The MIT Lincoln Laboratory / JHU / EPITA-LSE LRE17 System

Fred Richardson[1], *Pedro A. Torres-Carrasquillo*[1], *Jonas Borgstrom*[1],
Douglas Sturim[1], *Youngjune Gwon*[1], *Jesús Villalba*[2],
Jan Trmal[2], *Nanxin Chen*[2], *Réda Dehak*[3], *Najim Dehak*[2]

[1]MIT Lincoln Laboratory, Lexington, MA USA
[2]Johns Hopkins University, Baltimore, MD USA
[3]LSE-EPITA, Villejuif FR

frichard@ll.mit.edu, ptorres@ll.mit.edu, jonas.borgstrom@ll.mit.edu,
sturim@ll.mit.edu, gyj@ll.mit.edu, jesus.antonio.villalba@gmail.com,
jtrmal@gmail.com, bobchennan@gmail.com, reda.dehak@gmail.com, najim.dehak@gmail.com

Abstract

Competitive international language recognition evaluations have been hosted by NIST for over two decades. This paper describes the MIT Lincoln Laboratory (MITLL) and Johns Hopkins University (JHU) submission for the recent 2017 NIST language recognition evaluation (LRE17) [1]. The MITLL/JHU LRE17 submission represents a collaboration between researchers at MITLL and JHU with multiple sub-systems reflecting a range of language recognition technologies including traditional MFCC/SDC i-vector systems, deep neural network (DNN) bottleneck feature based i-vector systems, state-of-the-art DNN x-vector systems and a sparse coding system. Each sub-systems uses the same backend processing for domain adaptation and score calibration. Multiple sub-systems were fused using a simple logistic regression ([2]) to create system combinations. The MITLL/JHU submissions were selected based on the top ranking combinations of up to 5 sub-systems using development data provided by NIST. The MITLL/JHU primary submitted systems attained a C_{avg} of 0.181 and 0.163 for the fixed and open conditions respectively. Post evaluation analysis revealed the importance of carefully partitioning for the development data, using augmented training data and using a condition dependent backend. Addressing these issues - including retraining the x-vector system with augmented data - yielded gains in performance of over 17%: a C_{avg} of 0.149 for the fixed condition and 0.132 for the open condition.

Index Terms: language recognition NIST LRE17

1. Introduction

Language recognition is a closed set identification task and requires automatic systems to produce scores that reflect the likelihood of a given waveform containing speech from one of a fixed set of known language classes (which can include an out-of-set class). The NIST language recognition evaluations (LREs) have focused on progressively more difficult challenges over the years with the most recent 2017 LRE (LRE17) [1] focusing on performance in a challenging domain with limited prior domain data. For LRE17 a substantial amount of mismatched out-of-domain data and a limited amount of in-domain data was made available by NIST for each language prior to the evaluation. Performing well on the evaluation required judicious use of the limited amount of in-domain data for domain adaptation and for system development and evaluation.

The LRE17 as specified by NIST consisted of a two tasks: a required "fixed" condition task an an optional "open" condition task. There are 14 language classes in the evaluation and for the fixed condition NIST specifies prior telephony training data from the Linguistic Data Consortium (LDC) that can be used to train models for each class (see [1] for details). For the open condition there are no restrictions on the data that can be used for training. The evaluation data consists of both narrow band telephony data (referred to as MLS14) and wide band video data (referred to as VAST). The only wide-bandwidth data available for system development prior to the evaluation was included in the limited amount of development data provided by NIST.

The MITLL/JHU LRE17 submission was a collaboration between MIT Lincoln Laboratory (MITLL), Johns Hopkins University - Center for Speech and Language Processing (JHU) and LSE-EPITA Systems Laboratory (EPITA). Several different types of sub-systems were developed for the evaluation including several i-vector systems [3], a sparse coding system [4] and more recent state-of-the-art x-vector systems [5]. The i-vector systems were trained with different types of time varying feature representations including traditional shifted-delta cepstra (SDCs) [6] and deep neural network (DNN) based bottleneck features (BNFs) [7] both of which have been shown to perform well in past NIST LREs [8, 9]. The x-vector system is a more recent approach that also uses a DNN but unlike i-vector systems the DNN is used to extract a vector representation for an entire speech utterance [5].

