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

The 2015 NIST Language Recognition Evaluation: the Shared View of I2R, Fantastic4 and SingaMS

Kong Aik Lee[1], *Haizhou Li*[1], *Li Deng*[2], *Ville Hautamäki*[3], *Wei Rao*[4], *Xiong Xiao*[4], *Anthony Larcher*[5], *Hanwu Sun*[1], *Trung Hieu Nguyen*[1], *Guangsen Wang*[1], *Aleksandr Sizov*[1,3], *Jianshu Chen*[2], *Ivan Kukanov*[3], *Amir Hossein Poorjam*[3], *Trung Ngo Trong*[3], *Cheng-Lin Xu*[4], *Hai-Hua Xu*[4], *Bin Ma*[1], *Eng-Siong Chng*[4], *Sylvain Meignier*[5]

[1]Institute for Infocomm Research, A*STAR, Singapore
[2] Microsoft Research, Redmond, WA 98052, USA
[3] University of Eastern Finland (UEF), Finland
[4] Nanyang Technological University, Singapore
[5] Universite du Maine - LIUM, France

Abstract

The series of *language recognition evaluations* (LRE's) conducted by the National Institute of Standards and Technology (NIST) have been one of the driving forces in advancing spoken language recognition technology. This paper presents a shared view of five institutions resulting from our collaboration toward LRE 2015 submissions under the names of I2R, Fantastic4, and SingaMS. Among others, LRE'15 emphasizes on language detection in the context of closely related languages, which is different from previous LRE's. From the perspective of language recognition system design, we have witnessed a major paradigm shift in adopting deep neural network (DNN) for both feature extraction and classifier. In particular, deep bottleneck features (DBF) have a significant advantage in replacing the shifted-delta-cepstral (SDC) which has been the only option in the past. We foresee deep learning is going to serve as a major driving force in advancing spoken language recognition system in the coming years.

Index Terms: spoken language recognition, evaluation

1. Introduction

Spoken language recognition is the task to determine the identity of the language spoken in a given speech utterance. It serves to aid general-purpose multilingual speech-based applications, such as spoken language translation and multilingual speech recognition. Spoken *language recognition evaluation* (LRE) campaigns, regularly conducted by the National Institute of Standards and Technology (NIST), is one of the main driving forces advancing language recognition technology. Due to the relatively high recognition accuracy obtained in the basic language detection task [1], the focus has shifted, first, to closely related language pairs in LRE'11 [2] and later to whole clusters of closely related languages and dialects in LRE'15 [3].

The NIST LRE's have been focusing on language detection: given a segment of speech and a language hypothesis, the task is to decide whether a target language was spoken in the given segment. LRE'15 is different from the prior LRE's in certain key aspects. The current LRE'15 emphasizes on making decisions in the context of languages that are closely related and frequently mutually intelligible [3]. Twenty such languages, grouped into six language clusters, are listed in Table 1. Also, LRE'15 dictates a closed-set scenario, where the set of non-target languages are limited to other languages in the same cluster. One more difference from previous editions is that the training data used to develop the models was limited to the set as listed in Table 2. The intention was to allow participants to focus on algorithm development instead of making the evaluation as merely a data selection exercise.

Table 1: Language clusters and target languages for NIST LRE'15.

Cluster	Target Languages
Arabic	Egyptian, Iraqi, Levantine, Maghrebi, Modern Standard
Chinese	Cantonese, Mandarin, Min, Wu
English	British, General American, Indian
French	West African, Haitian Creole
Slavic	Russian, Polish
Iberian	Caribbean Spanish, European Spanish, Latin American Spanish, Brazilian Portuguese

Our motivation in writing the current paper is twofold. Firstly, to report recent advances and the major paradigm shift that we have witnessed in spoken language recognition based on the joint efforts of five research groups on the recent NIST LRE'15. In particular, the progress obtained by the use of deep learning paradigm [4, 5, 6] was an impressive one. Secondly, to discuss and share our views on the potential future directions. Selected experimental results, focusing on feature and discriminative training, are presented to demonstrate the strengths and weaknesses of the techniques discussed.

