INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

THU-EE system description for NIST LRE 2015

Liang He, Yao Tian, Yi Liu, Jiaming Xu, Weiwei Liu, Cai Meng, Jia Liu

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
heliang@mail.tsinghua.edu.cn

Abstract

This paper describes the systems developed by the Department of Electronic Engineering of Tsinghua University for the NIST Language Recognition Evaluation 2015. We submitted one primary and three alternative systems for the fixed training data evaluation and didn't take part in the open training data evaluation for our limited data resources and computation capability. Both the primary system and three alternative systems are fusions of multiple subsystems. The primary system and alternative systems are identical except for the training, development and fusion data. The subsystems are different in feature, statistical modeling or backend approach. The features of our subsystems include MFCC, PLP, TFC, PNCC and Fbank. The statistical modeling of our subsystems can be roughly categorized into four types: i-vector, deep neural network, multiple coordinate sequence kernel (MCSK) and phoneme recognizer followed by vector space models (PR-VSM). The backend approach includes LDA-Gaussian, SVM and extreme learning machine (ELM). Finally, these subsystems are fused by the FoCal toolkit. Our primary system is presented and briefly discussed. Post-key analyses are also addressed, including comparison of different features, modeling backend approaches and a study of their contribution to the whole performance. The processing speed for each subsystem is also given in the paper.

Index Terms: NIST LRE 2015, spoken language recognition, deep neural network, bottleneck, i-vector

1. Introduction

This paper describes the systems developed by the Department of Electronic Engineering of Tsinghua University (THU-EE) for the NIST Language Recognition Evaluation (LRE) 2015 [1].

The NIST LRE 2015 included 20 languages and featured 6 language clusters: Arabic (Egyptian, Iraqi, Levantine, Maghrebi, Modern Standard), Chinese (Cantonese, Mandarin, Min, Wu), English (British, General American, Indian), French (West African, Haitian Creole), Slavic (Polish, Russian) and Iberian (Caribbean Spanish, European Spanish, Latin American Spanish, Brazilian Portuguese). Different from the past NIST LREs [2,3], the NIST LRE 2015 tried to make progress in the context of languages that are similar to each other and frequently mutually intelligible. It was emphasized by defining a new performance metric which only considered distinguishing languages within each cluster. From the view of linguistics and historical NIST LRE experiences, the language belongs to the same cluster are apt to be confusable languages. Thus, one feature of the NIST LRE 2015 was to distinguish confusable languages. Another feature was the segment duration. Segments were selected to cover a broad range of speech durations, not limited to approximately 3 seconds, 10 seconds, or 30 seconds.

Currently, there are two dominant approaches to spoken language recognition: acoustic and phonotactic. The acoustic systems are based on short time spectral features. Our acoustic subsystems not only include some of cutting edge approaches, such as deep neural network (DNN), convolution neural network (CNN), long short term memory (LSTM) and i-vector [4], but also are designed to explore noise robust features, high order statistics and more effective backend approaches. The phonotactic systems are based on lattices of tokens extracted by phone recognizers [11,12]. Our phonotactic subsystem is a triphone-VSM-SVM subsystem based on the phone decoder developed by our lab.

The rest of the paper is organized as follows. Section 2 illustrates data used for the NIST LRE 2015. Section 3 describes our submitted systems. We put more emphasize on the cutting edge methods and our novel parts. Section 4 details system configurations. Post-key experiments are conducted to examine our primary, alternative and subsystem performance. A brief analysis is also discussed based on the experimental results.

2. Training and development data

NIST provided a training and development dataset specifically collected for the fixed training data condition [1]. This dataset contains two parts. The first part contains segments with word alignment from Switchboard-1 for training NN-related models. The second part includes 3511 full 2-channel telephone calls (CTS) and segments extracted from broadcast recordings containing narrow-band speech (BNBS) for 20 target languages. The hours of training speech for each target language are not balanced. For example, the Arabic-Modern Standard language has 0.5 hours while the Arabic-Levantine language has 41.1 hours.

We randomly split the second part into two sets: *lre15-train* (5262 utterances) was used to train language models and *lre15-dev* (1760 utterances) was used to do self-evaluation and estimate backend and fusion parameters. To mitigate the degradation caused by the length variation, we cut each segment into 3-60 seconds.

