A Study on Low-resource Language Identification

Zhaodi Qi*, Yong Ma†*, Mingliang Gu*
* School of Physics and Electronic Engineering, Jiangsu Normal University, Xuzhou, China
† Kewen College, Jiangsu Normal University, Xuzhou, China
Corresponding Author E-mail: zdqi0707@163.com, may@jsnu.edu.cn, mlgu@jsnu.edu.cn

***Abstract*—Modern language identification (LID) systems require a large amount of data to train language-discriminative models, either statistical (e.g., i-vector) or neural (e.g., x-vector). Unfortunately, most of languages in the world have very limited accumulation of data resources, which result in limited performance on most languages.**

In this study, two approaches are investigated to deal with the LID task on low-resource languages. The first approach is data augmentation, which enlarges the data set by incorporating various distortions into the original data; and the second approach is multi-lingual bottleneck feature extraction, which extracts multiple sets of bottleneck features (BNF) based on speech recognition systems of multiple languages. Experiments conducted on both the i-vector and x-vector models demonstrated that the two approach are effective, and can obtain promising results on both in-domain data and out-of-domain data.

Index: low-resource, data augmentation, multi-lingual, bottleneck feature, language identification

I. Introduction

Low-resource spoken language identification (LID) is an urgent problem in the LID study now. For most of languages in the world, the speakers are very limited, and so the accumulation for these languages are rather limited. This leads to the low-resource problem when LID is applied to diverse languages. Typical low-resource languages include Indonesian, Kazakh and Tibetan.

Most modern LID systems are based on language-discriminative models, for example i-vector model [1], closely following the developments in the Speaker Identification (SID) [2]. More recently, Deep Neural Network (DNN) [3] have been proposed, for example x-vector model [4] and the PTN model [5]. It has been reported that in many cases the DNN-based models show superior performance compared to i-vector based LID techniques formulated using GMM-UBM framework [6]. However, all these models require a large amount of data; in the low-resource scenarios, the training data is often too limited to support large-scale models.

In order to adapt to low-resource scenarios, we need to adopt some other features which have more language discriminative information than raw acoustic features. The phone recognition followed by language modeling (PRLM) model [7] encourages us to build a feature extractor which can extract phonetic information. In [8], [9], some researchers have proposed an i-vector based on LID formulation using phonetic bottleneck features (BNFs) extracted from a neural network and proved that it has better performance compared with i-vector based on LID system using Shifted Delta Cepstra (SDC) features. However, few researchers have applied these methods to deal with the low-resource scenarios in the LID task, and few researchers have explored the performance of low-resource LID. The idea of using extra information to boost LID performance was also reported in the phone-aware modeling [10] and speaker-aware modeling [11].

In this paper, we investigate how to improve the performance of the low-resource scenarios in the LID task. As a preliminary study, we apply two systems: traditional i-vector system and x-vector system. The input features are mfcc and fbank features. Additionally, BNFs are also applied. We consider only two conditions: in-domain data and out-of-domain data. The methods are the data augmentation and multi-lingual bottleneck feature extraction.

- **Data augmentation** approach is enlarging the data resources for modelling. There are two kinds of data augmentation. One is additive technique, which uses 2-fold augmentation strategy that combines the original ”clear” training list with 1 additive noise of multiple noises. The other is combined technique, which uses 5-fold augmentation strategy that combines the original ”clean” training list with 4 augmented copies involving speed perturbation, volume perturbation, reverberation and noise.
- **Multi-lingual bottleneck feature extraction** approach extracts multiple sets of bottleneck features (BNF) based on speech recognition systems of multiple languages. There are three nets in this approach. The first net is combined by appending directly which is assume that different phonetic BNFs is related; The second net is inputting independently which is assume that there is no relevance with each other; And the last net is score fusion.