2. Baseline, data and performance metric

2.1. Baseline

We started off with the baseline system as shown in Fig. 1. Speech utterances are first parameterized as sequences of shifted delta cepstral (SDC) feature vectors [7, 8]. They are then represented as i-vectors which compress the variable-length sequences along the time axis and project the resulting vectors to the low-dimensional total variability space. Having its root in

http://dx.doi.org/10.21437/Interspeech.2016-624

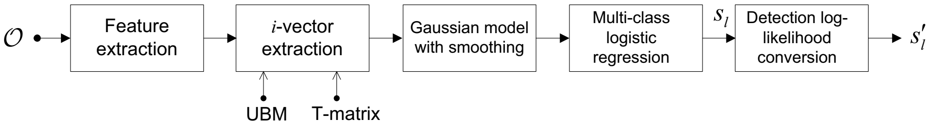
$\mathcal{O}$
Feature extraction
i-vector extraction
Gaussian model with smoothing
Multi-class logistic regression
s_l
Detection log-likelihood conversion
s'_l
UBM
T-matrix

Figure 1: Baseline i-vector system consisting of Gaussian modeling followed by multi-class logistic regresion.

Table 2: Training data for fixed training data condition.

Dataset	Usage
LRE'15 Training Data Part 1 *LDC2015E87*	Dev set
LRE'15 Training Data Part 2 *LDC2015E88*	Dev set
Switchboard Cellular Part 2	No transcription
Switchboard-1 Release 2	Bottleneck DNN

speaker recognition [9], i-vector was shown to be useful as well for language recognition [10, 11, 12, 13, 14]. One precaution to be taken into account is the parameters of the i-vector extractor, including the universal background model (UBM) and the total variability matrix, should be trained from speech utterances covering multiple languages.

As shown in Fig. 1, each language is modeled as a Gaussian distribution in the i-vector space. The next component is the multi-class logistic regression (MCLR) module. The idea is to do a global scaling followed by a language-dependent shifting on the log-likelihood scores from the Gaussian classifiers such that they are better calibrated to a single decision threshold for all languages [15, 16]. For the language detection task as defined for LRE'15, the calibrated scores s_l are converted to detection log-likelihood ratio

$$s'_l = \log \frac{p(\mathcal{O}|L_l)}{p(\mathcal{O}|\neg L_l)} = s_l - \log\left(\frac{1}{N-1}\sum_{m\neq l}\exp(s_m)\right) \tag{1}$$

The conversion is done per cluster since the performance cost is evaluated in this manner (e.g., N is 5 for the Arabic cluster).

2.2. Data

Table 2 shows four LDC datasets made available by NIST for LRE'15 participants. Among the four datasets, *Training Data Part 1* consists of 267 hours of telephone recordings in 3 languages including Egyptian Arabic, Mandarin Chinese and US English. This dataset was compiled from previously collected CallHome and CallFriend corpora. The *Training Data Part 2* is new. It consists of 249 hours of speech recordings in 17 other languages listed in Table 1. Clearly, the training data is imbalanced among the target languages, with some languages are limited to less than an hour of speech (e.g., Brazilian Portuguese). Such constraint was taken into account in our baseline system of Fig. 1. The language-specific covariance matrices are derived by smoothing between the sample covariance matrix estimated for each language and a global covariance matrix. The latter was trained by pooling training data from all languages.

Also shown in Fig. 1 is the UBM and T matrix. These components and the Gaussian classifiers were trained using $2/3$ of available training data drawn from the Dev set. The remaining $1/3$ of the data was used to train the MCLR module. All i-vectors were whitened and projected to the unit sphere [17].