Our primary system used both the *lre15-train* and *lre15-dev* for training and used the *lre15-dev* for self-evaluation, fusion and calibration. The alternative-1 and alternative-2 systems were the same to the primary system except for the usage of data. The alternative 2 system used the *lre15-train* to train system and used the *lre15-dev* to perform self-evaluation, fusion and calibration. The alternative-1 system used the *lre15-train* utterances to train system but the fusion parameters were the same to the alternative-2 system. The alternative-3 system was the same to the alternative-2 system but without PRVSM-SVM subsystem.

http://dx.doi.org/10.21437/Interspeech.2016-791

2.1. Data re-usage

As we stated, our primary system used both the *lre15-train* and *lre15-dev* for training and did self-evaluation on the same *lre15-dev.* If the training and self-evaluation used the same segments, the self-evaluation results were likely to be over optimistic, especially for subsystems using SVM as classifiers. We took random segmentation strategy to produce two datasets on the same *lre15-dev.* One was for the training and the other was for the self-evaluation. Experimental results had shown that this simple data re-usage strategy could avoid over optimistic results, thus provided relatively reliable parameters for subsystem fusion.

3. System

Our primary system is a linear fusion of 19 subsystems, see Figure 1, Table 1. We take Fbank, MFCC, RASTA, SDC, GMM, i-vector, SVM, LDA and Gaussian as common configurations [13,14]. Details about them are presented in the next section. Here, we put more emphasize on the cutting edge methods and our novel parts.

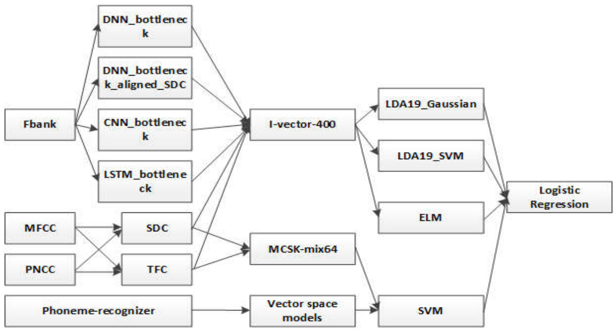


Figure 1. Primary system framework for NIST LRE 2015

Table 1. Subsystems for NIST LRE 2015

tag	brief description
1	fbank_dnn_bn_mix2048_ivec400_lda19_gaussian
2	fbank_dnn_bn_mix2048_ivec400_lda19_svm
3	fbank_dnn_bn_mix2048_ivec400_ELM
4	fbank_cnn_bn_mix2048_ivec400_lda19_gaussian
5	fbank_cnn_bn_mix2048_ivec400_lda19_svm
6	fbank_cnn_bn_mix2048_ivec400_ELM
7	fbank_lstm_bn_mix2048_ivec400_lda19_gaussian
8	fbank_lstm_bn_mix2048_ivec400_lda19_svm
9	fbank_dnn_bn_align_sdc_mix2048_ivec400_lda19_gaussian
10	mfcc_sdc_mcsk_mix64_svm
11	mfcc_sdc_mix1024_ivec400_lda19_gaussian
12	mfcc_tfc_mcsk_mix64_svm
13	mfcc_tfc_mix1024_ivec400_lda19_gaussian
14	pncc_sdc_mcsk_mix64_svm
15	pncc_sdc_mix1024_ivec400_lda19_gaussian
16	pncc_tfc_mcsk_mix64_svm
17	pncc_tfc_mix1024_ivec400_lda19_gaussian
18	sdc_mix2048_ivec400_lda19_gaussian
19	pr_vsm_svm

3.1. imPNCC

Improved Multitaper Power Normalized Cepstral Coefficients (imPNCC) are extracted followed the recipes described in the [15]. The PNCC is enhanced with the gamma-chirp filterbank and uses a multitaper named *multi-peak* to improve the performance under noise and clean condition. The imPNCC uses a frequency domain 40-channel gamma chirp filter banks to analyze the segments with 10ms frame shift and 25ms frame length. Instead of conventional Hamming window, imPNCC pre-processes the frame with a multi-peak multitaper before short time Fourier transform. The cutoff frequencies of the filter-bank are at 0Hz and 4000Hz, respectively. The remaining procedure and parameters keep the same to the [15]. The extraction flow chart is shown as Figure 2 (left).

