The Sheffield language recognition system in NIST LRE 2015

Raymond W. M. Ng, Mauro Nicolao, Oscar Saz, Madina Hasan,
Bhusan Chettri, Mortaza Doulaty, Tan Lee and Thomas Hain

Department of Computer Science, University of Sheffield, United Kingdom
Department of Electronic Engineering, The Chinese University of Hong Kong
{wm.ng,m.nicolao,o.saztorralba,m.hasan,b.chettri}@sheffield.ac.uk;
{mortaza.doulaty,t.hain}@sheffield.ac.uk; tanlee@ee.cuhk.edu.hk

Abstract

The Speech and Hearing Research Group of the University of Sheffield submitted a fusion language recognition system to NIST LRE 2015. It combines three language classifiers. Two are acoustic-based, which use i–vectors and a tandem DNN language recogniser respectively. The third classifier is a phonotactic language recogniser. Two sets of training data with duration of approximately 170 and 300 hours were composed for LR training. Using the larger set of training data, the primary Sheffield LR system gives 32.44 min DCF on the official LR 2015 eval data. A post-evaluation system enhancement was carried out where i–vectors were extracted from the bottleneck features of an English DNN. The min DCF was reduced to 29.20.

1. Introduction

In a spoken language recognition (SLR) task, an automatic system is used to infer the language identity of the given acoustic signal [1]. The National Institute of Standards and Technology (NIST) has conducted a number of evaluations of automatic language recognition technology [2, 3, 4, 5]. In these evaluations, benchmark test sets were distributed and the participants were required to perform language detection, i.e. to accept or reject the language identity hypothesis for a given speech segment.

Different information from a speech signal can be used to identify the language. Classical SLR methods can be grouped by the features they use. The two most popular SLR approaches are acoustic-phonetic and phonotactic [6, 7, 8]. In the acoustic-phonetic approach, low-level acoustic features such as Mel-frequency cepstral coefficients (MFCC)[9], or shifted-delta cepstral coefficient (SDC)[10] were extracted, on which statistical models such as Gaussian mixture models are trained to model languages [11, 12]. In the phonotactic approach, an ASR-style tokeniser is needed to convert the speech signal into lattices of discrete tokens such as phonemes. The occurrence patterns of these tokens conditioned on different target languages are then modelled and language classifiers are constructed [6, 12, 13].

In recent years, the use of i-vectors [14, 15], deep neural networks [16, 17] or the combination of the two [18] became popular also for language recognition. The i-vector approach uses low-dimension latent variables which could be understood as the eigenvalues of the principle component of variations in the high-dimensional supervector space. Deep neural networks use non-linear transformation to represent structures directly related to language identity or indirect/auxiliary linguistic units such as phonemes/senones.

This paper describes the Sheffield language recognition system submitted to NIST LRE 2015. Recently NIST Language recognition evaluations were held in 2011 and 2015 [19, 20]. These evaluations focus on languages that are similar to each other and frequently mutually intelligible [20]. For NIST LRE 2015, twenty target languages in six language clusters, as shown in Table 1, were tested. Only within-cluster language detection is performed. i.e. assume the cluster identity of the speech is known, the automatic SLR is required to accept of reject a particular language in the cluster considering other competitors from the same cluster.

The Sheffield SLR system comprises three language classifiers. Two of them are acoustic and one of them is phonotactic. The acoustic system includes an i–vector based system and a DNN classifier for language recognition and are introduced in §5.1 and §5.2 respectively. The phonotactic system is introduced in §5.3. After the formal evaluation period, an extra bottleneck i–vector system was constructed and tested. It makes use of a DNN trained on conversational telephone data for English phone classification, and extract the bottleneck feature, on which the i-vector extractor was trained on. A similar setting was also shown in [18] as the best performing system with DNN phone recogniser trained on English in a NIST LRE 2009 task.

The contribution of this paper is to describe the relative performance of different language recognition techniques in NIST LRE 2015, where target languages are highly confusable compared with earlier NIST LRE data. All reported systems conformed to the Fixed Training Data condition, which is a new requirement in the 2015 evaluation. In the following, an introduction for the four systems will be given.