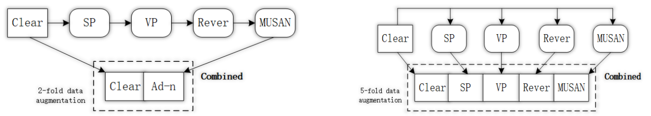
II. Methods

In this section, we describe data augmentation and multi-lingual bottleneck feature extraction.

A. Data augmentation

The data augmentation approach is inspired by [12], [13]. The main idea is enlarging the data resources for modelling. We use the additive technique and combined technique to explore the performance of data augmentation in low-resource LID task.

In Fig.1, we show two kinds of data augmentation. One is 2-fold augmentation strategy. we combine the original ”clear” training data with 1 additive noise. The 1 additive is synthesized the MUSAN (including Babble, Music and



(a) Additive technique (b) Combined technique

Fig. 1. Two combination techniques of data augmentation.

Noise) with the original ”clear” training data after speed perturbation, volume perturbation and reverberation as shown in Fig.1.(a). The other is 5-fold augmentation strategy. we combine the original ”clear” training data with 4 augmented copies which are obtained by speed perturbation, volume perturbation, reverberation and MUSAN respectively as shown in Fig.1.(b). The data augmentation strategy are as follows:

- Speed perturbation: apply 1.1 times or 0.9 times speed of the original recording.
- Volume perturbation: the scale of volume is chosen randomly between 0.125 and 2.
- Reverberation: the artificially reverberated data is convoluted with simulated RIRs.
- Babble: add the summation of the speech from several speakers randomly selected from MUSAN [12] to the original signal (13-20dB SN
- Music: add a randomly selected music file from MUSAN to the original signal (5-15dB SNR).
- Noise: add MUSAN noises every second throughout the recording (0-15dB SNR).

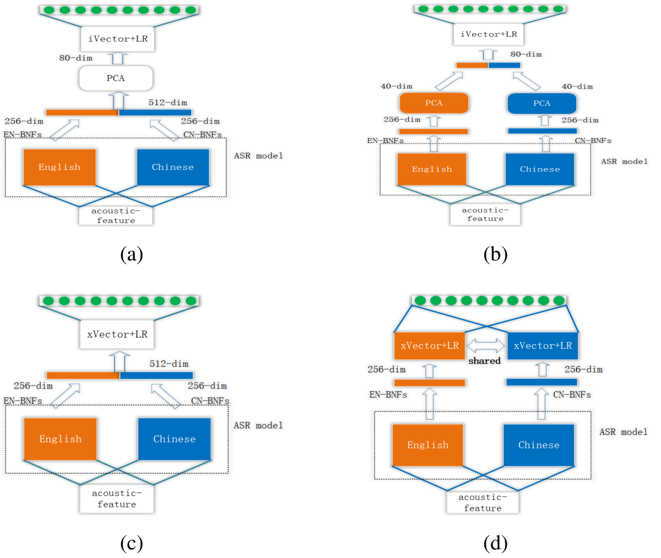
Speed perturbation uses a specified speed factor [14] to change the speed of the speech signal. Reverberation convolves room impulse responses (RIR) with audio. For additive noise, we use the MUSAN dataset, which consists of over 900 noises, 42 hours of music from various genres and 60 hours of speech from twelve languages [15]. Both MUSAN and the RIR datasets are from http://www.openslr.org.

B. *Multi-lingual bottleneck feature extraction*

Multi-lingual bottleneck feature extraction approach extracts multiple sets of bottleneck features (BNF) based on speech recognition systems of multiple languages. It is inspired by the complementarity of universal speech attributes and language-dependent phonemes.

The accuracy of phone recognizer is critical, but not the only factor for LID performance in the phone-aware approach. In other words, it is fine to model the phonemes in the language model based on the assumption of similarity between these two language if a phoneme of another language to be recognized is always recognized as the one in the phone set designed for the phone recognizer. It is quite common for spoken languages in different language families that the phonemes cannot be represented well in language modeling if some phonemes are very different from the language for phone recognizer. Meanwhile, it might be possible that some language ASR systems attribute to one aspect of the language space.