2.3. Performance metric

The official performance metric defined for LRE'15 is the average cost computed across all languages in the same cluster:

$$\begin{aligned} C_{\text{avg}} &= C_{\text{miss}} P_{\text{tar}} \underbrace{\frac{1}{N}\sum_{l=1}^{N} P_{\text{miss}}(L_l)}_{P_{\text{miss}}(\theta_{\text{DET}})} \\ &+ C_{\text{fa}}(1-P_{\text{tar}}) \underbrace{\frac{1}{N}\sum_{l=1}^{N}\left[\frac{1}{N-1}\sum_{m\neq l} P_{\text{fa}}(L_l, L_m)\right]}_{P_{\text{fa}}(\theta_{\text{DET}})} \end{aligned} \tag{2}$$

where the application parameters C_{miss}, C_{fa}, and P_{tar} are set to 1, 1, and 0.5, respectively. The probabilities of miss $P_{\text{miss}}(L_l)$ and false alarm $P_{\text{fa}}(L_l, L_m)$ are computed for each language L_l against other languages L_m in the same cluster.

The performance of the baseline system using different input features is shown Table 3. In addition to the actual cost, we also show the minimum C_{avg} and the equal-error-rate (EER). For the actual cost, the decision threshold was set to 0, whereas the minimum cost is obtained by varying the threshold θ_{DET} in (2) to give the minimum value. The EER corresponds to the decision where P_{miss} equals P_{fa}. Official results of NIST LRE'15 revealed that development and evaluation datasets have a severe mismatch for French cluster where the majority of participants obtained C_{avg} close to 50%. Therefore, we decided to exclude French cluster in this paper.

3. Advances in feature engineering

A key issue in language recognition is the effective representation of language cues embedded in speech utterances, i.e., the extraction of features with good descriptive and discriminative properties for language classification. A major paradigm shift was witnessed in the latest LRE'15 with the use of DNN for feature extraction. We presented below a comparative study of some new features that have emerged in the past few years and have shown to be effective for LRE'15. We use the same baseline system as described earlier for a more consistent comparison when different types of feature are used.

3.1. From SDC to bottleneck features

Commonly used in early language recognition systems, SDC features are obtained by stacking multiple time-shifted blocks of delta features. The base acoustic feature could be MFCC or PLP, though the former is more common. The success of SDC relies on the use of contextual information from a wide temporal window. For instance, the commonly used 7-1-3-7 configuration [7] incorporates 7 consecutive deltas with 3 frames apart. This amounts to a temporal context of 21 frames.

Table 3: Performance comparison of shifted-delta cepstral (SDC), deep bottleneck feature (DBF), stacked DBF, and phone log-likelihood ratio (PLLR).

Feature type	Actual Cost (%)	Min Cost (%)	EER (%)
SDC	30.14	29.89	30.07
PLLR	20.29	20.20	20.48
DBF	17.18	16.97	17.05
Stacked DBF	13.88	13.80	13.81

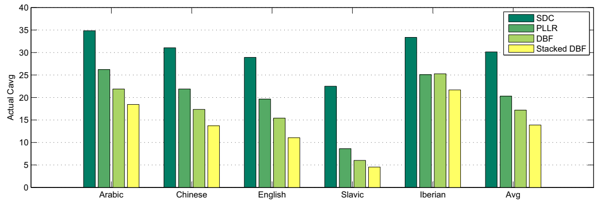

Figure 2: Performance comparison of various features in terms of actual cost for each language cluster.

Similar to that of SDC, bottleneck features are derived from a wide temporal context. In particular, bottleneck features are generated from a DNN in which one of the hidden layers has a smaller number of units (i.e., the bottleneck layer). In its common setup, the input layer takes in a stack of 10 to 20 frames, while the output nodes are set to predict tied states (i.e., senones) of context-dependent HMMs [18]. When input features are propagated to the output layers, the bottleneck creates a constriction in the network that forces the information into a low-dimensional representation. The linear outputs of the bottleneck layer give rise to the so-called deep bottleneck features (DBF) [18, 19, 20, 21]. In our implementation, the DNN consists of 7 hidden layers with a configuration of 2520-1024×5-64-1024-6111. The input to the DNN is the raw information rich spectral features (e.g., filter bank outputs). The bottleneck layer is the second to the last hidden layer with 64 linear units while all hidden layers use the ReLU activation function. The DNN was trained on Switchboard data (Table 2) using the KALDI toolkit [22].