3.2. TFC

Time frequency cepstral (TFC) feature extraction is performed as follows, see Figure 2 (right): 9 successive frames of basic features are extracted first to form a cepstral matrix. Then a DCT is implemented on the cepstral matrix in the temporal direction to remove correlation. Finally, the elements (39 dim) in the upper-left triangular area are selected by scanning in a zigzag order [17].

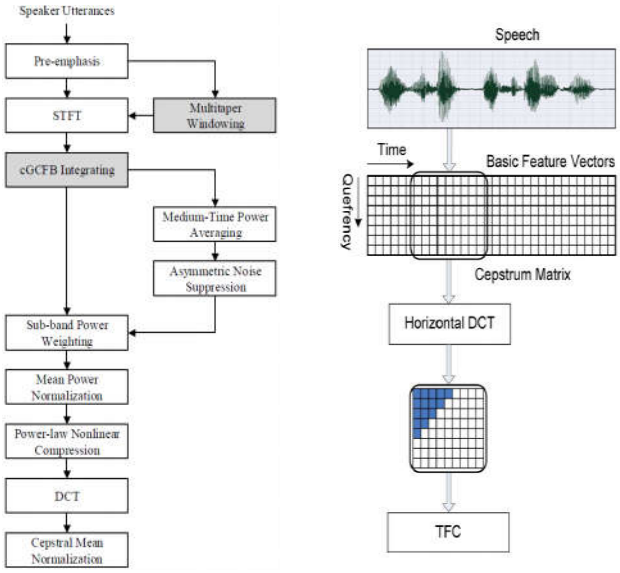


Figure 2. Flow charts of imPNCC (left) and TFC (right)

3.3. DNN-bottleneck

The deep neural network (DNN) is a feed-forward artificial neural network with multi-hidden layers [6,7]. Bottleneck here refers to a hidden layer placed in the middle of the neural network which has fewer nodes than other layers. The activation of this layer is regarded as the bottleneck feature. The DNN is trained to discriminate senones (tied triphone states). The input layer of the DNN has 1320 nodes composed of 11 frames (5 frames on each side of the frame) where each frame consists of 120 Fbank features. The DNN has five hidden layers and each hidden layer has 1200 nodes except that the fifth hidden layer which is the bottleneck layer has 39 nodes. The output of the DNN with respect to senones has 2227 nodes. After the extraction of the bottleneck features, they are fed into the traditional total variability matrix to extract 400 dim i-vectors, see Figure 3 (left).

3.4. DNN-bottleneck-align

The DNN-bottleneck-align model is the same as [18] proposed in speaker verification. During i-vector modeling, a GMM is trained in the traditional unsupervised way using bottleneck features to calculate frame posterior probabilities and MFCC-SDC features are combined with these posterior probabilities to calculate sufficient statistics, see Figure 3 (right).

3.5. CNN-bottleneck

The Convolutional Neural Network (CNN) contains one convolutional layer and four fully-connected layers. For the convolutional layer, the filter size is 8 and the number of filters for each receptive field is 100. The pooling size is 3. For the fully-connected layer, each layer has 1200 nodes and the last hidden layer which is the bottleneck layer has 39 nodes. The rest CNN-bottleneck model configurations are the same as the DNN-bottleneck model.

3.6. LSTM-bottleneck

The architecture of the Long Short Term Memory (LSTM) model is similar to [19]. It has two LSTM layers. The first LSTM layer has 800 cells and 512 recurrent projection units. The second LSTM layer has 800 cells and 39 recurrent projection units. The activation of the recurrent projection units in the second LSTM layer is regarded as the bottleneck feature. The rest LSTM-bottleneck model configurations are the same as the DNN-bottleneck model.

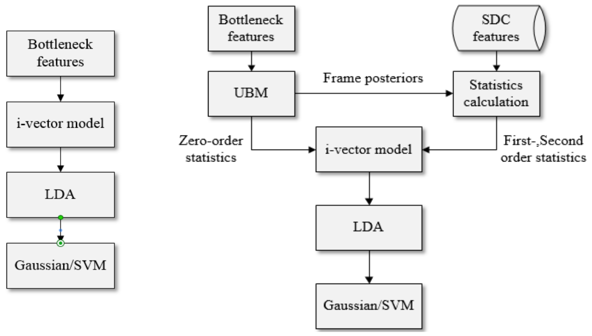


Figure 3. Bottleneck + i-vector (left) and bottleneck alignment + SDC + i-vector (right)

3.7. MCSK

Multiple coordinate sequence kernel (MCSK) can be regard as a mixed method of generalized linear discriminant sequence (GLDS) kernel and kullback-leibler (KL) kernel [20,21]. The GLDS kernel benefits from the high order statistic information from spectral feature. On the contrary, the KL kernel benefits from the occupation information. The MCSK combines them together [22]. For a given spectral feature, the discrimination information originates from two sources: the selected coordination (mixture component) and its representation (we use 2-order polynomials here). The MCSK take advantages both of them and demonstrates good performance in low computation resources, see Figure 4.