Table 1: Target languages in NIST LRE 2015

Cluster	Target languages
Arabic	Egyptian (ara-arz), Iraqi (ara-acm), Levantine (ara-apc), Maghrebi (ara-ary), Modern Standard (ara-arb)
English	British (eng-gbr), General American (eng-usg), Indian (eng-sas)
French	West African (fre-waf), Haitian Creole (fre-hat)
Slavic	Polish (qsl-pol), Russian (qsl-rus)
Iberian	Caribbean Spanish (spa-car), European Spanish (spa-eur), Latin American Spanish (spa-lac), Brazilian Portuguese (por-brz)
Chinese	Cantonese (zho-yue), Mandarin (zho-cmn), Min (zho-cdo), Wu (zho-wuu)

2. Data

Training and development data comes from four corpora, namely Switchboard 1, Switchboard Cellular Part 2 and two

Table 2: Speech / non–speech data distribution for VAD training

	Duration	
Dataset	(Speech)	(Non–speech)
Switchboard 1	210h	288h
VOA2	55h	61h
VOA3	93h	72h
Total	358h	421h

multi–lingual datasets (LDC2015E87, LDC2015E88) designated for LRE training [20]. The Switchboard 1 corpus comes with phone-level alignments and was used for training the voice activity detector (§3) and the tokeniser for the phonotactic language recogniser (§5.3). The two Switchboard corpora are used in the training of the universal background model (UBM) for the i–vector based language recogniser (§5.1).

Broadcast narrowband speech (BNBS) data from VOA2 and VOA3 datasets from the LRE2009 training data were used to supplement the Switchboard 1 telephone data in the training of voice activity detector.

The first multi–lingual dataset (LDC2015E87) comprises conversational telephone speech from the CallHome and CallFriend corpora in Egyptian Arabic, Standard Mandarin and US English. The second multi–lingual dataset (LDC2015E88) comprises data in seventeen other target languages in LRE 2015. The amount of data for different languages varies from 0.4 hours to 159 hours. Data is provided in NIST Sphere format. For these two datasets, no annotations of speech / non–speech or silence are provided. Voice activity detection and resegmentation is performed. By varying the resegmentation parameters, segments of different nominal durations (3, 10 or 30 seconds) were extracted for subsequent language recogniser training.

3. Voice activity detection (VAD)

Transcriptions from the Switchboard 1 corpus and the annotations from VOA2 and VOA3 datasets from LRE 2009 training data were used to derive speech / non–speech labels. Switchboard 1 data represented CTS data. VOA2 and VOA3 data represented broadcast narrowband speech (BNBS) data. A deep neural network (DNN) for speech / non–speech classification was trained on these data. The total amount of speech and non–speech in the dataset are listed in Table 2.

The input to the DNN are filterbank features of 23 dimensions with a context window of 15 frames on both sides, and a DCT to reduce dimensionality to 368. 2 hidden layers of 1000 nodes were used and the output layer consisted of 2 nodes only, for speech and non–speech. A frame–wise cross–entropy criterion was used as target function in DNN training. For an input audio file this DNN provided the estimated values of the posterior probabilities of speech or non–speech for each frame. A two–state HMM, with a minimum state duration of 20 frames, was used to smooth the sequence of posteriors to a sequence of speech segments. Speech segments were merged when the intermediate silence was shorter than 2 seconds in order to produce chunks of speech which are long enough and linguistically meaningful.

To validate the effectiveness of the VAD, it was applied on a held–out test set from Switchboard 1. The segmentation error rates provided by the VAD, before segment merging, had a detection (of speech frames) miss rate of 2.21% and a false alarm rate of 2.63%.

Table 3: V1 data amount across languages (hours)

Language	TRAIN	DEV	HELDOUT
ara-acm	11.4	2.6	1.6
ara-apc	11.5	2.5	1.6
ara-arb	2.8	0.4	0.4
ara-ary	8.6	1.6	1.4
ara-arz	23.6	4.2	3.1
eng-gbr	0.4	0.05	0.05
eng-sas	2.0	0.3	1.9
eng-usg	36.6	4.4	4.6
fre-hat	2.1	0.3	0.3
fre-waf	0.9	0.2	0.03
por-brz	0.5	0.03	0.05
qsl-pol	8.8	5.3	5.5
qsl-rus	3.2	3.7	3.5
spa-car	9.7	1.2	0.8
spa-eur	1.5	0.2	0.04
spa-lac	2.2	0.3	0.2
zho-cdo	1.7	0.5	0.04
zho-cmn	18.3	3.7	3.9
zho-wuu	1.5	0.3	0.08
zho-yue	0.5	0.1	0.1

Table 4: V3 (30–second) data amount across languages (hours)

