}}^t = \boldsymbol{W}_i^i \cdot \boldsymbol{i}^{t-1} + \boldsymbol{W}_i^f \cdot \boldsymbol{f}^{t-1} + \boldsymbol{W}_i^o \cdot \boldsymbol{o}^{t-1} \tag{11}$$

$$\boldsymbol{i}^t = \sigma_i\left(\boldsymbol{W}_i \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_i^c \cdot \boldsymbol{c}^{t-1} + \tilde{\boldsymbol{i}}^t + \boldsymbol{b}_i\right) \tag{12}$$

$$\tilde{\boldsymbol{f}}^t = \boldsymbol{W}_f^i \cdot \boldsymbol{i}^{t-1} + \boldsymbol{W}_f^f \cdot \boldsymbol{f}^{t-1} + \boldsymbol{W}_f^o \cdot \boldsymbol{o}^{t-1} \tag{13}$$

$$\boldsymbol{f}^t = \sigma_f\left(\boldsymbol{W}_f \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_f^c \cdot \boldsymbol{c}^{t-1} + \tilde{\boldsymbol{f}}^t + \boldsymbol{b}_f\right) \tag{14}$$

$$\tilde{\boldsymbol{o}}^t = \boldsymbol{W}_o^i \cdot \boldsymbol{i}^t + \boldsymbol{W}_o^f \cdot \boldsymbol{f}^t + \boldsymbol{W}_o^o \cdot \boldsymbol{o}^{t-1} \tag{15}$$

$$\boldsymbol{o}^t = \sigma_o\left(\boldsymbol{W}_o \cdot \tilde{\boldsymbol{p}}^t + \boldsymbol{W}_o^c \cdot \boldsymbol{c}^t + \tilde{\boldsymbol{o}}^t + \boldsymbol{b}_o\right) \tag{16}$$

where the nine matrices $\boldsymbol{W}_{\{i,f,o\}}^{\{i,f,o\}}$ are diagonal so that a gate can only have access to the gates of the same cell.

With these new links the three gates of a cell can interact more efficiently and improve the cell behavior. We call this network *BLSTM+*.

3.4.2. BLSTM training

Training of the BLSTM+ neural network was performed using back-propagation through time as described in [26] and its LSTM version [24]. The algorithm used was SMORMS3 as described in [27].

Training is a four-step process. First, 20 small binary classifiers (2 BLSTM+ layers composed of 8 cells and a feed-forward layer of 2 hidden units) are trained to separate each of the 20 languages from the other 19 languages. In the second step, the 20 small RNNs are combined into a multi-class classifier: the weight matrices of the forward and recurrent links of the small RNNs are combined into block diagonal matrices for the weights of the multi-class RNN. The final RNN is composed of 2 BLSTM+ layers each with 160 cells, a feed-forward layer of 40 hidden units, and a softmax layer of 20 dimensions.

The feed-forward layer of 20x2 hidden units and the softmax layer are then trained in order to balance the behavior of the 20 small RNNs inside the multi-class RNN. Finally, the full network multi-class RNN is trained using the weights obtained before as a "very smart" initialization point.

During early experiments, this four-step training process was found to be both much faster and to lead to much better performance than a straightforward training of the multi-class RNN.

3.5. Fusion

We also tried to use a feed-forward neural network to carry out the fusion. The neural network was trained on part of the training set and on the development set. The fusion worked quite well, but due to the low quantity of data available for some of the languages, we were unable to set aside more of the training data to validate the NN-fusion. Since we could not test this approach thoroughly, we decided not to use it lest it should prove insufficiently robust.

Fusion of the outputs of the component LID systems is obtained by computing the geometric mean of their respective posteriors, the posteriors of each component having been optimized by maximizing the LID NCE.

4. Results

This section presents the results for the different systems evaluated in NIST OpenLRE15. Three evaluation metrics are used: the NIST Cavg metric which served as the primary metric for the OpenLRE15 evaluation [12], as well as the EER and the average language error rate defined as:

$$LER_{avg} = \frac{1}{n_C}\Sigma_{c \in C}\left(\frac{1}{n_{D_c}}\Sigma_{d \in D_c} Pmiss(d)\right) \tag{17}$$

where C is the set of clusters, n_C is the number of clusters, D_c is the set of variants for cluster c and n_{D_c} is the number of variants in cluster c.

Detailed results are provided for two internal development sets (long and short cuts), and the evaluation data as the system accuracy and the ranking of the component LID systems varies greatly across the data sets and metrics. Results of post-evaluation experiments aiming to reduce the mismatch between training and testing conditions are also reported.