As shown in Table 3, DBF exhibits significant advantage over the SDC feature, which has been the predominant option in LRE'11 and its predecessors. The relative improvement amounts to 43% on all the three performance metrics. It is also evident from Fig. 2 that DBF outperforms SDC across all language clusters. DBF benefits from the strength of deep architecture in modeling data correlation without the need of handcrafted transformation (e.g., the discrete cosine transform use in MFCC and PLP) which may causes loss of information embedded in the high dimensional inputs [4, 5, 6, 23].

3.2. Stacked bottleneck features

After the DNN training has been completed, all succeeding layers after the bottleneck are no longer required. In [24, 25, 26], the authors showed that it is beneficial to feed the DBF as inputs to a second DNN giving rise to the stacked bottleneck feature. A stacked-bottleneck approach can therefore provide an even wider temporal context than a single bottleneck DNN. Our implementation follows closely to that reported in [27, 28]. In particular, our stacked bottleneck features cover a temporal context of 5 frames in the first DNN and 10 frames (with a downsampling factor of 5) for the second DNN. From Table 3 and Fig. 2, it could be seen that stacked bottleneck feature is the clear winner. The relative improvement amounts to 19% compared to the DBF.

3.3. PLLR features

Another way to exploit the acoustic-phonetic information with a neural network is to use directly its output as features. This was first explored in [29] for speech recognition and more recently in [30, 31] for language recognition and in [32] for speaker recognition. For language recognition, the so-called temporal patterns neural network (TRAP/NN) was trained to predict the frame-based phone posteriors. Different from that of bottleneck features, the phone posteriors are probabilistic in nature. To overcome the non-Gaussian distribution of the posteriors, one solution is to transform the posterior probabilities into detection log-likelihood ratios [30]. Hence, the name phone log-likelihood ratio (PLLR).

One can see that the benefit of bottleneck feature is that the dimensionality is less dependent on the input temporal context and the size of the output targets. Given such flexibility and its good performance, bottleneck features should be the feature of choice for spoken language recognition in the years to come.

4. Discriminative training

Discriminative training has been widely used in spoken language recognition. The main reason is the closed-set nature of the task in which the set of target languages are known a priori. Even for an open-set scenario, out-of-set languages could be modeled as an additional class, by which a closed-set formulation could still be applied [1]. Some major parts of our efforts toward LRE'15 have been focusing on discriminative approach. In particular, we present below three approaches which we find having great potential for further development.

4.1. Pair-wise DNN post-processing

I-vector representation tends to be general in the sense it is not intentionally designed to segregate speaker and language information. This leads us to the idea of post-processing to derive new representations from the i-vectors. As shown in Fig. 3, each training sample consists of a pair of i-vectors and a 0 or 1 label. The label is 1 if two i-vectors are from the same language and 0 otherwise. The two i-vectors are processed separately by the post-processing subnets which consists of two hidden layers and one linear transform. The subnets are trained such that simple cosine distance is able to tell whether the two input i-vectors are from the same language. The training procedure follows closely of that reported in [33].

To create the training samples, both positive i-vector pairs (i.e., the two i-vectors are from the same languages) and negative pairs were randomly generated. On the LRE'15 Dev set, we created about 3 million training pairs. With this amount of data, the training usually converges after 10 to 20 epochs. After the training has been completed, the outputs of the subnet on the left are the new representation vectors to be used for language recognition. Fig. 4 shows the performance using the post-processed i-vectors in terms of the actual C_{avg} cost. Compared to the stacked bottleneck baseline, pair-wise training leads to a lower cost in language clusters where data imbalance is less serious. This is a point for future research.