The remainder of this document describes the MITLL/JHU primary and secondary system submissions for NIST 2017 LRE. Section...

2. The NIST LRE17 Task

NIST specified two data sets for LRE17 that were available prior to the evaluation: a narrow-band telephony training data set (TRN17) built from prior NIST evaluations and other LDC corpora with over 2000 hours of audio data and a development data set (DEV17) with over 60 hours of narrow and wide band audio data. Using statistics derived from a Gaussian mixture model based speech activity detection system (GMMSAD), the

This work was sponsored by the Department of Defense under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

Egyptian Arabic	American English
Iraqi Arabic	Polish
Levantine Arabic	Russian
Maghrebi Arabic	Caribbean Spanish
Chinese Mandarin	European Spanish
Chinese Min Nan	Latin American Continental Spanish
British English	Brazilian Portuguese

Table 1: LRE17 language classes

total amount of speech data is roughly 1000 hours for TRN17 and 40 hours for DEV17. The amount of data available for each language in TRN17 varies widely with Brazilian Portuguese, Chinese Min Nan and British English having between 2 and 8 hours of data while U. S. English, Chinese Mandarin and Levantine Arabic have 150 to 230 hours of data. The total set of 14 languages for LRE17 is given in Table 1.

DEV17 generally has a lower diversity in language class durations than TRN17. The MLS14 narrow-band telephony development data is partitioned into three duration "bins" of 3, 10 and 30 seconds to match prior NIST LREs while the wide-band VAST data uses the full duration of the original source video file. The duration for each language after GMMSAD ranges from 0.3 to 1.3 hours for MLS14 and 0.4 to 3.0 hours for VAST. The relatively small amount of DEV17 data per a language poses a challenge when developing adaptation techniques and assessing system performance without attaining overly optimistic results due to over-fitting.

The LRE17 evaluation data (EVAL17) is similar to the DEV17 data in that it is comprised of both MLS14 and VAST data for all 14 languages but EVA17 is much larger (over 200 hours). The LRE17 evaluation plan [1] specifies that each EVAL17 file must be processed independently without using data or information derived form the other EVAL17 files. Similar to prior NIST LREs, LRE17 uses a decision cost function (DCF) that computes the weighted combination of two detection based statistics. The two statistics, the false alarm and missed detection rates, are computed using the likelihood scores produced by a system for each evaluation file and each language class. Log likelihood ratios are computed treating each language class as a target and all other classes as a non-target. Prior target probabilities and costs are then used both to compute a log likelihood ratio threshold for determining target hits and to compute the weights used to combine the false alarm and missed detection rates for the CDF. Unlike prior NIST LREs, the LRE17 performance metric (C_{avg}) takes the average of the DCF computed at two target prior operating points, 0.5 and 0.1, using equal costs for both (for more information on LRE17 scoring see [1]).

The LRE17 fixed condition limits system development to use only the data in the TRN17 and DEV17 sets as well as the LDC Switchboard [10] and Fisher [11] English telephony corpora (including word-level transcripts). A fixed condition submission is required for any site participating in LRE17. Systems developed for the optional open condition can use data from any other source so long as the data is clearly described by the participating site. The MITLL/JHU open systems were developed using 2000 hours (spanning 24 languages) of IARPA Babel data listed in Table 6.

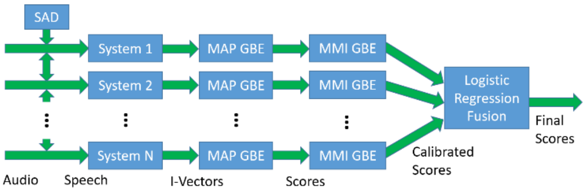

Figure 1: High level system.

