ADVERSARIAL MULTI-TASK DEEP FEATURES AND UNSUPERVISED BACK-END ADAPTATION FOR LANGUAGE RECOGNITION

Zhiyuan Peng[†*], *Siyuan Feng*[†*], *Tan Lee*[†]

[†]Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China

jerrypeng1937@gmail.com, siyuanfeng@link.cuhk.edu.hk, tanlee@ee.cuhk.edu.hk

ABSTRACT

This paper presents an investigation into speaker-invariant feature learning and domain adaptation for language recognition (LR) with short utterances. While following the conventional design of i-vector front-end and probabilistic linear discriminant analysis (PLDA) back-end, we propose to apply speaker adversarial multi-task learning (AMTL) to aim explicitly at learning speaker-invariant multilingual bottleneck features and perform unsupervised PLDA adaptation to alleviate performance degradation caused by domain mismatch between training and test data. Through a demo experiment, we show the adverse effect of domain mismatch and motivate the necessity of domain adaptation. LR experiments are carried out with the AP17-OLR challenge dataset to evaluate the effectiveness of the proposed methods in comparison with the state of the art. The results show that both speaker AMTL and unsupervised PLDA adaptation contribute significantly to performance improvement on the short-duration LR task. The effectiveness of PLDA adaptation is found to be insensitive to the number of clusters assumed in unsupervised data labeling. Our best system outperforms the state-of-the-art system of AP17-OLR and shows relative improvements of 6.98% in terms of C_{avg} and 4.80% in terms of EER on 1-second test set.

Index Terms— Language recognition, domain mismatch, unsupervised adaptation, adversarial learning

1. INTRODUCTION

Language recognition (LR) refers to the problem of automatically determining which language is being spoken in a given speech utterance. LR has many practical applications in multilingual and multimedia human-computer interaction and information processing. In the past decade, most LR systems were based on the i-vector approach [1, 2], i.e., training and extracting per-utterance i-vector representations [3], followed by classification with a back-end model [4]. The success of deep neural network (DNN) models in ASR [5] and speaker recognition (SR) [6, 7] has motivated extensive studies on applying DNN to LR [8–19]. One of the representative ideas is to extract phonetic bottleneck features (BNFs) from a DNN-based acoustic model and use BNFs to replace conventional spectral features for i-vector training [9, 10, 12]. Another idea is to train a DNN to classify language identities, followed by computing utterance-level embeddings as fixed-length representations of variable-length utterances [13, 14, 16, 18]. The fixed-length embeddings could be realized either by adding average pooling layer(s) in the DNN or via a recurrent neural network (RNN), similar to previous works on SR [6, 20]. This approach bypasses the process of i-vector training, though a back-end model is needed to produce LR results. Most recently, end-to-end (E2E) approaches to LR have been attempted enthusiastically [11, 17, 19]. In an E2E LR system, feature extraction and back-end classifier are jointly trained to minimize the error of language classification. E2E approaches have demonstrated great potential to outperform i-vector based approaches, especially when data augmentation methods are incorporated [19]. Nevertheless, for applications with limited training data, i-vector systems are more preferred [16]. The present study, as illustrated in Fig. 1, addresses the LR problem within the framework of i-vector front-end and probabilistic linear discriminant analysis (PLDA) back-end. The proposed system differs from previous designs in two aspects: (1) front-end frame-level feature representation learning; (2) back-end PLDA adaptation.

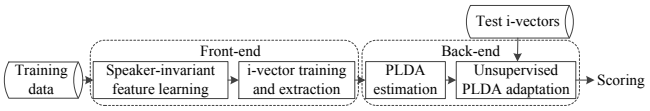

Fig. 1. System framework in this work.

First, speaker-invariant BNF learning is proposed to achieve improved feature representation for i-vector training in the front-end. Many previous studies showed that phonetically-discriminative BNFs [9, 10] and their multilingual variants [12] outperform spectral features for i-vector training. This is partially explained by that the BNFs optimized for ASR senone classification contain less linguistically-irrelevant information, e.g., speaker change. Motivated by this, we propose to apply speaker adversarial multi-task learning (AMTL) [21] to aim explicitly at learning speaker-invariant features. AMTL was applied first to robust ASR [22], and later to speaker adaptation [23], accent adaptation [24] and domain adaptation [25]. The basic idea is to introduce an adversarial speaker classification network on top of the bottleneck layer in the senone classification network, forcing the output representation of bottleneck layer to be speaker-invariant. The learned deep features can be regarded as a special type of phonetic BNFs, as they are not only phonetically-discriminative but also speaker-invariant. To the best of our knowledge, there was no previous attempt of applying speaker AMTL to LR.