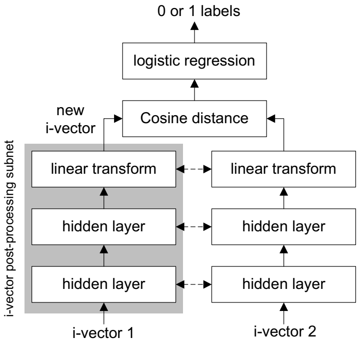

Figure 3: DNN pair-wise post-processing on i-vectors. The dotted lines indicate parameter tying.

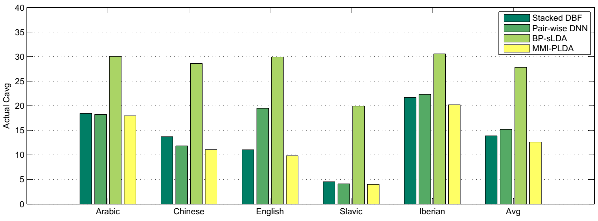

Figure 4: Performance comparison of classifiers in terms of actual cost for each language cluster.

4.2. Back propagation supervised LDA

Back propagation supervised latent Dirichlet allocation (BP-sLDA) is a deep feedforward network with a special layered structure designed for *supervised LDA* (sLDA) [34]. One important problem in sLDA is to infer topic distribution from the observed input data. Traditionally, either Gibbs sampling or variational approach was used to perform inference and learning for the model. In BP-sLDA, mirror descent algorithm is used to perform maximum a posteriori (MAP) inference of the topic distribution. The inference process can be described by the deep feed-forward network in Fig. 5, where the input of the network, x_d, is the bag-of-word vector of the document. To train the network, we use back propagation to compute the stochastic gradient, and then use stochastic gradient descent to update the model parameter. The blue arrows in Fig. 5 illustrate the back propagation process over the deep network.

The BP-sLDA network assumes the input vector consists of nonnegative components. For this reason, we applied exponential function to the i-vector to convert all the elements to have positive values, which is then fed into the BP-sLDA network. This strategy is suboptimal and might be the main reason why the BP-sLDA does not perform as well compared to the baseline as shown in Fig. 4. Using the senone or phone softcounts might be a better option.

4.3. MMI training in PLDA latent subspace

MMI training aims at increasing the discrimination abilities of a classifier by maximizing the posterior probabilities of correct class on the training data [35]. MMI training could be applied directly on the i-vector or after LDA projection as reported earlier in [36]. We explore a new way to apply MMI training through the use of probabilistic LDA (PLDA) consisting of the following steps:

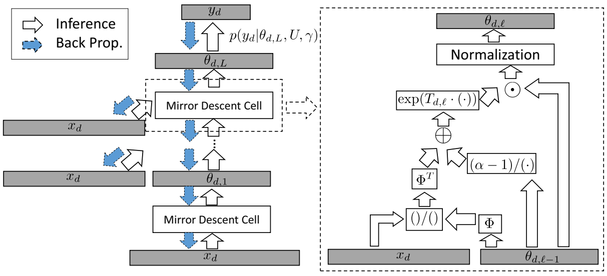

Figure 5: Deep feed-forward network for inferring the prediction variable in BP-sLDA [34]. Here, θ_d and y_d are the topic distribution and class label of the d-th sample, while U and Φ are the model parameters to be learned from the training data.

1. **Probabilistic projection to the latent space**: project an i-vector onto the low-dimensional language subspace by inferring the full posterior distribution.
2. **MMI training**: estimate and retrain the language-dependent mean vectors and covariance matrices to optimize for better separation between classes.
3. **Parameter lifting**: Lift the mean vectors and covariance matrices from the latent space back to the i-vector space.