3. Data Augmentation

The TRN17 data set was augmented heavily using a recipe inspired by the work in [12] to create a more diverse data set (AUG17) for developing system for the LRE17 fixed condition. Unlike the recipe described in [12], both real isotropic and real point-source noise types were used together. Point source noises were added at a higher rate of 12 occurrences per a minute since many of the audio cuts in TRN17 are short. Lastly, since the VAST data was observed to be quite noisy and with very little reverberation, more aggressive SNR levels of 15, 10, 8, 6, 5, 2 and 0 were used and only the small room real RIRs were used from [12]. Some of the Kaldi [13] systems described later in this document also use speed and volume perturbation for training DNNs.

4. System Architecture

The high-level system architecture for the MITLL/JHU systems is shown in Figure 1. Although the component pipelines are essentially the same for all systems, slightly different algorithms were used for the speech activity detection (SAD), the maximum a posteriori (MAP) adapted linear Gaussian backend (GBE) and the logistic regression score fusion depending on the sub-system and the LRE17 submission type (primary or secondary). These differences will be described in detail in the remaining portions of this document.

5. System Components

5.1. Voice Activity Detection

The MITLL and JHU systems used different voice activity detectors. The MITLL systems used a Gaussian mixture model based SAD trained on a subset of Switchboard. The JHU systems used Kaldi energy VAD, which puts a threshold based on average C0 MFCC of utterance. In all systems described below, the SAD time marks are applied first prior to any further processing.

5.2. I-vector Systems

Several different types of i-vector systems were developed for LRE17. The MITLL SDC (ll_sdc) and JHU SDC (jh_sdc) i-vector systems use 56 conventional SDC features in an i-vector framework, as described in [3] and are similar to the systems submitted in LRE 2011 [9] and 2015 [8]. Both systems use 56 MFCC-SDC features with a 7-1-3-7 configuration followed by short-time cepstral mean subtraction with a 3 seconds sliding window. Other i-vector systems described below use features extracted from DNN bottleneck layers instead of SDCs.

The MITLL, JHU and EPITA systems use slightly different i-vector extraction algorithms. All systems use 2048 component Gaussian mixture universal background model (UBM) for extracting statistics and the corresponding i-vectors all have

600 components. The JHU systems use Kaldi to train full covariance UBMs and extract i-vectors. The MITLL and LSE systems use diagonal UBMs. EPITA i-vectors hyper parameters (T matrices) are trained with a mix of minimum divergence and maximum likelihood training.

5.3. DNN BNF Systems

Several different i-vector systems using DNN BNFs were developed by MITLL, JHU and EPITA. Details for the training data, network type, activation function and architecture for each DNN system developed for LRE17 is given in Table 2. The table also gives the size and location of the bottleneck layer used for extracting features. The sw_bn [14] system is a legacy DNN described used in LRE15 [8]. The remaining DNNs use different Kaldi architectures ("nnet2" versus "nnet3") and training data. Applying the augmentation recipe used to create AUG17 described in Section 3 together with speed perturbation effectively multiplied the amount of training data for the asw_bn DNN by a factor of 6 to 600 hours. Speed perturbation was also applied while training the open condition multi-lingual ba_bn [15] which effectively multiplied the amount of Babel training data by a factor of 3 to 6000 hours.

5.4. X-vector Systems

The xvec x-vector systems use DNNs trained to extract end-to-end embedding [5]. The input features for the x-vector network are BNFs extracted from the fs_bn DNN. The output are 14 languages posteriors, with just one set of posteriors per a recording. The TDNN network architecture was used with exponential linear unit (ELU) non-linearities. The temporal architecture is as follows (see also Table 3):

- Layers 1-4 are time delay layers with different contexts evaluated at the frame level.
- Layer 5 (pooling) computes the mean and standard deviation of layer 4 along the time dimension. These statistics summarize the sequences of frames as unique vectors.
- Layers 6-7 use the sequence summarization vectors from layer 5.
- Layer 8 produces 14 language class output posteriors.