Second, unsupervised PLDA adaptation is applied to alleviate performance degradation caused by domain mismatch between training and test data. Commonly adopted back-end models in LR systems include PLDA [10] and Gaussian linear classifier [1]. While they generally perform well in a domain-matched scenario, they may suffer severe performance degradation if test utterances are

*Equal contribution. This research is partially supported by a GRF project grant (Ref: CUHK14227216) from the Hong Kong Research Grants Council and a direct grant from Research Committee of the Chinese University of Hong Kong.

recorded under different conditions from training data [26]. Methods of mismatch compensation developed originally for SR [27–31] are expected to be applicable to LR. Meanwhile, a simple and effective approach named unsupervised PLDA adaptation [31] uses a domain-mismatched PLDA to perform test i-vector clustering. With these cluster labels, a new domain-matched PLDA is obtained as the in-domain back-end. The present work follows [31] with some modifications to better fit the LR problem.

Although the two aforementioned contributions are made within the i-vector framework, they are also applicable and may potentially be beneficial to DNN embedding-based LR frameworks, which have been actively investigated in the recent past [16, 18, 19].

2. SPEAKER-INVARIANT FEATURES FOR I-VECTOR TRAINING

2.1. Speaker Adversarial multi-task learning

Adversarial multi-task learning (AMTL) was first proposed by Ganin et al. [21] for unsupervised domain adaptation. In our work, AMTL is applied to the problem of learning speaker-invariant and phonetically-discriminative feature representation for i-vector training.

Fig. 2 shows the architecture of a speaker AMTL-DNN. It comprises three sub-networks, namely the shared-hidden-layer feature extractor (M_h), the senone classifier (M_y), and the speaker classifier (M_s). This architecture is similar to the MTL-DNN [32] used in multilingual ASR [33–35]. The major difference of AMTL from MTL is on how learning error is propagated from M_s to M_h. In AMTL, the error is reversely propagated such that the output layer of M_h is forced to learn speaker-invariant features so as to confuse M_s, while M_s tries to correctly classify features into their corresponding speakers. At the same time, M_y learns to predict the senone identities of input features, and back-propagates errors to M_h in a usual way. After training, the feature representation learnt by M_h is expected to be both phonetically-discriminative and speaker-invariant. By arranging a low-dimensional linearly-activated layer at the output of M_h, BNFs can be obtained for subsequent i-vector training.

Let θ_h, θ_y and θ_s denote the network parameters of M_h, M_y and M_s, respectively. With the stochastic gradient descent (SGD) algorithm, these parameters are updated as,

$$\theta_y \leftarrow \theta_y - \delta \frac{\partial \mathcal{L}_y}{\partial \theta_y}, \tag{1}$$

$$\theta_s \leftarrow \theta_s - \delta \frac{\partial \mathcal{L}_s}{\partial \theta_s}, \tag{2}$$

$$\theta_h \leftarrow \theta_h - \delta \Big[\frac{\partial \mathcal{L}_y}{\partial \theta_h} - \lambda \frac{\partial \mathcal{L}_s}{\partial \theta_h} \Big], \tag{3}$$

where δ is the learning rate, $\mathcal{L}_y$ and $\mathcal{L}_s$ are the loss values of senone and speaker classification tasks respectively, both in terms of cross-entropy (CE). λ denotes the *adversarial weight*, which controls the trade-off of training losses between M_y and M_s.

Training data of an LR task usually come from multiple languages. An intuitive approach is to build multiple language-dependent output layers and train the senone classifier M_y as in [33]. Though feasible, this approach would lead to an undesirably large number of model parameters in M_y, especially if there are a large number of target languages. For certain tasks of LR, e.g., the AP17-OLR challenge [36], transcriptions and/or lexicons are not well supported, in contrary to the case of large-vocabulary ASR. In this study, an out-of-domain (OOD) phone recognizer is utilized to generate senone labels for M_y training. In this way, the model size of M_y is controlled and fixed.