We relieve this problem by using attribute units which are potentially language-universal across all spoken languages. In this study, we show the complementary nature of speech attribute detectors to phone recognizers by combining the BNFs extracted from different ASR decoders with phones and attributes.



(a) (b) (c) (d)

Fig. 2. Several combination techniques of multi-lingual bottleneck feature extraction. Here (a)(c) shows that different phonetic BNFs is related, and (b)(d) shows that there is no relevance with each other.

In Fig.2, we use just two mono-lingual BNFs to fuse different ways in the preliminary study. In Fig.2.(a)(c), we append directly two BNFs by frame level. And in i-vector system, we reduce it from 512 dimensional to 80 dimensional. In Fig.1.(b), we first reduce the BNFs from 256 dimensional to 40 dimensional respectively, then append by frame level. At last the combined feature is the input feature of i-vector system. In Fig.2.(d), we input the two BNFs into the x-vector separately and parameters of the x-vector system are shared. Additionally, score fusion is also used in this study.

III. Experiments

We build several systems to deal with the low-resource scenarios in the LID task. All systems are built on the Kaldi speech recognition toolkit [16].

A. *Database*

*1) **Training data:*** The training data is taken from the 2018 Oriental Language Recognition (OLR) Competition, which was organized by Tsinghua University and SpeechOcean [17]. This competition has been arranged for three times, with the aim of promoting the research on LID techniques for oriental languages [17]–[19]. The data is provided by Haitian Ruisheng and contains 10 languages. The recordings are the traditional telephone channel with 16kHz, 16 bit, mono format. Each language is about 10 hours, and the gender ratio of men and women is 1:1. In order to emphasize the influence of data resources on language identification, this training data is divided into four data equilibrium quantity set, which are 25h, 50h, 75h, 106h, named train_25h, train_50h, train_75h and train_106h respectively in the experiment.

*2) **Evaluation data**:* Our evaluation consists of two distinct datasets: in-domain data and out-of-domain data. The **in-domain** data is the standard test data for AP18-OLR, which contains 10 closed languages, containing 1800 utterances each. The **out-of-domain** data is downloaded from the Internet, which involves 6 closed languages, each in a particular language. The six languages are: Mandarin, Japanese, Russian, Vietnamese,Tibetan and Uyghur. containing about 1800 utterances each. The channel of out-of-domain recordings is the video channel. Before extracting the features of the out-of-domain data speech segment, we converted the recordings to 16KHz, 16 bit, mono format.

The training data is 106.58h in length, the in-domain data is 34.05h in length, and the out-of-domain data is 15.71h in length.

B. Baseline

*1) **i-vector**:* Our acoustic-feature baseline system is a traditional i-vector system. This system is based on the GMM-UBM recipe described in [2]. The features are 13 mfccs with a frame-length of 25ms. They are mean normalized over a sliding window of up to 3 seconds. Delta and acceleration are appended to create 39 dimension feature vectors. An energy-based speech activity detection (VAD) system selects features corresponding to speech frames. The UBM is a 1024 component diagonal-covariance GMM. The system uses a 400 dimensional i-vector extractor.

*2) **x-vector**:* The x-vector system is based on a framework that developed for speaker recognition [20]. The recipe is based on the SRE16 v2 recipe available in the main branch of Kaldi as https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2. The feature learning component is a 5-layer time-delay neural network (TDNN). The statistic pooling layer computes the mean and standard deviation of the frame-level features from a speech segment. The size of the output layer is 10, corresponding to the number of languages in the training data. Once trained, the 512 dimensional activations of the penultimate hidden layer are read out as an x-vector.

C. Features

- **Acoustic features**: The acoustic features are 39 mfccs with a frame-length of 25ms in i-vector system and 40 fbanks in x-vector system.
- **English BNFs**: The English ASR model is trained by alignments provided by a standard chain model. 1300h data is used, and its input features are 40 fbanks. The ASR DNN has 11 layers, and its total left-context is 21 and right-context is 21. The softmax output layer computes posteriors for 5297 triphone states. Excluding the output layer, the DNN has 19.96 million parameters. we use 256 dimensional BNFs extracted from ASR DNN.
- **Chinese BNFs**: Chinese ASR model is trained with 3000h data, which architecture is same with English ASR model except that the posteriors is 5984 triphone states.