This network is used to compute sequence level embeddings. The embeddings are obtained from layers 6 (xvec4) or from the concatenation of layers 6 and 7 (xvec5) after applying the affine transform but before applying any non-linearities. The network was trained using 5 second sequences extracted from TRN17.

An additional x-vector DNN was trained after the evaluation using the AUG17 data. X-vector embeddings were either extracted from layer 6 (xvec6) or from the concatenation of layers 6 and 7 (xvec7). An analysis of the impact of training the x-vector system with augmented data will follow.

5.5. Sparse Coding System

The MITLL sparse coding system (spar) uses BNFs from the asw_bn DNN as the input. The system trains a sparse coding dictionary using an online learning algorithm [4] based on block-coordinate descent. The dictionary size hyper-parameter K was determined between 160 (2x80) and 640 (8x80). Using the final K = 400, llspark solves for the L1-regularized sparse-coded feature vectors with the sparsity penalty parameter $\lambda = 0.15$. The per-frame sparse feature vectors are then average-pooled to extract an utterance level representation.

6. MAP Adapted Gaussian Scoring

A MAP adapted Gaussian backend (MAP GBE) was used to produce log likelihood scores for each language class using utterance level vector representations from each system. The MAP GBE model uses a single tied full covariance for all language classes and a mean for each language class. The tied covariance is trained with data centered on each language class and equally weighted across all language classes. The JHU systems use relevance MAP with a relevance factor of 64 for the class means and 128 for the tied covariance while the MITLL systems use lambda MAP with a lambda of 0.6 for the class means and the tied covariance. The MITLL systems also used lambda MAP adapted whitening with a lambda factor of 0.9. Different systems used different data sets for training the GBE parameters prior to MAP adaptation (see Table 4).

An alternate form of MAP adaptation was used in JHU systems where different means and a covariance were estimated on the MLS14 and VAST data sources separately. During scoring the data source for an utterance was identified by the data format (FLAC for VAST and Sphere for MLS14). These systems were only used in secondary LRE17 submissions. In Table 4, C1 refers to a single MAP GBE while C2 refers to two source dependent MAP GBEs.

For both the C1 and C2 MAP GBEs, 5-fold cross validation (CV) was used to generate scores on DEV17. These scores were then used to train the Gaussian MMI score calibration and logistic regression fusion parameters and to rank system combinations by performance (see Section 8). The 5 CV folds were picked randomly which turned out to have unfortunate consequences described in .

7. Hyper Parameter Training Data

A summary of the data used to train the utterance level vector extraction hyper parameters (the UBM and T for the i-vector systems and the DNN for the x-vector systems), and the whitening and GBE hyper parameters is summarized in Table 4. All of the systems used TRN17 data to train hyper parameters and language class models. Some systems included additional training data such as 4,000 utterances extracted from Switchboard English (4K-SWB), DEV17 or AUG17.

8. Calibration and Fusion

The scores from the MAP GBE for each sub-system in Table 4 were calibrated using a Gaussian MMI backend [8, 9]. System fusion was accomplished using a logistic regression with a single weight for each system and no offset. This backend was used in primary and secondary submissions. In prior work, this system has been referred to as the "3G" backend.

An alternate C2K score calibration system was used which replaces the single logistic regression described above with two logistic regressions: one trained on the VAST partition of DEV17 and the other trained on the MLS14 partition of DEV17. A channel detector was built using i-vectors from the ll_sdc system. The channel detector uses WCCN scoring and the "3G" backend for score calibration. For each test cut, scores from both logistic regressions are combined using posterior probabilities estimated by the channel detector.