Language	TRAIN	DEV	HELDOUT
ara-acm	23.0	4.8	3.8
ara-apc	26.0	5.4	3.6
ara-arb	2.8	0.4	0.4
ara-ary	21.3	4.3	3.4
ara-arz	68.1	8.6	7.0
eng-gbr	0.4	0.0	0.1
eng-sas	2.0	0.3	3.8
eng-usg	75.8	9.4	9.7
fre-hat	2.1	0.3	0.2
fre-waf	2.1	0.5	0.2
por-brz	0.5	0.0	0.1
qsl-pol	12.6	8.0	6.3
qsl-rus	3.6	5.9	6.0
spa-car	20.4	2.7	1.7
spa-eur	2.5	0.3	0.1
spa-lac	3.1	0.5	0.3
zho-cdo	3.1	0.6	0.3
zho-cmn	46.9	8.7	9.1
zho-wuu	3.2	0.5	0.3
zho-yue	1.0	0.2	0.2

4. Training and development data

There was no officially defined development data for LRE 2015. For each of the 20 training languages, the first 80% of the files (alphabetically sorted) were taken as training (TRAIN), the subsequent 10% as development data (DEV), which is used for the training of system fusion parameters. The final 10% was the held–out data for evaluating system performance (HELDOUT). In case the training of a single system needs development data, 10% of the files from TRAIN was selected for such purpose.

4.1. V1 training data

VAD, together with the minimum speech duration heuristics and linking of consecutive speech segments, was applied on the multi–lingual training data. This resulted in speech segments of different durations. This data formed the V1 training data set, with speech segments with a nominal duration of 30 seconds. Speech segments with a duration between 20 and 45 seconds were selected and the remaining part of the data was truncated. The total duration of TRAIN is 147.8 hours (speech: 115.1 hours, non-speech: 32.7 hours). Details on the data amount for different languages in the V1 data set are shown in Table 3.

4.2. V3 training data

The V1 training data was created with a tight filtering criterion, yielding only 147.8 hours of training data selected from over 700 hours of raw acoustic data. Hence, another segmentation trial was conducted in an attempt to obtain more training data. The V1 data set was first taken as the seed data. The data not selected into the V1 dataset were returned as long segments. All data was decoded by an English phone tokeniser (§5.3). A maximum silence duration threshold was imposed. The data was resegmented to give short segments, from which data selection was conducted to obtain candidate segments of nominal durations of 3, 10 or 30 seconds ,for both training and development.

Table 4 shows the amount of data for the V3 30s data set. V3 TRAIN contains 320.5 hours of data, which is 172.7 hours more compared to V1 data. We also prepared TRAIN, DEV and HELDOUT in different nominal durations (3sec, 10sec) by using different resegmentation and segment selection criteria. The amount of TRAIN data for 3 seconds and 10 seconds are 308.4 hours and 262.0 hours, respectively.

5. Language recogniser systems

In the following, the two acoustic LR systems (the i–vector language recogniser and the DNN language recogniser) and the phonotactic language recogniser will be introduced. System structure and results on V1 and V3 DEV and HELDOUT, and the processing time of the most significant system components will be reported. All scoring was based on the within–cluster C_{avg} metric described in [20]. However, for each classifier an optimal detection threshold was assumed (either a single optimal threshold across all languages or multiple thresholds which are language-dependent). The reported metric is the minimum detection cost function (min DCF), which is C_{avg} at the optimal operating point. All processing times reported are equivalent wall–clock time on a single Intel Xeon E5 grade CPU running in hyperthreaded mode.

5.1. i–vector language recogniser

The i–vector based language recogniser used a universal background model (UBM) and a total variability matrix (T) trained on 884 hours of data. This included all provided training data for LRE 2015 with Switchboard 1 and Switchboard Cellular (502 hours), CallHome, CallFriend (LDC2015E87) and the provided multi–lingual data (LDC2015E88) for LR training (382 hours).

Frame–level VAD based on thresholding the log MEL energy was first performed and a 256-component Gaussian mixture model with diagonal covariance was trained as the universal background model (UBM). The UBM was then used to compute vocal tract length normalisation (VTLN) on the Mel–frequency cepstral coefficient (MFCC) features for each utterance. A second-pass feature extraction was carried out with frequency warping, shifted delta cepstral coefficient with the standard $7-1-3-7$ configuration (resulting in $7\times(7+1) = 56$-dimensional feature), and mean normalisation and voiced frame selection, on which the UBM training were performed again. UBM training was repeated with full covariance and the number of mixtures was increased to 2048. For each training utterance, MAP adaptation from the UBM was performed to derive an utterance–specific GMM. The GMM means are concatenated to form supervectors of dimension $56 \times 2048 = 114688$. Finally, a total variability matrix was trained to project the supervectors to a reduced space with 600 dimensions [21].