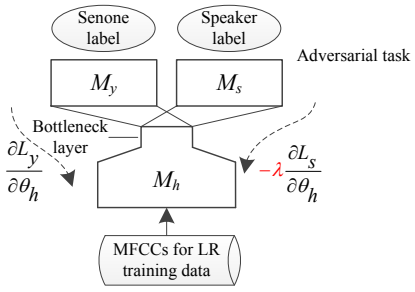

Fig. 2. Speaker AMTL-DNN architecture.

2.2. GMM-UBM/i-vector with speaker-invariant BNFs

With the speaker-invariant and phonetically-discriminative BNFs described above, conventional GMM-UBM i-vector training pipeline as proposed in [3] can be applied to convert variable-length utterances into low-dimensional and fixed-length i-vector representations. Details of GMM-UBM/i-vector estimation algorithms can be found in [3].

3. UNSUPERVISED ADAPTATION OF PLDA

The back-end being investigated here employs the simplified PLDA [4]. Unsupervised adaptation of PLDA parameters [31] is applied to alleviate the domain mismatch between training and test data.

3.1. PLDA back-end

PLDA model parameters are estimated from whitened and length-normalized [4] i-vector representations of training utterances and their ground-truth language labels. Given an utterance j of language i, PLDA assumes the corresponding i-vector $\boldsymbol{\omega_{ij}}$ is generated as,

$$\begin{aligned} \boldsymbol{\omega_{ij}} &= \boldsymbol{\mu} + \mathbf{F}\mathbf{h_i} + \boldsymbol{\epsilon_{ij}}, \\ \mathbf{h_i} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \boldsymbol{\epsilon_{ij}} &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \end{aligned} \tag{4}$$

where $\boldsymbol{\omega_{ij}} \in \mathbb{R}^D$, $\mathbf{F} \in \mathbb{R}^{D\times P}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$. Columns of the D-by-P matrix $\mathbf{F}$ provide the basis for the language-specific subspace, or 'eigen-language', by imitating the terminologies in [4]. P denotes the dimension of this subspace. A reasonable P to be smaller than the number of classes, i.e., the number of languages in our case. According to Eq. (4), each i-vector is drawn from the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}+\mathbf{F}\mathbf{F}^T)$, where $\boldsymbol{\mu}$ is global mean and can be precomputed and removed. $\boldsymbol{\Sigma}$ and $\mathbf{F}\mathbf{F}^T$ represent within- and between-class variability. These parameters can be estimated by an EM algorithm as described in [37].

Given a test i-vector $\boldsymbol{\omega_t}$ and language i, PLDA gives a similarity score by computing the log likelihood ratio (LLR) as,

$$\mathcal{R}(\boldsymbol{\omega_t}, i) = \log \frac{p(\overline{\boldsymbol{\omega^i}}, \boldsymbol{\omega_t} | \mathbf{F}\mathbf{F}^\intercal, \boldsymbol{\Lambda})}{p(\overline{\boldsymbol{\omega^i}} | \mathbf{F}\mathbf{F}^\intercal, \boldsymbol{\Lambda}) p(\boldsymbol{\omega_t} | \mathbf{F}\mathbf{F}^\intercal, \boldsymbol{\Lambda})}, \tag{5}$$

where $\overline{\boldsymbol{\omega^i}}$ is the average of training i-vectors that belong to language i. Details of the LLR computation can be found in [38].

3.2. Unsupervised PLDA adaptation

An unsupervised PLDA adaptation method is applied to alleviate performance degradation caused by the mismatch between training and test domains. This method was proposed in [31] for SR tasks. It is modified in the present study to fit the LR problem. The idea of PLDA adaptation is to leverage test (in-domain) i-vectors for adapting parameters $\{\mathbf{F_0}, \mathbf{\Lambda_0}\}$, which are estimated from training (out-of-domain) data. Lacking labelled in-domain data poses a major problem. By assigning a label to each test i-vector through a clustering process, a new domain-adapted PLDA model $\{\mathbf{F_{ad}}, \mathbf{\Lambda_{ad}}\}$ can be estimated.