We reduce the BNFs from 256 dimensional to 40 dimensional by Principal Component Analysis (PCA) in the i-vector system. BNFs extractor trained by one language can learn features to recognize other language. It is important for language identification in low-resource scenarios.

IV. Result

The evaluation standard is the accuracy metric, defined as

$$Accuracy = \frac{L_T}{L_T + L_N} \times 100\% \qquad (1)$$

where L_T and L_N are target and non-target languages. In the following tables, scores of each data source (in-domain data or out-of-domain data) or language have been balanced and contribute equally to the metric.

A. Baseline

Our purpose is investigating on how to improve the performance of LID system in low-resource scenarios. we first implement two state-of-the-art which are i-vector systems and the x-vector baseline system without phone-aware information in different duration of training data and different channels of test set. Mfcc is the input feature of the system iVec_mfcc_lr. fbank is the input of the xVec_fbank_lr. And their back-end is Logistic regression (LR).

TABLE I
Comparing the accuracy of different durations of training data in in-domain and out-of-domain. All systems conform to the fixed training condition

Evaluation	System	25h	50h	75h	106h
in-domain	iVec_mfcc_lr	**71.43**	84.00	88.05	**90.62**
	xVec_fbank_lr	**61.70**	76.76	82.87	**86.34**
out-of-domain	iVec_mfcc_lr	31.94	37.28	40.15	37.51
	xVec_fbank_lr	31.05	35.56	37.76	35.48

In TABLE I, we find that the smaller the amount of training data, the lower the accuracy in the in-domain. And the accuracy of out-of-domain is much lower than in-domain on the same training data. In in-domain, compared train_106h with train_25h, the accuracy decreased sharply from 90.62% to 71.43% in i-vector system. And the accuracy decreased sharply from 86.34% to 61.70% in x-vector system. In i-vector system, compared in-domain with out-of-domain, the accuracy decreased sharply from 71.43% to 31.94% when training data is 25h, and the accuracy decreased sharply from 90.60% to 37.51% when training data is 106h. The results above demonstrate sufficiently the limited training degrading the performance of LID systems.

B. Data Augmentation

In this section, we test the performance of augmenting the i-vector and x-vector training data. The system iVec_mfcc_2f_lr uses 2-fold additive augmentation, and iVec_mfcc_5f_lr uses 5-fold combined augmentation. In the systems above, the input feature is mfcc. The name of x-vector is the same except fbank instead of the mfcc.

TABLE II
RESULTS USING DATA AUGMENTATION IN VARIOUS SYSTEMS

Evaluation	System	25h	50h	75h	106h
in-domain	iVec_mfcc_2f_lr	69.89	76.46	87.83	89.25
	iVec_mfcc_5f_lr	**72.52**	**86.54**	**90.09**	**91.83**
	xVec_fbank_2f_lr	60.51	75.98	80.12	83.31
	xVec_fbank_5f_lr	**62.05**	**76.89**	**83.07**	**89.89**
out-of-domain	iVec_mfcc_2f_lr	25.23	31.27	44.94	46.74
	iVec_mfcc_5f_lr	**43.31**	**44.61**	**44.86**	**45.36**
	xVec_fbank_2f_lr	25.12	28.98	35.29	37.35
	xVec_fbank_5f_lr	**33.57**	**36.43**	**37.97**	**43.73**

[1] 2f: 2-fold data augmentation strategy.
[2] 5f: 5-fold data augmentation strategy.