With the extracted i–vectors, support vector machines and logistic regression models were trained and used as language classifiers. The support vector machines used linear kernels. The classifiers were configured either as within–cluster or global. With the total varaibility matrix and i–vector extraction algorithm remained unchanged, the within–cluster classifiers select in-class i–vectors from the training utterances belonging to the target language and out-of-class training i–vectors only from the training utterances spoken in other languages within the language cluster. The global classifiers were 1–versus–19 binary language classifiers. Table 5 summarises the results. Looking into the test data with SVM model applied on V1 data, it can be seen that the within–cluster model gave better performance compared to the global model. Augmenting the UBM and total variability training data to full data set (884 hours) further reduced min DCF. Logistic regression generally performed better than SVM. An increase of logistic regression training data further improved the performance.

For V3 data, logistic regression again outperformed SVMs. Among different training duration, logistic regression models trained on 321 hours of V3 30s data (matched duration) performed the best. The within–cluster model further reduced min DCF. Finally, the experiments were repeated with V3 10–second and 3–second data and the results were shown The minDCF on HELDOUT data for the three durations are 6.09%, 15.90% and 20.51% respectively.

5.2. DNN language recogniser

Intuitively, a single-pass DNN language classifier can be used to directly mapped the input features to the output layer which was a 20–dimensional one–hot vector denoting the language identity. Another approach is a two-pass DNN language classifier. A first-pass DNN, possibly trained for a different purpose (e.g. phone recognition), is used to extract data (in terms of posterior probabilities, bottleneck, etc.). These data is then used to train a secondary classifier for languages [17].

After an initial evaluation of both methods, the two-pass setting was preferred, since the single-pass DNN model was showing a decrease in performance due to its overfitting to the vocal tract characteristics of the training speakers. In the two-pass appraoch, the first-pass DNN was an English phone recogniser DNN. This DNN was used to extract bottleneck features from the multi–lingual input. It followed the setting described in §5.3, except that the DNN did not include speaker MLLR transform and sequence training was conducted. This design was aimed to create different outputs compared with the phonotactic system and to maximise complementary effects during system combination.

The bottleneck features were fed to a secondary DNN for language classification. The dimension of the bottleneck feature was 64. Before feeding into the secondary DNN for language recognition, feature splicing with 12 left and right contexts were carried out. Discrete cosine transform (DCT) was applied to project the temporal sequence into 9–dimensional feature vectors. The final structure of the secondary DNN was $576 \times 750 \times 750 \times 750 \times 750 \times 20$. The initial learning rate is 0.0001 and the new–bob learning rate was applied. Two DNNs were trained, for the $V1$ and $V3$ data distributions respectively. For the 163 hours of training data in $V1$, the total training time was 5 hours; while for the 350 hours of $V3$, the total training time was 11 hours.

The DNN language classifier outputted frame–based poste-

Table 5: i–vector based LR results with different model settings

UBM & Total variability matrix training	SVM / LogReg (Train duration)	Within–cluster classifier	min DCF (%) DEV	 HELDOUT
	(Test on V1 30-second data)			
V1 30s train (148h)	SVM (148h)	No	6.34	10.75
V1 30s train (148h)	SVM (148h)	Yes	3.35	6.35
Switchboard + Unsegmented LR train (884h)	SVM (148h)	Yes	3.12	6.00
Switchboard + Unsegmented LR train (884h)	LogReg (148h)	No	2.52	4.54
Switchboard + Unsegmented LR train (884h)	LogReg (884h)	No	1.82	4.42
	(Test on V3 30-second data)			
Switchboard + Unsegmented LR train (884h)	SVM (321h)	Yes	4.51	7.90
Switchboard + Unsegmented LR train (884h)	LogReg (148h)	No	4.79	7.74
Switchboard + Unsegmented LR train (884h)	LogReg (887h)	No	4.24	7.48
Switchboard + Unsegmented LR train (884h)	LogReg (321h)	No	3.87	6.78
Switchboard + Unsegmented LR train (884h)	LogReg (321h)	Yes	2.88	6.09
	(Test on V3 10-second data)			
Switchboard + Unsegmented LR train (884h)	LogReg (10s train, 262h)	Yes	7.71	15.90
	(Test on V3 3-second data)			
Switchboard + Unsegmented LR train (884h)	LogReg (3s train,308h)	Yes	14.08	20.51

rior probability of languages. The posterior probability was the scaled product of the language likelihood and the prior probability. Prior normalisation was performed by calculating the number of frames used for training in a given language divided by the total number of frames used for training. The likelihood of each of the 20 languages was then obtained by dividing the DNN posterior by the prior probability of the language. For a given audio file or segment, the score assigned to each of the 20 languages was the average likelihood extracted from the DNN. For this calculation, only areas which have been considered as speech by the VAD described in §3 were used.