4.1. Results on development data

Table 3 reports results obtained on the 1180 long-cut segments containing 10s to 50s of speech with an average speech duration of 27.6s. It can be seen that both acoustic systems and the phonotactic one have somewhat comparable performances, with an average LER below 8% and an EER of about 3%. Combining the RNN-based component with the phonotactic one reduces the average LER by 50% over the best single component. The performance can also be improved by including the i-vector component to the combination. Unfortunately, we were not able to further reduce the LER by combining with the lexical system on this development set.

System	*LER*	*EER*	*CAVG*
PHO	7.5	2.9	0.044
RNN	5.9	3.4	0.031
IVC	5.7	2.5	0.029
LEX	12.5	4.7	0.078
PHO+RNN	2.3	1.6	0.012
PHO+RNN+IVC	2.4	1.5	0.014
PHO+RNN+IVC+LEX	3.5	1.5	0.021

Table 3: Results on the long cut development data (1180 segments) with the four LID components and some combinations.

Table 4 shows the results obtained on the short-cut development data containing 7500 segments ranging in duration from 3s to 10s of speech, with an average speech duration is 4.5s. Surprisingly on these shorter segments, the acoustic systems are better than the phonotactic and the lexical systems. However, as was observed for the longer segments, the combination of the best of both types is very effective, leading to a 26% relative gain over the performance of the best system alone. Moreover, adding the i-vector and the lexical components to the combination yields further improvements with respect to the two-way combination. It can be seen that the best results depend on the metric, where including the lexical component improves the EER, but not the LER or Cavg over the 3-way combination.

System	*LER*	*EER*	*CAVG*
PHO	25.4	13.1	0.151
RNN	19.0	11.8	0.118
IVC	19.5	11.2	0.120
LEX	35.4	17.9	0.237
PHO+RNN	14.0	8.1	0.080
PHO+RNN+IVC	12.4	6.7	0.073
PHO+RNN+IVC+LEX	13.7	6.5	0.075

Table 4: Results on the short cut development data (7500 segments) with the four LID components and some combinations.

Figure 4 illustrates the impact of the quantity of speech on the performance of each system according to the average LER metric. The segments in the two development sets are combined and grouped into intervals according to their speech duration. It can be seen that the performance of the acoustic systems (RNN and IVC) degrades less with decreasing speech duration than the token-based approaches (PHO and LEX). For speech durations longer than 10s, the performances of the four components are quite similar. Figure 4 also shows that the large performance gain brought by combining systems holds for all speech durations.

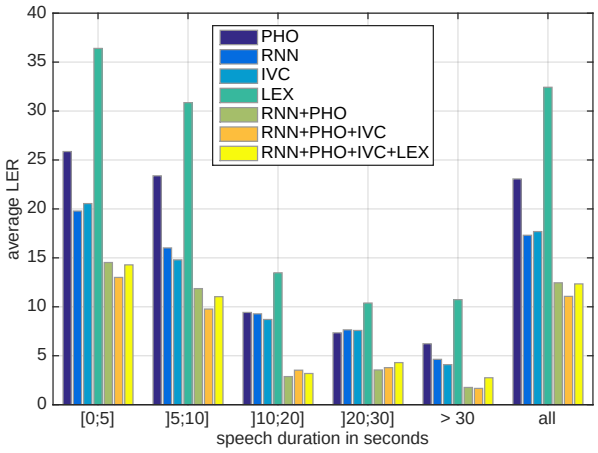

Figure 4: Average language error rate on the combined development data (1180 long and 7500 short cuts) grouped into intervals according to their speech duration.

The OpenLRE15 data are grouped into language clusters (cf. Table 1), which contain closely related languages or variants of the same language. Table 5 gives the Cavg of the best system combination (PHO+RNN+IVC) for each language cluster. The Cavg for the English, French and Slavic clusters is significantly lower than for the three other clusters. The confusions for the more difficult Arabic, Chinese and Iberian clusters (ara, zho, spa) can be seen in Figure 5 which plots the within cluster confusions for all languages. It can also be seen that within the Iberian cluster Brazilian Portuguese is easily separated from the three Spanish variants.

	ara	*zho*	*eng*	*fre*	*qsl*	*spa*
Cavg	0.12	0.14	0.03	0.001	0.06	0.12

Table 5: Cavg for each language cluster on the combined development data (8680 segments).

4.2. Evaluation results

Table 6 presents the results with the LER, EER and Cavg metrics on the official NIST evaluation data set. It can be seen that performance of the individual components and the combined systems is significantly less good than the results obtained on the development data. After analysis, we were able to identify two factors that play an important role in this degradation:

- for several variants there is a strong mismatch between the training data and the evaluation data,
- very short segments (speech duration $<$ 5s) are preponderant in the evaluation data set ($\approx 40\%$).