We use a bottom-up agglomerative hierarchical clustering (AHC) algorithm here. The algorithm starts by treating each input pattern as an initial cluster, and proceeds by merging similar clusters based on a pre-defined distance measure. Following [31], the distance between a pair of i-vectors $\boldsymbol{\eta_1}$ and $\boldsymbol{\eta_2}$ is defined as,

$$d(\boldsymbol{\eta_1}, \boldsymbol{\eta_2}) = -\log \frac{p(\boldsymbol{\eta_1}, \boldsymbol{\eta_2} | \mathbf{F_0}\mathbf{F_0}^\intercal, \mathbf{\Lambda_0})}{p(\boldsymbol{\eta_1} | \mathbf{F_0}\mathbf{F_0}^\intercal, \mathbf{\Lambda_0}) p(\boldsymbol{\eta_2} | \mathbf{F_0}\mathbf{F_0}^\intercal, \mathbf{\Lambda_0})}. \quad (6)$$

The complete-linkage criterion is chosen as the distance measure between two clusters, i.e., the maximum inter-cluster pair-wise i-vector distance. The stopping criterion is based on a pre-defined cluster number. After clustering, cluster labels assigned to test i-vectors serve as supervision for in-domain PLDA estimation. The in-domain PLDA will be used for final scoring according to Eq. (5). Note that unlike [31], this work does not employ parameter interpolation.

4. AP17-OLR TASK DESCRIPTION

4.1. Dataset and evaluation metric

LR experiments in this study are carried out on the dataset provided for the second Oriental Language Recognition (OLR) Challenge held at the APSIPA ASC 2017 (AP17-OLR) [36]. The dataset covers 10 oriental languages, each with about 10-hour speech recorded by mobile phones. The training set consists of $54,266$ utterances with total length of about 79 hours. Test utterances are divided into three groups: 1 second, 3 second and full duration. Our work is focused on the 1 second test condition. The development set for this condition (dev_1s) contains $17,948$ utterances, 5 hours in duration. The test set (test_1s) contains $22,051$ utterances, 6 hours in duration. More details about the dataset could be found in [36].

The primary evaluation metric of the AP17-OLR challenge is C_{avg}, which is defined as,

$$C_{avg} = \frac{1}{N} \sum_{L_t} 0.5 \cdot [P_{MS}(L_t) + \frac{1}{N-1} \sum_{L_n} P_{FA}(L_t, L_n)], \quad (7)$$

where N is the number of languages, L_t and L_n denote the target and non-target languages, P_{MS} and P_{FA} are the missing and false alarm probabilities. In addition, equal error rate (EER) metric is also evaluated.

4.2. Measuring training and development set mismatch

According to the organizer of the AP17-OLR Challenge, there exists noticeable domain mismatch between training and development/test data [36]. To gain a better understanding about the mismatch, a demo experiment is carried out as described below.

A subset of training data (12 hours), denoted as **pseudo-dev**, is randomly selected and designated as the *domain-match* evaluation set, while dev_1s is regarded as as the *domain-mismatch* evaluation set. The remaining 67-hour training data are denoted as **training-part**. There is no overlap of speakers between training-part and pseudo-dev. Utterances of training-part and pseudo-dev sets are optionally trimmed to 1 second in duration. If trimming is applied, the datasets are denoted as **training-part_1s** and **pseudo-dev_1s**. A multi-layer perception (MLP) classifier is adopted as the back-end to map each i-vector to the respective language identity. The MLP has only one ReLU layer with 512 neurons before softmax output layer, and is trained using the cross-entropy criterion. A GMM-UBM i-vector extractor is trained with 60-dimensional voiced MFCCs+Δ+$\Delta\Delta$ without CMVN for training-part or training-part_1s, resulting in 100-dimensional i-vectors. A full covariance UBM with 256 Gaussian mixtures is estimated beforehand. After training, the MLP back-end is evaluated on pseudo-dev(_1s) and dev_1s to obtain C_{avg} and EER results as listed in Table 1.

Table 1. C_{avg}/EER% results of the demo experiment.

Training data	Pseudo-dev	Pseudo-dev_1s	Dev_1s
Training-part	3.50/3.97	7.78/9.56	13.42/13.18
Training-part_1s	—	7.61/8.94	14.01/13.88

We compare results on pseudo-dev and pseudo-dev_1s with training-part as the training data, the performance gap manifests the difficulty caused by short duration of test utterances. The drastic degradation from pseudo-dev_1s to dev_1s suggests that a large part of the mismatch is not related to the utterance duration, especially when the training utterances are trimmed to 1 second long. This confirms the necessity of domain adaptation for short-duration LR, which is the main motivation of our work.