Table 6 shows the results of the DNN language recogniser in different dataset. Min DCF for 30–second data is 13.91% and 15.96% for V1 data; and 15.44% and 18.07% for V3 data. Error rate decreases with duration.

5.3. Phonotactic language recogniser

The phonotactic language recogniser contained a phone tokeniser built on switchboard 1 data and a support vector machine language classifier based on position–dependent phone trigram tf–idf statistics [22]. The tokeniser followed a feedforward DNN hybrid setting with 6 hidden layers each having 2048 neurons followed by a bottleneck layer with 64 neurons and an output layer with 3815 neurons (number of senones). The input features of the DNN were mel–frequency cepstral coefficient (MFCC) features with deltas and normalisation, followed by global feature transform with linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT) and feature splicing with 5 contextual frames on the left and the right. The training targets were the senone alignment results from a constrained maximum likelihood linear regression (CMLLR) adapted, maximum mutual information (MMI)–trained acoustic model.

Table 6: DNN language recogniser results

	minDCR (%)	
Test data	DEV	HELDOUT
V1 30s	13.91	15.96
V1 10s	15.69	18.74
V1 3s	18.11	21.55
V3 30s	15.44	18.07
V3 10s	N/A	N/A
V3 3s	18.82	21.71

The English tokeniser was applied on the multi–lingual training data. WFST-based decoding was implemented and a di-phone language model (trained on Switchboard 1 data) was used to achieve minimum phonotactic constraints on the phone tokeniser outputs. Then utterance–based phone n–gram occurrence statistics were computed, from which term frequency (tf) and inverse document frequency (idf) were derived. idf was computed across the full multi–lingual training set. tf–idf vector was constructed for each utterance, on which 20 binary classifiers were trained for the 20 target languages. In the training for each classifier, positive training vectors comprised all training utterances belonging to the target language. Negative training vectors consisted of only the training utterances within the language cluster. During testing, idf was inherited from TRAIN and the likelihood for each of the 20 languages were computed.

Four different aspects of the phonotactic LR classifier were investigated. In every aspect, two conditions were tried, leading to $2^4 = 16$ different tokeniser settings. These trial settings cover: tokenisers with and without utterance–based MLLR adaptation; tf–idf statistics on phone bigrams ($n = 2$) and trigrams ($n = 3$); DNN with and without discriminative training (sequence trained) on the English switchboard data; and different di-phone language model scales in the WFST decoding of the tokeniser.

The full list of results for 30s V1 data is shown in Table 7. Tokenisers with MLLR adaptation behaved slightly better than without MLLR. Trigram tf–idf statistics were more useful than bigram ones. For MLLR models, sequence training on the English data did not give a better tokeniser in the multi–lingual setting. There was no consistent performance gain on using either LM scale factor in the English tokeniser. However, LM scale factor 2 was believed to be small enough for this task. Based on these results, tokeniser with MLLR adaptation, trigram tf–idf, no sequence training and LM scale factor equal to 2 was chosen. In terms of the processing speed, tokeniser GMM–HMM model training took 15.29 hours. DNN training took 47.70 hours. Trigram tf–idf statistics gathering took 172.49 hours.

We also tried the global classifier setting where 1–versus–19 binary classifiers were trained for all languages. The resulting min DCF were $2 - 5\%$ absolute higher. On V1 30s data, the