The phonotactic component is clearly the most robust to this mismatch, as it outperforms the other LID components. The difference in average LER is about 10% absolute less than the

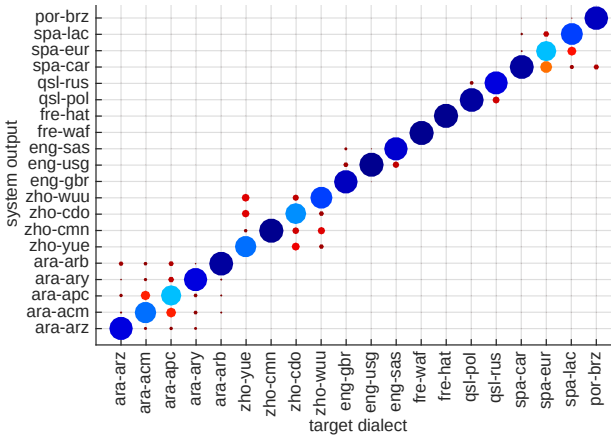

Figure 5: Confusion matrix for the development data by language cluster.

other 3 components. Combining the PHO system with the RNN or both the RNN+IVC results in a very minor decrease in LER. However, in contrast with the results on the development data, the combination of the 4 LID components is seen to improve the average LER over the phonotactic component alone.

System	*LER*	*EER*	*CAVG*
PHO	27.8	16.9	0.209
RNN	38.4	27.1	0.282
IVC	36.2	22.7	0.268
LEX	36.9	24.6	0.283
PHO+RNN	27.6	19.2	0.207
PHO+RNN+IVC	27.6	19.1	0.207
PHO+RNN+IVC+LEX	27.0	18.8	0.207

Table 6: Results on the official evaluation data with the four LID components and some combinations.

Figure 6 shows the performance of the LID components and some combinations in terms of average LER as a function of the speech duration quantiles. Speech duration appears to have less of an impact on the LER than was the case for the development data, and for the best system (PHO) the LER is about 20% even for the long segments.

Table 7 reports the Cavg of the best system combination (PHO+RNN+IVC+LEX) computed for each cluster in the evaluation set and Figure 7 shows the confusions occurring in each cluster. The Cavg and confusions increase significantly from the development results for all clusters, except Chinese and Slavic. For the other language clusters, there is a mismatch between the training and testing data. This is especially true for the French cluster for which nearly all the fre-hat files were misclassified. For the English cluster, the increase in Cavg can be mainly explained by the very small amount of training data available for the British variant, resulting in a poorer model for this variant than for the other two. In the confusion matrix it can be seen that the correct classification for eng-gbr is only about half that of the other English variants.

4.3. Post-Evaluation Results

Since there was a large mismatch between the official evaluation and training data sets, we performed an experiment where

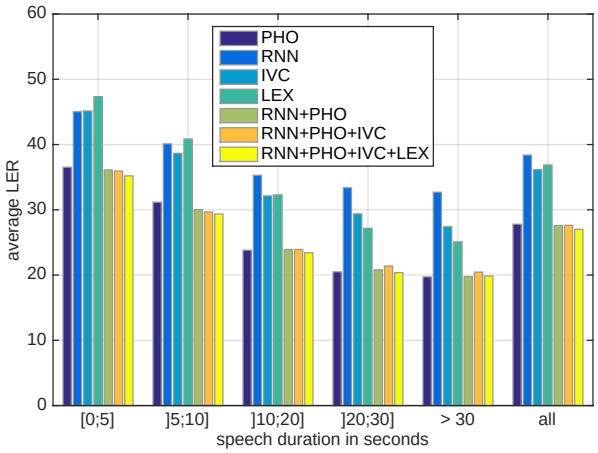

Figure 6: Average language error rate on the evaluation set grouped into intervals according to speech duration.

	ara	*zho*	*eng*	*fre*	*qsl*	*spa*
Cavg	0.23	0.14	0.12	0.51	0.03	0.21

Table 7: Cavg for each cluster on the evaluation set

10% of files of the evaluation data set were randomly selected and added to the training data. The four LID components were retrained and system performance was assessed on the remaining 90% of the evaluation data set. The results of this post-evaluation experiment, given in Table 8 demonstrate that by reducing the mismatch the evaluation data have the same general comportment observed on the development data. The different components perform quite soundly and the combining the acoustic LID components with the phonotactic one yields a 33% relative reduction in the average LER.