5. EXPERIMENTAL SETUP

5.1. Front-end

For the learning of speaker-invariant and phonetically-discriminative feature representation, 40-dimensional MFCCs without cepstral truncation and CMVN are used as the input to AMTL-DNN as shown in Fig. 2. The feature extractor M_h is a time-delay NN (TDNN) with contextual configuration as stated in Table 2. All hidden layers except the linear bottleneck layer are activated with ReLU. For both senone and speaker classifiers M_y and M_s, the network comprises a 1024-neuron ReLU layer followed by a softmax output layer. To obtain the senone labels for training of M_y, a language-mismatched (Czech) phone recognizer [39] is used for decoding and generating state-level alignment of training data. The total number of states is 135. The speaker labels are obtained from per-utterance speaker information provided in training data. The total number of speakers in training set is 641. Four adversarial weight values λ are tested: $\{0, 0.125, 0.250, 0.375\}$. Note that $\lambda = 0$ is equivalent to training the DNN without adversarial learning. The minibatch size is 256. The learning rate starts from $1.5 \cdot 10^{-3}$ to $1.5 \cdot 10^{-4}$ with exponential decay. The number of training epochs and iterations are 2 and 90, respectively.

After AMTL-DNN training, 64-dimensional BNFs for voiced training frames are extracted for GMM-UBM i-vector training. A full-covariance 2048-mixture UBM and a 400-dimensional i-vector extractor are estimated, by which i-vectors for training, dev_1s and test_1s are extracted. Experiments are implemented in Kaldi [40].

Table 2. Contextual configuration of M_h.

Layer	Layer context	#Neurons
1	$\{-2,-1,0,1,2\}$	1024
2	$\{0\}$	1024
3	$\{-1,2\}$	1024
4	$\{-3,-3\}$	1024
5	$\{-7,-2\}$	1024
6	$\{0\}$	64

5.2. Back-end

An out-of-domain PLDA back-end (PLDA#1) is estimated on centered, length-normalized and within-class covariance normalized (WCCN) training i-vectors. The language subspace dimension P is 9, as there are 10 languages inside training set. The model is estimated for 10 iterations. Subsequently, unsupervised adaptation is applied using dev_1s i-vectors as adaptation data. To generate dev_1s labels, AHC algorithm with complete-linkage criterion is adopted to perform i-vector clustering towards dev_1s set. Pair-wise distance is defined in Equation (6). The stopping criterion for clustering is based on a pre-defined cluster number. In this work, five cluster numbers are tested, i.e. $\{10, 50, 100, 200, 500\}$. After clustering, each dev_1s i-vector is assigned with a cluster label, with which an in-domain PLDA (PLDA#2) is estimated. Same as PLDA#1, PLDA#2 is estimated on centered, length-normalized and WCCN i-vectors for 10 iterations, with subspace dimension 9.

6. RESULTS AND ANALYSES

6.1. Effectiveness of speaker-invariant BNFs

The effectiveness of speaker-invariant BNFs is evaluated on dev_1s set. A simple cosine similarity scoring is used so that the LR results reflect mainly the front-end features. Fig. 3 plots the C_{avg} and EER given by speaker AMTL-DNN BNFs with varying adversarial weight λ. By increasing λ from 0 to 0.250, both C_{avg} and

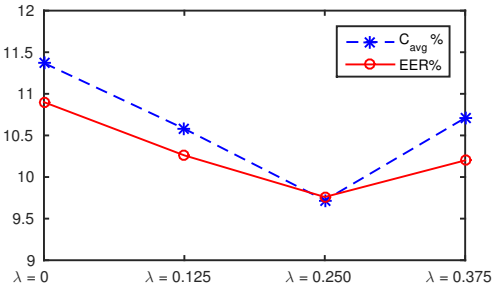

Fig. 3. C_{avg}/EER% results by employing speaker AMTL-DNN BNFs on dev_1s. Back-end is cosine scoring.

EER show improvements. $\lambda = 0.250$ appears to be the optimum. There is a relative improvement of 14.5% for C_{avg}, with respect to $\lambda = 0$. This demonstrates the effectiveness of applying adversarial speaker classification to suppress speaker variation in phonetic BNFs. Consequently i-vectors trained from the new features encapsulate the total variability subspace that are more speaker-irrelevant. This front-end improvement is expected to benefit not only cosine scoring back-end model but also more advanced models, such as PLDA with unsupervised adaptation adopted in this work.