Table 7: Phonotactic LR results with different tokeniser settings

Use	N–gram	Sequence	LM	min DCF (%)	
MLLR	order	trained	scale	DEV	HELDOUT
[V1 30s]					
Yes	3	No	0.5	4.6	7.0
Yes	3	No	2	4.3	9.0
Yes	3	Yes	0.5	6.3	9.4
Yes	3	Yes	2	4.8	9.8
Yes	2	No	0.5	5.2	11.1
Yes	2	No	2	5.5	9.8
Yes	2	Yes	0.5	7.0	11.0
Yes	2	Yes	2	5.4	10.7
No	3	No	0.5	4.8	11.9
No	3	No	2	5.1	11.6
No	3	Yes	0.5	4.7	11.1
No	3	Yes	2	5.0	9.8
No	2	No	0.5	5.3	11.3
No	2	No	2	6.5	11.9
No	2	Yes	0.5	5.4	9.7
No	2	Yes	2	5.7	10.3
[V3 30s]					
Yes	3	No	2	7.38	11.30

chosen tokeniser gave min DCF of 4.3% and 9.0% on DEV and HELDOUT data. Replicating the same experiment on V3 data gave 7.38% and 11.30% for DEV and HELDOUT 30–second data. The whole experiment was replicated on 10–second and 3–second data (training and testing). Training and testing with different duration combination were tried. It was found that the 30–second model gave the best results on test data in different durations.

Table 8: LR results for standalone systems with Gaussian backend score calibration

	min DCF (%)			
	(No calibration)		(GMM backend)	
System	DEV	HELDOUT	DEV	HELDOUT
[Multiple thresholds]				
i-vector	2.88	6.09	4.88	9.09
DNN	15.44	18.07	13.65	16.30
Phonotactic	7.38	11.30	6.90	11.65
[Global threshold]				
i-vector	6.29	12.48	11.89	20.17
DNN	24.43	26.54	18.28	22.50
Phonotactic	24.96	29.51	16.08	22.00

6. Primary system

The primary system submission to NIST LRE 2015 comprises three systems introduced above. Before system combination, Gaussian backend score calibration was conducted for the acoustic DNN and the phonotactic language classifier [23]. For each target language, a Gaussian mixture model (GMM) with 4 components was trained, with the maximum likelihood criterion, on the multi-dimensional scores resulted from decoding the TRAIN dataset. During decoding, the likelihood of each language-dependent GMMs was computed. Table 8 shows the LR results with Gaussian backend for the three component systems. The rows under the [Multiple thresholds] section include the language detection results with language-dependent detection threshold (§5). Gaussian backend gave no improvements for the i–vector based system, but consistent gains to the other two systems were observed. To make the Gaussian backend work for the i–vector scores, different parameterisations (e.g. increased mixture size), or further scaling of scores might be needed.

Table 9: System fusion results

	min DCF (%)					
	[Multiple thresholds]			[Global threshold]		
System	DEV	HELDOUT	EVAL	DEV	HELDOUT	EVAL
[30s]						
i-vector	2.88	6.09	20.90	3.73	10.21	27.51
DNN	10.79	14.59	31.50	12.79	19.97	37.68
phonotactic	6.69	12.53	25.84	11.72	19.21	31.19
3-sys fusion	2.17	5.76	21.79	3.14	9.42	27.90
[10s]						
i-vector	6.87	12.23	28.49	8.44	17.38	34.22
DNN	12.83	16.34	33.41	15.47	21.53	40.42
phonotactic	14.96	23.85	29.83	19.18	27.43	36.07
3-sys fusion	4.95	10.09	27.11	5.59	13.83	32.92
[3s]						
i-vector	10.92	17.2	32.12	11.50	22.83	36.61
DNN	14.80	17.13	36.43	16.08	21.81	41.98
phonotactic	25.95	31.10	37.99	31.49	36.11	41.63
3-sys fusion	9.20	13.38	31.40	9.69	17.70	36.67
[overall]						
i-vector	–	–	28.56	–	–	32.92
DNN	–	–	34.83	–	–	40.16
phonotactic	–	–	33.72	–	–	36.93
3-sys fusion	–	–	28.29	–	–	32.44

Table 10: Post-evalution system enhancement

	min DCF (%)					
	[Multiple thresholds]			[Global threshold]		
System	DEV	HELDOUT	EVAL	DEV	HELDOUT	EVAL
[30s]						
3-sys fusion	2.17	5.76	21.79	3.14	9.42	27.90
BN + IV	2.24	5.13	19.00	2.80	10.84	25.75
4-sys fusion	1.17	5.05	19.94	2.00	8.87	25.25
[10s]						
3-sys fusion	4.95	10.09	27.11	5.59	13.83	32.92
BN + IV	5.10	9.06	23.37	7.01	14.85	29.78
4-sys fusion	3.03	7.26	23.05	3.60	11.63	28.95
[3s]						
3-sys fusion	9.20	13.38	31.40	9.69	17.70	36.67
BN + IV	9.29	13.69	27.55	10.05	18.47	32.55
4-sys fusion	6.74	10.97	27.22	7.18	15.53	33.26
[overall]						
3-sys fusion	–	–	28.29	–	–	32.44
BN + IV	–	–	25.14	–	–	29.56
4-sys fusion	–	–	24.69	–	–	29.20

We also looked at the performance of min DCF under a global threshold. Compared with the multiple–threshold performance above, Gaussian backend gave bigger relative reductions of min DCF in the DNN and the phonotactic systems.