Calibration and fusion performance was estimated on DEV17 using 5-fold cross validation. The final score calibration and fusion backends were estimated on all the DEV17 data and applied to the LRE17 evaluation data. All possible com-

Name	Train Data	Network Type	Act	Layers		Nodes	
				BN	Total	BN	Hidden
sw_bn	SWB 300hr	MLP	Sig	5	7	80	1024
asw_bn	SWB 100hr (aug)	TDNN / NNET3 chain	Relu	9	11	80	625
fs_bn	Fisher 1800hr	TDNN / NNET2	P-norm	6	8	80	3500/350
ba_bn	Babel 2000hr (aug)	TDNN / NNET3 chain	Relu	7	9	80	1024

Table 2: Training data and architecture used for DNN bottleneck feature extraction.

Layer	Context	Size
1 elu	t-4:t+4	256
2 elu	t-4, t, t+4	256
3 elu	t-6, t, t+6	256
4 linear	t	300
5 pooling mean+stddev	sequence	600
6 elu	sequence	256
7 elu	sequence	256
8 softmax	sequence	14

Table 3: TDNN layers for sequence embeddings.

binations of systems up to a maximum of 5 in Table 4 were evaluated on DEV17 for system selection. The combinations were evaluated for the fixed and open conditions systems separately. The primary submission used only the C1 systems while the secondary submission used only C2 systems (both exclude the asw_bn, xvec6 and xvec7). The C2K backend was also applied to the C1 or C2 systems separately and exclude xvec6 and xvec7 systems. System combinations were ranked for the primary (C1) systems, secondary (C2) systems, the C2K C1 and C2K C2 systems separately.

9. Performance Analysis

Performance for the primary, secondary and C2K systems is given in Table 5. The primary systems are the official submission for the evaluation while the others systems were submitted for analysis purposes. The systems combinations were chosen based on the ranking of each system. It's clear that the C2 MAP GBE systems out-perform the C1 systems and the C2K systems out-perform the single backend systems. The C1 C2K system was not submitted but demonstrates the two techniques did not perform well for the original evaluations systems.

After the evaluation several issues were realized. The first is the MLS14 subset of the DEV17 data was highly correlated - the 10 and 3 second duration cuts were sub-segments of each 30 second cut. Fixing this issue in the C2 MAP GBE leads to the results in the "Post Eval" column of Table 5. One can also see that the C2 C2K system, which was not submitted, is now the best performing fixed condition system which uses both a condition dependent MAP GBE and condition dependent score calibration. Initial attempts at decorrelating the VAST data using clustering have not further improved performance for the C2 MAP GBE. The overall gain for using condition dependent systems with fixed CV is about 12% for both the fixed and open conditions.

It is also interesting to look at the oracle results on EVAL17. Evaluating all possible system combinations up to a maximum of 5 systems reveals that nearly the optimal performance (C_{avg} of 0.144) can be obtained with only three systems: fs_bn3c2, xvec7c2 and asw_bn with a C_{avg} of 0.149. This is slightly better than the post eval 5-way fusion result for the fixed condition C2 C2K system in Table 5. Furthermore, the top-ranking single system on the EVAL17 is now the post evaluation x-vector xvec7c2 system with a C_{avg} of 0.195 (the evaluation x-vector system xvec5c2 has a C_{avg} of 0.257). The other BNF systems are only slightly worse than xvec7c2: fs_bn3c2 has a C_{avg} of 0.208 and asw_bn has a C_{avg} of 0.221. The excellent fusion of these three systems may be due to the different training criteria: the fs_bn3c2 DNN is trained with much more English data than the asw_bn DNN, but the asw_bn DNN uses augmentation that the fs_bn3c2 DNN does not. Perhaps a single DNN trained on all of Fisher with augmentation would yield a better performing BNF system (asw_bn and fs_bn3c2 fused perform 17% better than fs_bn3c2 alone). The xvec7c2 system uses a very different sequence level DNN which may be why it fuses well with the two frame level BNF i-vector systems.