6.2. Effectiveness of unsupervised PLDA adaptation

To evaluate the unsupervised PLDA adaptation back-end, we fix the front-end architecture of speaker AMTL with $\lambda = 0.250$. C_{avg} and EER with and without applying the adaptation method are summarized as in Table 3 and 4.

Table 3. C_{avg}% results with/without unsupervised PLDA adaptation. Back-end is PLDA.

	No Adapt.	Adapt. with cluster number					SOTA [41]
		10	50	100	200	500	
Dev_1s	8.25	6.68	6.61	**6.47**	7.07	7.45	N/A
Test_1s	9.46	—	—	7.36	—	—	7.65

Table 4. EER% results with/without unsupervised PLDA adaptation. Back-end is PLDA.

	No Adapt.	Adapt. with cluster number					SOTA [41]
		10	50	100	200	500	
Dev_1s	7.56	6.84	6.65	**6.49**	6.99	7.26	N/A
Test_1s	8.78	—	—	7.53	—	—	7.91

Applying unsupervised adaptation in the PLDA back-end leads to consistent improvement on LR performance, as compared to that without adaptation. The results indicate the importance of reducing domain mismatch for short-duration LR. It is also noted that the improved performance is relatively insensitive to the number of clusters. With 100 clusters, the proposed system achieves the best performance on dev_1s set, i.e., C_{avg} of 6.47% and EER of 6.49%, which exceeds the system without adaptation by absolute 1.8% in C_{avg} and 1.1% in EER.

Our best system is compared with the state of the art (SOTA) on test_1s, the designated evaluation set of the AP17-OLR challenge. As shown in Table 3 and 4, our system outperforms SOTA [41] in both C_{avg} and EER. The system without applying back-end adaptation does not perform as well as SOTA.

7. CONCLUSIONS

This paper addresses the problem of short-duration language recognition (LR), especially when there is significant domain mismatch between training and test data. In the front-end, speaker adversarial multi-task learning (AMTL) is applied to learn speaker-invariant multilingual BNFs. In the back-end, unsupervised PLDA adaptation is adopted to alleviate the performance degradation caused by domain mismatch between training and test data. Through a demo experiment, we show the adverse effect of domain mismatch and motivate the necessity of domain adaptation. LR experiments are carried out with AP17-OLR challenge dataset. Experimental results show that both speaker AMTL and unsupervised PLDA adaptation contribute significantly to performance improvement on short-duration LR task. The effectiveness of PLDA adaptation is found to be insensitive to the number of clusters. Our best system outperforms the state-of-the-art system of AP17-OLR. In the future, we plan to apply the proposed methods to DNN embedding-based LR frameworks.

8. REFERENCES

[1] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language recognition in ivectors space," in *Proc. INTERSPEECH*, 2011, pp. 861–864.

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

[3] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. ASLP*, vol. 19, no. 4, pp. 788–798, 2011.

[4] 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.

[5] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[6] E. Variani, X. Lei, E. McDermott, I. Lopez-Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification." in *Proc. ICASSP*, 2014, pp. 4052–4056.

[7] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. ICASSP*, 2014, pp. 1695–1699.

[8] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *Proc. ICASSP*, 2014, pp. 5337–5341.

[9] P. Matejka, L. Zhang, T. Ng, H. S. Mallidi, O. Glembek, J. Ma *et al.*, "Neural network bottleneck features for language identification," in *Proc. Odyssey*, 2014, pp. 299–304.

[10] F. Richardson, D. A. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," in *Proc. INTERSPEECH*, 2015, pp. 1146–1150.

[11] A. Lozano-Diez, R. Zazo-Candil, J. Gonzalez-Dominguez, D. T. Toledano, and J. Gonzalez-Rodriguez, "An end-to-end approach to language identification in short utterances using convolutional neural networks," in *Proc. INTERSPEECH*, 2015, pp. 403–407.

[12] R. Fér, P. Matějka, F. Grézl, O. Plchot, and J. Černockỳ, "Multilingual bottleneck features for language recognition," in *Proc. INTERSPEECH*, 2015, pp. 389–393.

[13] R. Li, S. H. Mallidi, L. Burget, O. Plchot, and N. Dehak, "Exploiting hidden-layer responses of deep neural networks for language recognition," in *Proc. INTERSPEECH*, 2016, pp. 3265–3269.

[14] G. Gelly and J. Gauvain, "Spoken language identification using LSTM-based angular proximity," in *Proc. INTERSPEECH*, 2017, pp. 2566–2570.