In TABLE II, we observe that augmentation using 2-fold significantly degrades in in-domain. The reason maybe is that the training data is heavily damaged by noise. And the performance without augmentation degrades significantly compared with 5-fold augmentation strategy. Due to augmentation increasing the amount of limited training data, the system is more robust against degraded audio. Through the results, we can see that data augmentation (e.g. 5-fold augmentation strategy) is effective for low-resource LID task when training data is not damaged (e.g. 2-fold augmentation strategy).

C. Mono-lingual BNFs

In this session, we first analysis the impact of a mono-lingual BNFs in this study. Then we compare the performance of different BNFs extraction layers on LID. Finally, we present the impact of BNFs on this task by T-SNE [21] visualizing on the x-vector system.

TABLE III
RESULTS USING ENGLISH BNFS AND CHINESE BNFS IN VARIOUS SYSTEMS

Evaluation	System	25h	50h	75h	106h
in-domain	iVec_enbnf_lr	93.62	95.38	97.38	97.61
	iVec_cnbnf_lr	**96.29**	**98.53**	**98.27**	**98.41**
	xVec_enbnf_lr	93.72	97.66	98.31	98.41
	xVec_cnbnf_lr	**96.81**	**98.53**	**98.65**	**98.91**
out-of-domain	iVec_enbnf_lr	**71.24**	**73.90**	**76.72**	**77.23**
	iVec_cnbnf_lr	67.22	69.30	70.80	71.70
	xVec_enbnf_lr	59.13	60.78	61.02	64.22
	xVec_cnbnf_lr	**64.51**	**64.53**	**66.40**	**68.99**

[1] enbnf: BNFs is extracted from English ASR model.
[2] cnbnf: BNFs is extracted from Chinese ASR model.

In TABLE III, mono-lingual BNFs are used in the experiment. In in-domain, it shows that the performance of Chinese ASR model is better than the English ASR model. Because the Chinese ASR model has more training data and the accuracy of phone recognizer is higher. But it must be considered that Chinese belongs to oriental languages while English is not. In out-of-domain, the performance is opposite in i-vector model. The results demonstrate that English BNFs is more robust against domain mismatch in i-vector system. In in-domain, mono-lingual BNFs is better in x-vector system. However, i-vector is more robust against domain mismatch, mono-lingual BNFs is better in i-vector system in out-of-domain. The accuracy of xVec_cnbnf_lr is nearly 57% better than the xVec_fbank in in-domain when training data is train_25h. In out-of-domain, the accuracy of iVec_enbnf is 123% better than iVec_mfcc in train_25h. It can be seen that the addition of phone-aware information greatly solves the low-resource problem, x-vector system with Chinese BNFs is better in in-domain, and i-vector system with English BNFs is better in out-of-domain. And i-vector system is better in out-of-domain, which is more robust against domain mismatch.

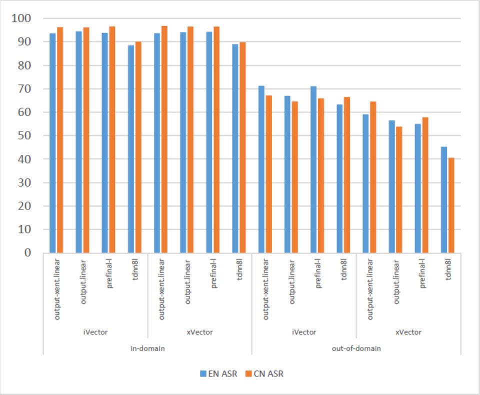


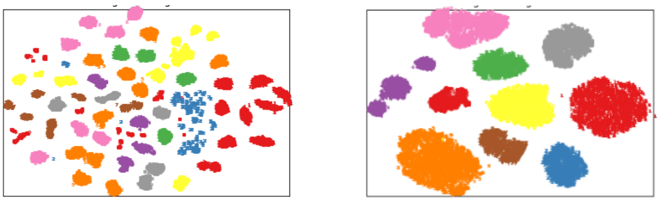
Fig. 3. The effect of varying the position of the BN layer in a ASR DNN when training data is 25h, under different ASR models

In Fig.3, we compare the effects of different extraction layers of different ASR models to the LID. Layer output-xent.linear denotes the last hidden layer of frame level. Layer output.linear denotes the last hidden layer of word level. Layer prefinal-l.linear denotes the penultimate layer. Layer tdnn8l.linear denotes the seventh from the end layer. Interestingly, it is not the last hidden layer giving the best performance in the in-domain data, but the layers nearer to the last. However, it is the last hidden layer of frame level giving the best performance in the out-of-domain data. Maybe this layer loses channel information.