Fig. 4 shows the performance of the MMI-PLDA method. Its performance is consistently better across all language clusters compared to other classifiers using the same stacked DBF. We refer the reader to [37] for more details.

5. Discussion and summary

The center theme of feature engineering, as presented in Section 2, has been focusing on using DNN for the extraction of acoustic-phonetic features. The benefit is the long temporal context at the input, non-linear transformation, and its capacity to take in huge amounts of training data. What is lacking at the current stage is the encoding of phonotactic cues, i.e., the phonological rules that govern the order of phones and their frequency in a language. Despite their superior performance in capturing acoustic-phonetic information, DBF, stacked DBF, or PLLR do not capture phonotactics. Traditionally, phonotactic information has been modeled as discrete events using a phone *n-gram* similar to the modeling of word sequences. We expect DNN to be applicable for modeling the underlying phonotactic constraints of languages, perhaps, with the use of *long short term memory* (LSTM) network. This remains an interesting question for future research.

State-of-the-art language recognition systems used in LRE'15 rely heavily on a pipeline of feature extraction followed by language classification. One drawback of this cascaded approach is that the feature representation might not be optimum for language detection task. The alternative approach would be to train a single neural network, the so called end-to-end system. In such a system, language labels will directly influence the feature extraction type, which is typically composed of a few stacked convolutional layers. Our exploration in this direction was reported in [38]. The key to our approach is the recurrent structure of DNN, which has recently shown to outperform the state-of-the-art DNN-HMM model in speech recognition [39]. In our experience, in order to design a successful end-to-end system for language recognition task the regularization of the network has to be very carefully designed so that network will generalize to an evaluation set.

6. References

[1] H. Z. Li, B. Ma, and K. A. Lee, "Spoken language recognition: from fundamentals to practice," *Proceedings of the IEEE*, vol. 101, pp. 1136–1159, 2013.

[2] "The 2011 NIST language recognition." [Online]. Available: http://www.nist.gov/itl/iad/mig/lre11.cfm

[3] "The 2015 NIST language recognition." [Online]. Available: http://www.nist.gov/itl/iad/mig/lre15.cfm

[4] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pretrained deep neural networks for large vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30 – 42, Jan. 2012.

[5] G. Hinton, L. Deng, D. Yu, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. S. abd George Dahl, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82 – 97, Nov. 2012.

[6] D. Yu and L. Deng, *Automatic Speech Recognition - A Deep Learning Approach*. Springer, Oct. 2014.

[7] P. Torres-Carrasquillo, E. Singer, M. Kohler, R. Greene, D. Reynolds, and J. D. Jr, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Proc. ICSLP*, 2002, pp. 89 – 92.

[8] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language identification: a tutorial," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82 – 108, 2011.

[9] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[10] D. Martínez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language recognition in ivectors space," in *Proc INTERSPEECH*, 2011.

[11] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, , and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc INTERSPEECH*, 2011.

[12] 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 IEEE ICASSP*, 2012.

[13] N. Brümmer, S. Cumani, O. Glembek, M. Karafiát, and P. Matějka, "Description and analysis of the Brno276 system for LRE2011," in *Proc Odyssey: the Speaker and Language Recognition Workshop*, 2012.

[14] A. Larcher, K. A. Lee, and S. Meignier, *SIDEKIT: an open source package for Speaker and Language recognition*. [Online]. Available: http://www-lium.univ-lemans.fr/sidekit/

[15] N. Brümmer and D. van Leeuwen, "On calibration of language recognition scores," in *Speaker Odyssey 2006*, 2006.

[16] N. Brümmer, *FoCal Multi-class: Toolkit for Evaluation, Fusion and Calibration of Multi-class Recognition Scores*. [Online]. Available: http://niko.brummer.googlepages.com/focalmulticlass

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

[18] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Proc. INTERSPEECH*, 2-11, pp. 237 – 240.