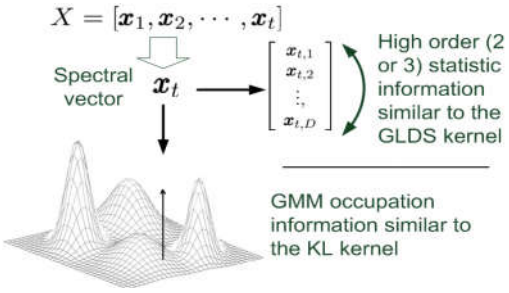

$X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_t]$
Spectral vector
$\boldsymbol{x}_t$
$\boldsymbol{x}_{t,1}$
$\boldsymbol{x}_{t,2}$
$\boldsymbol{x}_{t,D}$
High order (2 or 3) statistic information similar to the GLDS kernel
GMM occupation information similar to the KL kernel

Figure 4. Motivation of MCSK

3.8. ELM

Extreme learning machine (ELM) is a single-hidden layer feed forward network which randomly selects input weights and hidden neuron biases without training [23]. The output weights are analytically determined by the Moore-Penrose generalized inverse. Here, the ELM is used as a backend classifier for the extracted i-vectors, similar to LDA-Gaussian or SVM backend.

4. Experiments

4.1. Configuration

MFCC features were computed with 25ms frame length and 10ms frame shift. RASTA and cepstral mean and variance normalization (CMVN) were applied in basic MFCC computation. Shifted-delta-cepstral (SDC) coefficients consist of 7 static MFCC and 49 shifted delta cepstral coefficients, under a 7-2-3-7 configuration [13,14]. We randomly selected 10 hours from *lre15-train* to train three gender independent universal background models (UBM) with 64, 1024 and 2048 mixture components respectively. The UBM with 64 mixture components was used for *tag10* subsystem[1], the UBM with 1024 mixture components was used for *tag11, tag13, tag15, tag17* subsystems, and the UBM with 2048 mixture components was used for *tag1 to tag9* and *tag18* subsystems. Each language had at least 0.3 hours. Both the total variability matrix and linear discriminant analysis (LDA) matrix were trained on the whole *lre15-train.* The dimensions of i-vector and LDA were 400 and 19 respectively. The zero-order and centered first-order Baum-Welch statistics were extracted by using the UBM. The backend approaches for i-vectors were LDA-Gaussian, LDA-support vector machine (SVM) and extreme learning machine (ELM). In the LDA-Gaussian setting, each language was assumed to be a Gaussian distribution with a full covariance matrix shared by all the languages. In the LDA-SVM setting, the kernel of SVM was linear and the training of target language adopted the *one-vs-the-rest* strategy. The ELM setting was similar to the LDA-SVM setting. The *tag19* subsystem was developed using an English phone recognizer (GMM-HMM) developed by our lab and trained on given Switchboard database. A high-dimensional phonotactic feature vector with the phone 3-gram statistics was obtained by the *lattice-tool of SRILM* [24]. Our 19 subsystems were fused using multiclass logistic regression by the *FoCal* toolkit [25]. For each system, the performance is evaluated using the metric ($\min C_{\text{avg}}$) defined by the NIST LRE 2015 [1].

[1] See Table 1

4.2. Result

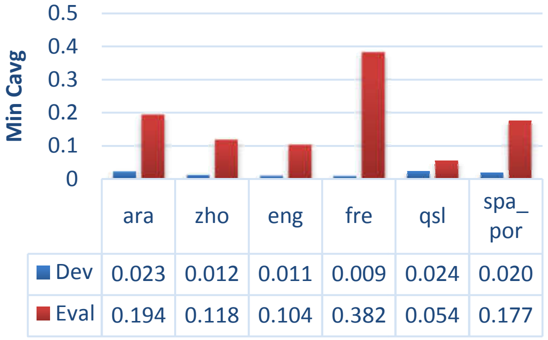
The minimum average costs of the THU-EE primary and alternative systems are shown in Table 2. To analyze the contribution of each method, Table 3 gives each subsystem's performance. Figure 5 also presents our primary system's performance on each language cluster.