T-SNE [21] is to visualize the language features in the 2 dimensional space. Fig.4 shows the impact of BNFs on x-vector system. In Fig.4.(a), it shows that the learned language features have intra-class divergence problem in x-vector system. In this paper, the BNFs which is phone-aware information is introduced, so that the linguistic features are compensated for the prior knowledge of the phoneme in the learning process to solve the problem of the volatility of the linguistic features caused by the pronunciation content and the speaker. BNFs makes each language more convergent and distinguishing in this task as shown in Fig.4.(b).

D. Multi-lingual BNFs

It has been well documented above that i-vector-based and x-vector-based LID systems all improve the accuracy greatly by using the mono-lingual BNFs. In this section, we show the performance of Multi-language BNFs by comparing systems trained on English BNFs (iVec_enbnf and xVec_enbnf) with Chinese BNFs (iVec_cnbnf and xVec_cnbnf).



(a) (b)

Fig. 4. The effect of BNFs on the distribution of the extracted features(best viewed in color). The figure shows t-sne visualizations of the x-vector embeddings (a) in case when xVector-fbank, (b) in case when xVector-BNF.

TABLE IV
COMPARING THE ACCURACY OF DIFFERENT DURATIONS OF TRAINING DATA IN IN-DOMAIN AND OUT-OF-DOMAIN. ALL SYSTEMS CONFORM TO THE FIXED TRAINING CONDITION

Evaluation	System	25h	50h	75h	106h
in-domain	iVec_fus1	**96.46**	**98.83**	**98.94**	**99.08**
	iVec_fus2	**96.41**	**98.12**	**98.71**	**98.83**
	iVec_fus3	**97.13**	**98.64**	**98.92**	**99.03**
	xVec_fus1	**96.95**	**98.67**	**99.01**	**99.56**
	xVec_fus2	93.88	97.58	98.30	98.50
	xVec_fus3	**97.62**	**98.98**	**98.99**	**99.06**
out-of-domain	iVec_fus1	64.96	71.12	73.24	75.27
	iVec_fus2	63.05	70.05	72.13	73.66
	iVec_fus3	**72.38**	**75.01**	**77.35**	**78.92**
	xVec_fus1	55.96	57.94	63.21	65.54
	xVec_fus2	58.68	61.60	63.63	64.19
	xVec_fus3	**65.02**	**65.22**	**68.95**	**70.14**

[1] fus1 is appending directly of two BNFs.
[2] fus2 is independent input of two BNFs.
[3] fus3 is shallow score fusion.

In TABLE IV, we find that in in-domain, both appending directly and input independently of two BNFs are much better than mono-lingual BNFs in i-vector system. And in both i-vector and x-vector system, fus1 is better than fus2 in in-domain, which is proved that the BNFs extracted by different ASR model are related. However, either fus1 or fus2 is lower than mono-lingual BNFs in out-of-domain. It proves that channel information is following phone-aware information in fusion of features level. In both in-domain and out-of-domain, score fusions are quite effective, and can obtain promising results in both the i-vector and x-vector models.

We find two main advantages with multi-lingual BNFs. Firstly, these units are defined universally across multiple languages [22]–[24]. As a result, it alleviates the problem missing phones in the front-end phone recognizer of PRLM systems [25]. It enhances the modeling capability by sharing of speech data from different languages. Secondly, different acoustic definitions often present complementary discrimination ability.