[15] Z. Tang, D. Wang, Y. Chen, L. Li, and A. Abel, "Phonetic temporal neural model for language identification," *IEEE/ACM Trans. ASLP*, vol. 26, no. 1, pp. 134–144, 2018.

[16] A. Lozano-Diez, O. Plchot, P. Matejka, and J. Gonzalez-Rodriguez, "DNN based embeddings for language recognition," in *Proc. ICASSP*, 2018, pp. 5184–5188.

[17] S. Irtza, V. Sethu, E. Ambikairajah, and H. Li, "End-to-end hierarchical language identification system," in *Proc. ICASSP*, 2018, pp. 5199–5203.

[18] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors," in *Proc. Odyssey*, 2018, pp. 105–111.

[19] J. Villalba, N. Brummer, and N. Dehak, "End-to-end versus embedding neural networks for language recognition in mismatched conditions," in *Proc. Odyssey*, 2018, pp. 112–119.

[20] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. ICASSP*, 2016, pp. 5115–5119.

[21] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by back-propagation," in *Proc. ICML*, 2015, pp. 1180–1189.

[22] Y. Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition," in *Proc. INTERSPEECH*, 2016, pp. 2369–2372.

[23] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gong *et al.*, "Speaker-invariant training via adversarial learning," in *Proc. ICASSP*, 2018, pp. 5969–5973.

[24] S. Sun, C.-F. Yeh, M.-Y. Hwang, M. Ostendorf, and L. Xie, "Domain adversarial training for accented speech recognition," in *Proc. ICASSP*, 2018, pp. 4889–4893.

[25] Z. Meng, J. Li, Y. Gong, and B.-H. Juang, "Adversarial teacher-student learning for unsupervised domain adaptation," in *Proc. ICASSP*, 2018, pp. 5949–5953.

[26] M. Mclaren, D. Castn, and L. Ferrer, "Analyzing the effect of channel mismatch on the SRI language recognition evaluation 2015 system," in *Proc. Odyssey*, 2016, pp. 188–195.

[27] F. Bahmaninezhad and J. H. Hansen, "Compensation for domain mismatch in text-independent speaker recognition," in *Proc. INTERSPEECH*, 2018, pp. 1071–1075.

[28] A. Misra and J. H. Hansen, "Maximum-likelihood linear transformation for unsupervised domain adaptation in speaker verification," *IEEE/ACM TASLP*, vol. 26, no. 9, pp. 1549–1558, 2018.

[29] T. Hasan, R. Saeidi, J. H. Hansen, and D. A. Van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," in *Proc. ICASSP*, 2013, pp. 7663–7667.

[30] E. Singer and D. A. Reynolds, "Domain mismatch compensation for speaker recognition using a library of whiteners," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 2000–2003, 2015.

[31] D. Garcia-Romero, A. McCree, S. Shum, N. Brummer, and C. Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proc. Odyssey*, 2014, pp. 260–264.

[32] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1998, pp. 95–133.

[33] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. ICASSP*, 2013, pp. 7304–7308.

[34] S. Zhou, Y. Zhao, S. Xu, and B. Xu, "Multilingual recurrent neural networks with residual learning for low-resource speech recognition," in *Proc. INTERSPEECH*, 2017, pp. 704–708.

[35] S. Feng and T. Lee, "Improving cross-lingual knowledge transferability using multilingual TDNN-BLSTM with language-dependent pre-final layer," in *Proc. INTERSPEECH*, 2018, pp. 2439–2443.

[36] Z. Tang, D. Wang, Y. Chen, and Q. Chen, "AP17-OLR challenge: Data, plan, and baseline," in *Proc. APSIPA*, 2017, pp. 749–753.

[37] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. ICCV*, 2007, pp. 1–8.

[38] D. Garcia-Romero and A. McCree, "Subspace-constrained supervector PLDA for speaker verification." in *Proc. INTERSPEECH*, 2013, pp. 2479–2483.

[39] P. Schwarz, "Phoneme recognition based on long temporal context," Ph.D. dissertation, Brno University of Technology, Brno, Czech Republic, 2009.

[40] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[41] "AP17-OLR challenge results," accessed: 2018-10-22. [Online]. Available: http://cslt.riit.tsinghua.edu.cn/mediawiki/index.php/OLRChallenge2017