After score calibration, the single system scores were converted to log likelihood ratios and DEV data was used to derive a linear weight for system combination with respect to minimum detection cost. System fusion trials were carried out independently for the six language clusters and the three nominal durations (3 seconds, 10 seconds and 30 seconds).

The performance of the standalone i–vector, DNN, phonotactic LR systems and the 3-system fusion results are listed in Table 9 in terms of min DCF, both in single and multiple thresholds. The results on DEV are oracle results as the calibration algorithm were tuned on the same data. The first three blocks denotes the results on 30-sec, 10-sec and 3-sec respectively and the bottom block is the overall result. Considering standalone system results, the i–vector system was consistently giving better results except for 3-second HELDOUT data. The DNN system performance was relatively invariant to duration. The phonotactic system degraded substantially when the duration of test segments decrease. Across different data set, the training-

Table 11: Pairwise system fusion results (min DCF with Global threshold). The upper right triangular blocks show HELDOUT results. The lower left blocks show EVAL results

30-sec	BN + IV	i–vector	phonotactic	DNN
BN + IV		8.16	9.70	10.39
i–vector	24.27		9.51	10.16
phonotactic	25.34	27.48		17.03
DNN	26.41	28.55	31.99	
10-sec	BN + IV	i–vector	phonotactic	DNN
BN + IV		13.70	13.67	13.30
i–vector	28.24		16.68	18.43
phonotactic	28.18	32.42		18.69
DNN	31.36	34.72	35.96	
3-sec	BN + IV	i–vector	phonotactic	DNN
BN + IV		18.68	18.04	16.94
i–vector	32.50		21.33	18.27
phonotactic	32.41	35.58		19.97
DNN	34.49	37.50	40.38	

testing mismatch causes a significant DCF increase from DEV to HELDOUT and further in EVAL. The overall DCF with EVAL is 32.44%.

7. System enhancement

After the official LRE 2015 evaluation, a system enhancement was carried out. The 64-dimension bottleneck features were extracted from the English DNN (§5.3). They substituted the frequency warped shifted delta cepstral coefficients in the i–vector system (§5.1), on which total variability matrix was re-trained. I–vectors for bottlenecks in the same dimension were extracted and logistic regression language recognition was retrained. This setting was parallel with one of the optimal settings with English DNN described in [18]. Our reported results aim at cross-validating the robustness of the bottleneck i–vector setting, in the noisy condition in NIST LRE 2015.

Table 10 shows the results of the bottleneck i–vector system (BN + IV) for DEV, HELDOUT and EVAL. Compared to the official 3-component fusion system (3-sys), the bottleneck i–vector system did not show consistent improvements in all DEV and HELDOUT data. Nevertheless, bottleneck i–vector system demonstrates lower min DCF for all EVAL data set.

Score calibration and fusion were performed on all four component systems (i.e. the three component systems in the official submission and the bottleneck i–vector system described above) using the identical method as described in §6. The 4–system fusion demonstrated the best performance in almost all data set. The single threshold min DCF for overall EVAL data is 29.20%, which was a relative improvements of 9.99% compared with the 3-system fusion system.

To compare the relative contributions of the 4 systems, pairwise system fusion among the 4 systems were carried out on HELDOUT and EVAL data. Table 11 shows the results. According to Table 9 and 10, standalone systems were ranked according to their performance in the order of: (1) bottleneck i–vector (BN + IV) (2) i–vector (3) phonotactic and (4) DNN. For any given system, pairwise system fusion with a better system generally gave better results. (i.e. (2)+(3) is better than (2)+(4)). This is particularly true on EVAL data. The only exceptions were found in system fusion between the BN + IV and the phonotactic systems on 3-second and 10-second data, where (1)+(3) was marginally better than (1)+(2). This might be explained by the stronger complementary effects between the bottleneck i–vector and the phonotactic systems, which made the final fusion system outperform the fusion of the best two i–vector based systems.