[19] Y. Song, B. Jiang, S. W. YeBo Bao, and L.-R. Dai, "I-vector representation based on bottleneck features for language identification," *Electronics Letters*, vol. 49, no. 24, pp. 1569 – 1570, 2013.

[20] B. Jiang, Y. Song, S. Wei, J.-H. Liu, I. V. McLoughlin, and L.-R. Dai, "Deep bottleneck features for spoken language identification," *PLOS One*, vol. 9, no. 7, 2014.

[21] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, pp. 1671–1675, 2015.

[22] 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. Vesely, "The Kaldi speech recognition toolkit," in *Proc IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[23] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Z. X. He, J. Williams, Y. Gong, , and A. Acero, "Recent advances in deep learning for speech research at Microsoft," in *Proc IEEE ICASSP*, 2013, pp. 8604 –8608.

[24] F. Grezl and M. Karafiat, "Semi-supervised bootstrapping approach for neural network feature extractor training," in *Proc IEEE ASRU*, 2013, pp. 470 – 475.

[25] F. Grezl, M. Karafiat, and K. Vesely, "Adaptation of multilingual stacked bottle-neck neural network structure for new language," in *Proc. IEEE ICASSP 2014*, 2014, pp. 7654–7658.

[26] P. Matejka, L. Zhang, T. Ng, S. H. Mallidi, J. M. O. Glembek, and B. Zhang, "Neural network bottleneck features for language identification," in *Proc Odyssey: the Speaker and Language Recognition Workshop*, 2014, pp. 299 – 304.

[27] H. H. Xu, H. Su, E. S. Chng, and H. Z. Li, "Semi-supervised training for bottle-neck feature based DNN-HMM hybrid systems," in *Proc. of Interspeech 2014*, Singapore, Sep. 2014.

[28] H. H. Xu, V. H. Do, X. Xiao, and E. S. Chng, "A comparative study of BNF and DNN multilingual training on cross-lingual low-resource speech recognition," in *Proc. of Interspeech 2015*, Dresden, Germany, Sep. 2015.

[29] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist features extraction for conventional HMM systems," in *Proc. IEEE ICASSP*, 2000, pp. 1635 – 1638.

[30] M. Diez, A. Varona, M. Penagarikano, L. Rodriguez-Fuentes, and G. Bordel, "On the use of log-likelihood ratios as features in spoken language recognition," in *Proc SLT*, 2012, pp. 274 – 279.

[31] O. Plchot, M. Diez, M. Soufifar, and L. Burget, "PLLR features in language recognition system for RATS," in *Proc INTERSPEECH*, 2014, pp. 3047 – 3051.

[32] M. Diez, A. Varona, M. Penagarikano, L. Rodriguez-Fuentes, and G. Bordel, "Using phone log-likelihood ratios as features for speaker recognition," in *Proc INTERSPEECH*, 2013.

[33] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc CVPR*, 2014.

[34] J. Chen, J. He, Y. Shen, L. Xiao, X. He, J. Gao, X. Song, and L. Deng, "End-to-end learning of LDA by mirror-descent back propagation over a deep architecture," in *Proc of NIPS*, 2015.

[35] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge,, 2005.

[36] A. McCree, "Multiclass discriminative training of ivector language recognition," in *Proc Odyssey: the Speaker and Language Recognition Workshop*, 2014.

[37] A. Sizov, K. A. Lee, and T. Kinnunen, "Discriminating languages in a probabilistic latent subspace," in *Odyssey: the Speaker and Language Recognition Workshop*, 2016, accepted.

[38] T. N. Trong, V. Hautamäki, and K. A. Lee, "Deep language: a comprehensive deep learning approach to end-to-end language recognition," in *Odyssey: the Speaker and Language Recognition Workshop*, 2016, accepted.

[39] D. Amodei, R. Anubhai, E. Battenberg *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," *CoRR*, vol. abs/1512.02595, 2015.