The oracle EVAL17 results for the open systems are also interesting. Taking the top performing oracle system combination on the fixed condition and adding a multi-lingual BNF yields nearly the best performance (C_{avg} of 0.130). That is, the fusion of fs_bn3c2, xvec7c2, asw_bn and ba_bn2c2 yields a C_{avg} of 0.132. This is slightly better than the post eval 5-way fusion result for the open condition C2 C2K system in Table 5. The open condition multi-lingual BNF system ba_bn2c2 has a C_{avg} of 0.189 on EVAL17 which is only slightly better than the performance of xvec7c2. While there is a clear relationship between asw_bn and fs_bn3c2 since they are both single language DNNs trained on English telephony data, it is not clear how one could combine the multi-lingual training of ba_bn2c2 with large amount of English data used to train fs_bn3c2. Augmenting the Babel data using the recipe described in Section 3 may help but would increase the amount of training data by another factor of 2 (to 12,000 hours!).

9.1. Conclusions

Post evaluation analysis reveals that the state-of-the-art x-vector system trained on augmented data is clearly a dominant technology. Not only is it the top ranking MITLL/JHU system on the LRE17 evaluation data, but it also fuses very well with other single and multi language DNN BNF systems. Post evaluation analysis revealed that unfortunately the development data was correlated in ways that adversely affected the use of cross validation for the MAP GBE. Addressing this issue and using source dependent fusion yielded a 12% gain over the original submission. The performance can be further improved by fusing the post evaluation x-vector system with two DNN BNF systems for an overall gain of 17% relative to the original submission with only three systems (a similar gain is achieved for the open condition by adding a multi-lingual DNN BNF system). It may be possible to reduce the number of systems by combining the training recipes for the two English trained DNN BNF systems further reducing the complexity of the optimal system combination on the EVAL17 data.

System	MAP Type		UBM/T/x-vector				Whiten		GBE	
	C1	C2	TRN17	4K-SWB	DEV17	AUG17	TRN17	AUG17	TRN17	AUG17
BNF	fs_bn1c1	fs_bn1c2	✓	✓			✓		✓	
	fs_bn2c1	fs_bn2c2	✓	✓	✓		✓		✓	
	fs_bn3c1	fs_bn3c2	✓	✓		✓	✓	✓	✓	✓
	sw_bn		✓				✓		✓	
	asw_bn		✓				✓		✓	
X-vector	xvec4c1	xvec4c2	✓				✓		✓	
	xvec5c1	xvec5c2	✓				✓		✓	
	xvec6c1	xvec6c2	✓				✓		✓	
	xvec7c1	xvec7c2	✓				✓		✓	
ML BNF	ba_bn1c1	ba_bn1c2	✓	✓			✓		✓	
	ba_bn2c1	ba_bn2c2	✓	✓			✓		✓	
SDC	jh_sdc1c1	jh_sdc1c2	✓	✓			✓		✓	
	jh_sdc2c1	jh_sdc2c2	✓	✓	✓		✓		✓	
	ll_sdc		✓				✓		✓	
Sparse	spar		✓				✓		✓	

Table 4: List of systems and data used to train them. The C2 and asw_bn systems were only used in secondary system submissions while xvec6 and xvec7 were developed after the evaluation.

Condition	Submission	Systems	Dev C_{avg}	Eval C_{avg}	Post Eval C_{avg}
Fixed	Primary	fs_bn2c1fs_bn3c1 sw_bn ll_sdc	0.1376	0.1807	0.1728
	Secondary	fs_bn2c2 fs_bn3c2 xvec5c2 sw_bn spar	0.1320	0.1728	0.1669
		fs_bn2c1	0.1814	0.2308	0.2157
	C1 C2K	fs_bn2c1 fs_bn3c1 jh_sdc2c1 asw_bn	0.1137	0.1737	0.1636
		fs_bn2c1 fs_bn3c1 xvec4c1 asw_bn ll_sdc	0.1110	0.1686	0.1630
	C2 C2K*	fs_bn2c2 xvec4c2 xvec5c2sw_bn asw_bn	0.1133	0.1718	0.1586
Open	Primary	fs_bn2c1 fs_bn3c1 ba_bn1c1 ba_bn2c1	0.1211	0.1625	0.1507
	Secondary	fs_bn2c2 xvec5c2 ba_bn2c2 sw_bn	0.1219	0.1588	0.1503
	C1 C2K	fs_bn2c1 fs_bn3c1 xvec4c1 asw_bn ll_sdc	0.1101	0.1579	0.1630
	C2 C2K	fs_bn2c2 fs_bn3c2 ba_bn2c2 jh_sdc1c2 asw_bn	0.1036	0.1551	0.1439

Table 5: Combination of sub-systems for each submission and performance on DEV17 (C2 C2K was not submitted).