System	*LER*	*EER*	*CAVG*
PHO	23.5	10.1	0.151
RNN	22.8	8.4	0.146
IVC	26.6	10.4	0.174
LEX	33.9	17.6	0.247
PHO+RNN	16.2	5.7	0.100
PHO+RNN+IVC	15.5	5.4	0.095
PHO+RNN+IVC+LEX	15.6	5.6	0.098

Table 8: Post-evaluation results on 90% of the evaluation data with the four LID components and some combinations.

Figure 8 shows the impact of speech duration on the performance of the 4 LID components and some combinations in terms of average LER of the post-evaluation data subset. These results are similar to those observed on the development data (cf. Fig 4).

Table 9 gives the Cavg of the best system combination (PHO+RNN+IVC) computed for each cluster and Figure 9 shows the confusions for the different language clusters. Compared to the results on the official evaluation data, the confusions are greatly reduced for almost all the clusters. The relative improvement in Cavg is 58% for the Chinese cluster and 75% for the French cluster. The Arabic cluster, on the other hand, is the most difficult with a Cavg of 0.2 and a lot of confusions between the Egyptian and the Levantine variants.

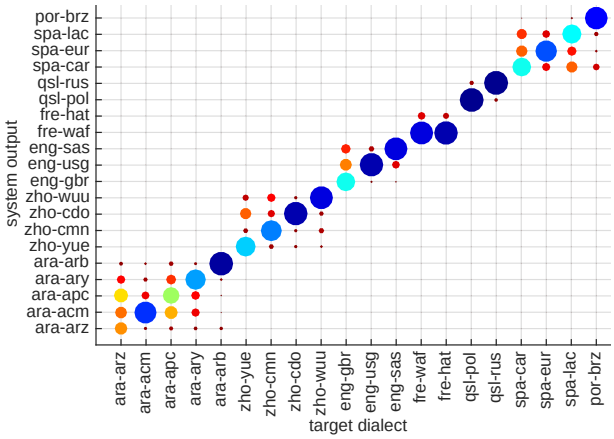

Figure 7: Confusion matrix for the evaluation data by language cluster.

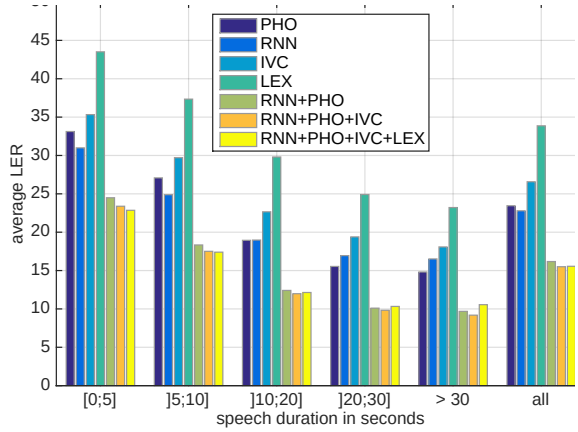

Figure 8: Average language error rate for the post-evaluation data subset grouped into intervals according to speech duration.

5. Conclusions

This paper has described the work carried out to develop a language recognition system for the 2015 NIST Language Recognition Evaluation. The language recognition system results from a fusion of four component classifiers, two acoustic-based and two token-based. One of the acoustic components uses an RNN model and the other an i-vector model. The token-based LID components aim to capture the phonotactic and lexical properties of the language. The motivation for using multiple components representing different speech knowledge is that some dialect distinctions may not be manifest at the acoustic level. Results were reported on an internally defined development set and on the official OpenLRE15 evaluation data for each LID component and some combinations. Each LID component generates language posterior probabilities optimized to maximize the LID NCE, and the combination is obtained by a simple geometric average of the posteriors.

The results on the evaluation data did not confirm those on the development set. This can be attributed to a large mismatch between the training and evaluation data, in particular for some language variants. The phonotactic system was more robust to this strong mismatch than the other LID systems and system combination did not significantly improve the evaluation results. Some post-evaluation experiments were carried out to reduce the training/evaluation data mismatch by including a small portion of the evaluation data in the LID training and testing on the remaining data. Doing so significantly improved the results, with the system combination being very effective as it had been on the development data.

	ara	*zho*	*eng*	*fre*	*qsl*	*spa*
Cavg	0.19	0.06	0.05	0.13	0.02	0.12

Table 9: Cavg for each language cluster on the post-evaluation data subset.