8. Summary

In this paper, the Sheffield submission system to NIST Language Recognition Evaluation 2015 was presented. The system comprises three LR classifiers. Two are acoustic-based, which used i–vectors and a tandem DNN language recogniser respectively. The third classifier is a phonotactic language recogniser. A post-evaluation system enhancement was carried out where i–vectors were extracted from the bottleneck features of an English DNN. Across four system settings, the i–vector and the bottleneck i–vector system demonstrated the best performance. System fusion brought further improvements. The performance of the tandem DNN classifier and the phonotactic classifier substantially degraded in 30-second and 3-second test segments respectively. In future study, variability compensation of channel variety and robustness of the phonotactic LR classifier can be enhanced.

9. Acknowledgement

This work was supported by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

10. References

[1] Y. K. Muthusamy, E. Barnard, and R. A. Cole, “Reviewing automatic language identification,” *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 33–41, Oct. 1994.

[2] “The 1996 NIST language recognition evaluation plan (LRE96),” `http://www.itl.nist.gov/iad/mig/tests/lre/1996/LRE96EvalPlan.pdf`.

[3] “The 2003 NIST language recognition evaluation plan (LRE03),” `http://www.itl.nist.gov/iad/mig/tests/lre/2003/LRE03EvalPlan-v1.pdf`.

[4] “The 2005 NIST language recognition evaluation plan (LRE05),” `http://www.itl.nist.gov/iad/mig/tests/lre/2005/LRE05EvalPlan-v5-2.pdf`.

[5] “The 2007 NIST language recognition evaluation plan (LRE07),” `http://www.itl.nist.gov/iad/mig/tests/lre/2007/LRE07EvalPlan-v8b.pdf`.

[6] M. A. Zissman, “Comparison of four approaches to automatic language identification of telephone speech,” *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, pp. 31–44, January 1996.

[7] E. Ambikairajah, Haizhou Li, Liang Wang, Bo Yin, and V. Sethu, “Language identification: A tutorial,” *Circuits and Systems Magazine, IEEE*, vol. 11, no. 2, pp. 82 –108, secondquarter 2011.

[8] Haizhou Li, Bin Ma, and Kong Aik Lee, “Spoken language recognition: From fundamentals to practice,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1136–1159, May 2013.

[9] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” *Acoustics, Speech and*

Signal Processing, IEEE Transactions on, vol. 28, no. 4, pp. 357–366, Aug 1980.

[10] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. Reynolds, and J. J. R. Deller, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proc. ICSLP*, 2002.

[11] M. A. Zissman, "Automatic language identification using Gaussian mixture and hidden Markov models," in *Proc. ICASSP*, 1993, vol. II, pp. 399–402.

[12] E. Singer, P. A. Torres-Carrasquillo, T. P. Gleason, W. M. Campbell, and D. A. Reynolds, "Acoustic, phonetic and discriminative approaches to automatic language recognition," in *Proc. Eurospeech*, 2003, pp. 1345–1348.

[13] Timothy J. Hazen and Victor W. Zue, "Segment-based automatic language identification," *J. Acoust. Soc. Am.*, vol. 101, no. 4, pp. 2324–2331, Apr. 1997.

[14] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via Ivectors and dimensionality reduction," in *Proc. Interspeech*, 2011, pp. 857–860.

[15] D. Martinez, L. Burget, L. Ferrer, and N. Scheffer, "Ivector-based prosodic system for language identification," in *Proc. ICASSP*, 2012, pp. 4861–4864.

[16] Luciana Ferrer, Yun Lei, Mitchell McLaren, and Nicolas Scheffer, "Spoken language recognition based on senone posteriors," in *Proc. Interspeech*, 2014, pp. 2150–2154.

[17] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *Signal Processing Letters, IEEE*, vol. 22, no. 10, pp. 1671–1675, Oct 2015.

[18] Luciana Ferrer, Yun Lei, Mitchell McLaren, and Nicolas Scheffer, "Study of senone-based deep neural network approaches for spoken language recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 24, no. 1, pp. 105–116, Jan. 2016.

[19] "The 2011 NIST language recognition evaluation plan (lre11)," 2011.

[20] "The 2015 NIST language recognition evaluation plan (lre15)," 2015.

[21] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via Ivectors and dimensionality reduction," in *Proc. Interspeech*, 2011.

[22] H. Li, B. Ma, and C.-H. Lee, "A vector space modelling approach to spoken language identification," *IEEE Trans. Audio, Speech, Lang. Prcs.*, vol. 15, no. 1, 2007.

[23] Niko Brummer, "Focal toolkit for evaluation, fusion and calibration of statistical pattern recognisers," 2010.