Language	Babel Language Pack
Cantonese	IARPA-babel101b-v0.4c
Assamese	IARPA-babel102b-v0.5a
Bengali	IARPA-babel103b-v0.4b
Pashto	IARPA-babel104b-v0.bY
Turkish	IARPA-babel105b-v0.4
Tagalog	IARPA-babel106b-v0.2g
Vietnamese	IARPA-babel107b-v0.7
Haitian	IARPA-babel201b-v0.2b
Swahili	IARPA-babel202b-v1.0d
Lao	IARPA-babel203b-v3.1a
Tamil	IARPA-babel204b-v1.1b
Kurmanji	IARPA-babel205b-v1.0a
Zulu	IARPA-babel206b-v0.1e
Tok Pisin	IARPA-babel207b-v1.0e
Cebuano	IARPA-babel301b-v2.0b
Kazakh	IARPA-babel302b-v1.0a
Telugu	IARPA-babel303b-v1.0a
Guarani	IARPA-babel305b-v1.0c
Igbo	IARPA-babel306b-v2.0c
Amharic	IARPA-babel307b-v1.0b
Mongolian	IARPA-babel401b-v2.0b
Javanese	IARPA-babel402b-v1.0b
Dholuo	IARPA-babel403b-v1.0b
Georgian	IARPA-babel404b-v1.0a

Table 6: 24 Babel languages used for training multi-lingual DNNs.

10. References

[1] "NIST 2017 language recognition evaluation plan," 2017.

[2] Niko Brummer, "Focal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scores," https://sites.google.com/site/nikobrummer/, 2007.

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

[4] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, 2010.

[5] D. Garcia-Romero, D. Snyder, G. Sell, D. .Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Proc. of ICASSP*, 2017.

[6] P. Torres-Carrasquillo, E. Singer, M. Kohler, R. Greene, D. Reynolds, and J. Deller, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *International Conference on Spoken Language Processing*, 2002.

[7] Y. Song, B. Jiang, Y. Bao, S. Wei, and L.-R. Dai, "I-vector representation based on bottleneck features for language identification," *IEEE Electronics Letters*, pp. 1569–1580, 2013.

[8] P. Torres-Carrasquillo, N. Dehak, E. Godoy, D. Reynolds, F. Richardson, S. Shum, E. Singer, and D. Sturim, "The MITLL NIST LRE 2015 language recognition system," in *Proc. of IEEE Odyssey*, 2016.

[9] E. Singer, P. Torres-Carrasquillo, D. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. Sturim, "The MITLL NIST LRE 2011 language recognition system," in *Proc. of IEEE Odyssey*, 2011, pp. 209–215.

[10] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. of ICASSP*, 1992, pp. 517–520.

[11] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: a resource for the next generations of speech-to-text," in *Proc. of LREC*, 2004.

[12] T. Ko, V. Peddinti, D. Povey, M. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. of ICASSP*, 2017.

[13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J Silovsky, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," in *Proc. of IEEE ASRU*, 2011.

[14] F. Richardson, Douglas Reynolds, and Najim Dehak, "Deep neural network approaches to speaker and language recognition," in *IEEE Signal Processing Letters*, 2015.

[15] J. Trmal, M. Wiesner, V. Peddinti, X. Zhang, P. Ghahremani, Y. Wang, V. Manohar, H. Xu, D. Povey, and S. Khudanpur, "The kaldi openkws system: Improving low resource keyword search," in *Proc. of Interspeech*, 2017.