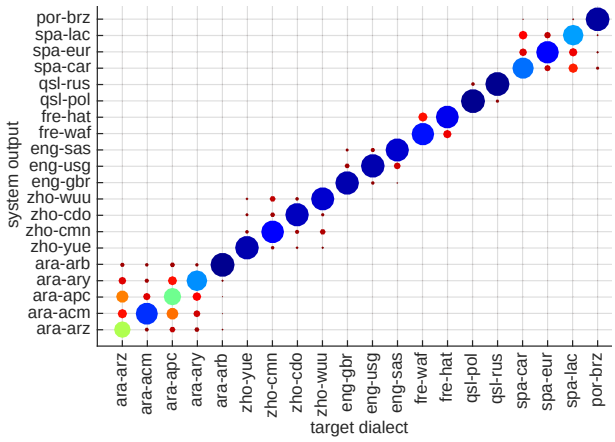

Figure 9: Confusion matrix for the post-evaluation data subset by language cluster.

6. References

[1] Lori Lamel and Jean-Luc Gauvain, "Identifying non-linguistic speech features.," in *Eurospeech*, 1993.

[2] Lori F Lamel and Jean-Luc Gauvain, "Language identification using phone-based acoustic likelihoods," in *ICASSP*, 1994.

[3] Marc A. Zissman et al., "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 31, 1996.

[4] Driss Matrouf, Martine Adda-Decker, Lori Lamel, and Jean-Luc Gauvain, "Language identification incorporating lexical information.," in *ICSLP*, 1998, vol. 98, pp. 181–184.

[5] Shubha Kadambe and James Hieronymus, "Language identification with phonological and lexical models," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 1995, pp. 3507–3510.

[6] Jean-Luc Gauvain, Abdelkhalek Messaoudi, and Holger Schwenk, "Language recognition using phone latices.," in *INTERSPEECH*, 2004.

[7] Dong Zhu and Martine Adda-Decker, "Language identification using lattice-based phonotactic and syllabotactic approaches," in *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*. IEEE, 2006, pp. 1–4.

[8] Mary P. Harper and Michael Maxwell, *Spoken Language Characterization*, springer, 2008.

[9] Mohamed Faouzi BenZeghiba, Jean-Luc Gauvain, and Lori Lamel, "Improved n-gram phonotactic models for language recognition.," in *INTERSPEECH*, 2010, pp. 2710–2713.

[10] Mohamed Faouzi BenZeghiba, Jean-Luc Gauvain, and Lori Lamel, "Phonotactic language recognition using MLP features," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012, pp. 2041–2044.

[11] Mohamed Faouzi BenZeghiba, Jean-Luc Gauvain, and Lori Lamel, "Fusing language information from diverse data sources for phonotactic language recognition.," in *Odyssey*, 2012, pp. 346–352.

[12] NIST, "The 2015 nist language recognition evaluation plan (lre15)," 2015, `http://www.itl.nist.gov/iad/mig//tests/lre/`.

[13] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *Proceedings of INTERSPEECH*, 2015.

[14] Dehak Najim, Kenny Patrick, Dehak Reda, Dumouchel Pierre, and Ouelle Pierre, "Front-End Factor Analysis for Speaker Verification," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[15] Patrick Kenny, "Bayesian speaker verification with heavy-tailed priors.," in *Odyssey*, 2010, p. 14.

[16] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems.," in *Interspeech*, 2011, pp. 249–252.

[17] David Martınez, Oldrich Plchot, Lukás Burget, Ondrej Glembek, and Pavel Matejka, "Language recognition in ivectors space," *Proceedings of Interspeech, Firenze, Italy*, pp. 861–864, 2011.

[18] Simon JD Prince and James H Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[19] Shivesh Ranjan, Gang Liu, and John H.L. Hansen, "An i-vector plda based gender identification approach for severely distorted and multilingual darpa rats data," in *ASRU*, Scottsdale, AZ, 2015, IEEE.

[20] Pedro A Torres-Carrasquillo, Elliot Singer, Mary A Kohler, Richard J Greene, Douglas A Reynolds, and John R Deller Jr, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features.," in *INTERSPEECH*, 2002.

[21] Povey Daniel and et al., "The kaldi speech recognition toolkit," in *ASRU*, Hawaii, 2011, IEEE.

[22] Gregory Gelly and Jean-Luc Gauvain, "Minimum word error training of rnn-based voice activity detection," *Interspeech*, 2015.

[23] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] Alex Graves, *Supervised sequence labelling with recurrent neural networks*, vol. 385, Springer, 2012.

[25] Alex Graves, A-R Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

[26] Paul J Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[27] Simon Funk, "Rmsprop loses to smorms3," 2015, `http://sifter.org/~simon/journal/20150420.html`.