V. Conclusions

In this paper, two approaches are investigated to deal with the low-resource scenarios in the LID task. Firstly, we find that 5-fold augmentation is a good choice for LID task in data augmentation approach. Then, we extend mono-lingual phone-aware model to multi-lingual bottleneck feature extraction which is inspired by the advantage and complementarity of universal speech attributes and language-dependent phonemes. We find mono-lingual BNFs perform much better than acoustic features alone, while multi-lingual BNFs are the best choice. The experiments show that the two approaches are quite effective on both the i-vector and x-vector models and perform well in both the in-domain data and the out-of-domain data. When training data is train_25h, our best model improved accuracy by 36%, 127% for in-domain data, out-of-domain data in i-vector system, and by 58%, 109% for in-domain data, out-of-domain data in x-vector system. In the future, we will explore which is the best method for the multi-domain LID.

Acknowledgement

The paper is supported by Jiangsu University Natural Science Research Project 17KJB510018. The work was also supported by the National Natural Science Foundation of China under Projects 61633013.

References

[1] David Martinez, Oldřich Plchot, Lukáš Burget, Ondřej Glembek, and Pavel Matějka, "Language recognition in ivectors space," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[2] N. Dehak, P. J. 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.

[3] Fred Richardson, Douglas Reynolds, and Najim Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[4] David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "Spoken language recognition using x-vectors," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 105–111.

[5] Zhiyuan Tang, Dong Wang, Yixiang Chen, Lantian Li, and Andrew Abel, "Phonetic temporal neural model for language identification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 134–144, 2017.

[6] E Wong, J Pelecanos, S Myers, and S Sridharan, "Language identification using efficient gaussian mixture model analysis," in *Australian International Conference on Speech Science and Technology*, 2000, vol. 4, pp. 7–6.

[7] M. A. Zissman and E. Singer, "Automatic language identification of telephone speech messages using phoneme recognition and n-gram modeling," in *IEEE International Conference on Acoustics*, 2002, pp. 305–308.

[8] Yan Song, Bing Jiang, YeBo Bao, Si Wei, and Li-Rong Dai, "I-vector representation based on bottleneck features for language identification," *Electronics Letters*, vol. 49, no. 24, pp. 1569–1570, 2013.

[9] Liu Yi, He Liang, Liu Jia, and Michael T. Johnson, "Speaker embedding extraction with phonetic information," 2018.

[10] Zhiyuan Tang, Dong Wang, Yixiang Chen, Ying Shi, and Lantian Li, "Phone-aware neural language identification," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–6.

[11] Lantian Li, Zhiyuan Tang, Dong Wang, Andrew Abel, Yang Feng, and Shiyue Zhang, "Collaborative learning for language and speaker recognition," in *National Conference on Man-Machine Speech Communication*. Springer, 2017, pp. 58–69.

[12] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*. IEEE, 2017, pp. 5220–5224.

[13] Zoltn Tske, Pavel Golik, David Nolden, Ralf Schlter, and Hermann Ney, "Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages," in *Interspeech*, 2014.

[14] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *Computer Science*, 2015.

[16] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *Workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[17] Zhiyuan Tang, Dong Wang, and Qing Chen, "Ap18-olr challenge: Three tasks and their baselines," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 596–600.

[18] Dong Wang, Lantian Li, Difei Tang, and Qing Chen, "Ap16-ol7: A multilingual database for oriental languages and a language recognition baseline," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–5.

[19] Zhiyuan Tang, Dong Wang, Yixiang Chen, and Qing Chen, "Ap17-olr challenge: Data, plan, and baseline," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 749–753.

[20] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[21] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

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

[23] Sabato Marco Siniscalchi, Torbjorn Svendsen, and Chin-Hui Lee, "Toward a detector-based universal phone recognizer," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 4261–4264.

[24] Chin-Hui Lee, "From knowledge-ignorant to knowledge-rich modeling: A new speech research paradigm for next generation automatic speech recognition," in *Proc. ICSLP*, 2004, vol. 4.

[25] Marc A Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on speech and audio processing*, vol. 4, no. 1, pp. 31, 1996.