Table 2. Experimental results on the self-evaluation and evaluation data

min Cavg	Self-eval	Eval
Primary	0.01348	0.2093
Alternative1	0.01542	0.2142
Alternative 2	0.01733	0.2048
Alternative 3	0.01985	0.2076

Table 3. Experimental results of THU-EE subsystems on the self-evaluation and evaluation data

Tag	Self-eval	Eval	Tag	Self-eval	Eval
1	0.06544	0.2225	11	0.09479	0.3087
2	0.09164	0.2154	12	0.06920	0.2848
3	0.07405	0.2340	13	0.08466	0.3027
4	0.06592	0.2211	14	0.08719	0.3009
5	0.09548	0.2182	15	0.09956	0.3210
6	0.07258	0.2353	16	0.08006	0.2828
7	0.08552	0.2271	17	0.08098	0.3090
8	0.11561	0.2353	18	0.09736	0.2860
9	0.10217	0.2675	19	0.1205	0.3245
10	0.08280	0.2931			



	ara	zho	eng	fre	qsl	spa_ por
🟦 Dev	0.023	0.012	0.011	0.009	0.024	0.020
🟥 Eval	0.194	0.118	0.104	0.382	0.054	0.177

Figure 5. Performance comparison of THU-EE primary system on different language clusters

4.3. Speed

The speed test was performed on one core of Intel Xeon E5-2640. Our primary system was about 1.5 real-time (RT). The *NN-related* subsystem was about 0.1 RT. The traditional *SDC + i-vector* subsystem was about 0.03 RT. The *MCSK* subsystem was about 0.005 RT. And the *PRVSM-SVM* subsystem was about 0.3 RT.

4.4. Analysis

No significant differences in performance can be found among the four developed systems from the Table 2. Although we adopt data re-usage strategy, the primary system is still a little more optimistic on the self-evaluation dataset. Perhaps, if we reserve more data for *lre15-dev,* the advantages of data re-usage would become more obvious.

From the Table 3, we find that:

(1) ImPNCC is worse than MFCC by comparing *tag10-13* with *tag14-17* subsystems. Perhaps, the speech quality of this evaluation is relatively good while imPNCC is more suitable for noisy speech.

(2) TFC is better than SDC by comparing *tag12,13,16,17* with *tag10,11,14,15* subsystems. TFC can benefits from joint time-frequency information. Besides, TFC is also a kind of long-term feature.

(3) By comparing *tag1,4,7*, *tag2,5,8* and *tag3,6* subsystems, the SVM backend is better than the Gaussian backend. And the Gaussian backend is better than the ELM. We notice that on the self-evaluation dataset, the $\min C_{\text{avg}}$ with either Gaussian or ELM backend is better than the $\min C_{\text{avg}}$ with SVM backend. On the evaluation dataset, the SVM performs better. We provide two explanations. First, the SVM has better generalization ability than the Gaussian and ELM. Second, the statistics of self-evaluation and evaluation data are different, which is further proved by our results on language clusters.

(4) NN-related subsystems are the most effective methods by comparing *tag1-9* with *tag10-18* subsystems. Deep neural network is good at extracting complicated and intrinsic structure from speech. It provides a more effective description than SDC for language recognition. That's the reason why NN-related subsystems outperform other subsystems with a reasonable improvement. To our surprise, the DNN, CNN and LSTM have similar performances. They have different unit structures and connection networks. The CNN is more suitable for spatial signal and the LSTM is more suitable for temporal signal. However, the $\min C_{\text{avg}}$ of them are very similar in the NIST LRE15 evaluation data. Perhaps, the language information can be well captured by either of them.

(5) By comparing *tag12* subsystem and *tag18* subsystem, the MCSK has a similar performance with the i-vector at a lower computation cost. The success of MCSK means that language information may be further investigated from the consideration of higher-order statistics.

Figure 5 gives performance comparison of our primary system on different language clusters. On the self-evaluation dataset, the primary system works best on the *fre* language cluster and worst on the *qsl* language cluster. On the evaluation dataset, the same system works best on the *qsl* language cluster and worst on the *fre* language cluster. The mismatch of self-evaluation and evaluation dataset is still a big problem for system calibration or fusion. We should do more research to find and measure the mismatch. Besides, we should put more emphasize on the generalization ability of statistical modeling.

5. Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 61403224, 61370034 and 61273268.

6. References

[1] NIST, "The 2015 NIST Language Recognition Evaluation Plan (LRE15)," http://www.nist.gov/speech/tests/lre/2015/LRE15_EvalPlan_v22.pdf, 2015.

[2] NIST, "The 2009 NIST Language Recognition Evaluation Plan (LRE09)," http://www.nist.gov/speech/tests/lre/2009/LRE05EvalPlan-v5-2.pdf, 2015.

[3] NIST, "The 2011 NIST Language Recognition Evaluation Plan (LRE11)," http://www.nist.gov/speech/tests/lre/2011/LRE07EvalPlan-v8b.pdf, 2015.

[4] Najim Dehak, Patrick Kenny, Reda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, no. 4, pp. 788–798, 2011.

[5] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 15, no.4, pp. 1435–1447, 2007.

[6] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdelrahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE,* vol. 29, no. 6, pp. 82–97, 2012.

[7] George E Dahl, Dong Yu, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, "Imagenet classification with deep convolutional neural networks," *in Advances in neural information processing systems,* 2012, pp. 1097–1105.

[9] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," *in Proceedings of the 25th international conference on Machine learning.ACM,* 2008, pp. 160–167.

[10] Tianfan Fu, Yanmin Qian, Yuan Liu, and Kai Yu, "Tandem deep features for text-dependent speaker verification," *in the proceedings of Fifteenth Annual Conference of the International Speech Communication Association,*2014.

[11] H. Li, B. Ma, and C. H. Lee, ''A vectorspace modeling approach to spoken language identification,'' *IEEE Transactions on Speech and Audio Processing,* vol. 15, no. 1,pp. 271–284, Jan. 2007.

[12] H. Li, B. Ma, and C.-H. Lee, ''Vector-based spoken language classification,'' in Springer Handbook of Speech Processing and Speech Communication, J. Benesty, M. M. Sondhi, and A. Huang, Eds. New York, NY, USA:Springer-Verlag, 2008.

[13] H. Hermansky and N. Morgan, ''RASTA processing of speech,'' *IEEE Transactions on Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, 1994.

[14] P. Torres-Carrasquillo, E. Singer, M. Kohler, R. Greene, D. Reynolds, and J. Deller, Jr., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features,'' *in Proc. Int. Conf. Spoken Lang. Process., Denver, CO, USA,* 2002,pp. 89–92.

[15] K. Chanwoo, R.M. Stern, "Power-Normalized Cepstral Coefficients for robust speech recognition," *in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE,* pp. 4101 - 4104, 2012.

[16] Y. Liu, L. He and J, Liu, "Improved multitaper PNCC feature for robust speaker verification," *ISCSLP, pp. 168 - 172, 2014.*

[17] W.Q. Zhang, L. He, Y. Deng, J. Liu, and M. T. Johnson, "Time-frequency cepstral features and heteroscedastic linear discriminant analysis for language recognition," *IEEE Transactions on Audio, Speech and Language Processing,* vol. 19, no. 2, pp. 266 - 276, 2011.

[18] Y. Tian, M.Cai, L. He and J. Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification," *in the proceedings of Sixteenth Annual Conference of the International Speech Communication Association,* 2015.

[19] Sak, Hasim and Senior, Andrew and Beaufays, Francoise, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling, *in the proceedings of Fifteenth Annual Conference of the International Speech Communication Association,* 2014.

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

[21] W.M. Campbell W, K. Assaleh, C. Broun, "Speaker recognition with polynomial classifiers," *IEEE Transactions on Speech and Audio Processing,* vol. 10, pp. 205 - 212, 2002.

[22] L. He, Y. Deng, and J. Liu, "Method and system for speaker recognition based on multiple coordinate sequence kernel," China Patent, Sept. 1, 2009, CN101640043.

[23] G.-B. Huang, H.-M. Zhou, X.-J. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 2, pp. 513 - 529, 2012.

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

[25] N. Brummer, "Focal Multi-Class Tools for Evaluation, Calibration and Fusion of, and Decision-Making with, Multi-Class Statistical Pattern Recognition Scores", http://sites.google.com/site/nikobrummer/. 2015