The GREAT System for the Far-Field Speaker Verification Challenge 2020

Zongze Ren, Zhiyong Chen, Shugong Xu

Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai, China

zongzeren@shu.edu.cn, bicbrv@shu.edu.cn, shugong@shu.edu.cn

Abstract

In this paper, we present the system developed by the Group of Research & Education in AI & Telecom (GREAT) team for the Far-Field Speaker Verification Challenge (FFSVC) 2020. We explore almost every part of the system pipeline, and our best single system contains elements of a combined training dataset, a deep neural network with specially designed layers and loss functions, score normalization component and so on. Moreover, we investigate some domain adaptation methods. The submitted system is based on the fusion of several deep learning subsystems. As the result, the fusion system achieves on the development set EER, minDCF of 5.21% and 0.555, and on the evaluation set EER, minDCF of 6.61% and 0.693, respectively in the task of far-field text-independent speaker verification from single microphone array (Task2). And the system also performs well in the other two tasks without any fine-tuning operations.

Index Terms: speaker verification, FFSVC 2020, domain adaptation

1. Introduction

Automatic speaker verification (ASV) is a technology to give a decision of whether two utterances said by one person. ASV can roughly split to two sub-tasks, text-dependent and text-independent, and the latter is more challenging cause it does not have any lexicon or pronunciation constraints. In past decades, researches on ASV have made great progress. Thanks to the rapid growth of deep learning methods, deep neural network (DNN) based systems attract more attention than i-vector[1] based statistical systems recently. Network building is critical for all deep learning based tasks, ResNet[2], x-vector[3] and E-TDNN[4] are widely used in ASV tasks for learning representation and training classifiers. Besides, many improvements have been made during the whole pipeline of ASV, example for pooling operation[5, 6], loss function[7, 8, 9] and back-end modeling[10, 11].

In real-world scenarios, ASV system often meets speaker utterances from complex recording environments, which leads challenging in applications such as access control. For example, situations of channel, device and its distance, background noise may affect the performance of ASV system. VOiCES from a Distance Challenge[12] consists of automatic speech recognition (ASR) and ASV tasks, which focus on single-channel far-field audio under various noisy conditions. Go a step further, the FFSVC 2020 [13] is designed to explore three ASV tasks under far-field distributed microphone arrays under noisy conditions. The tasks are Far-Field Text-Dependent Speaker Verification from single microphone array, Far-Field Text-Independent Speaker Verification from single microphone array and Far-Field Text-Dependent Speaker Verification from distributed microphone arrays.

Since the given training set has few speakers, more training data should be involved, which brings the domain gap to our work. Many works consider some semi-supervised or unsupervised domain adaption methods: gradient reversal layer (GRL)[14] or adversarial multi-task training[15]. But these target set does not have speaker labels, which is not similar to our work. To fully use the given data, we consider a more efficient domain adaption method with supervised learning.

To solve the FFSVC ASV tasks elegantly, we design the system pipeline that is suitable for all the three tasks. To sum up, the main contributions we make for FFSVC are the following:

1. To expand the size and diversity of training data, we combine datasets including FFSVC, Vox-celeb1&2 [16, 17] and CN-celeb[18], totally 8443 speakers for training the model.
2. We choose an E-TDNN framework with residual links and RNN layers, and use attention mechanism to make the network focus on speaker-dependent information.
3. To minimize the inference of the difference between training sets, we design a speaker-independent branch[1] to learn domain labels, which makes the speaker embeddings more speaker-dependent.
4. By score normalization and system fusion, we improve the performance from the original cosine similarity scores. Compared with the official baseline, we achieve better results with less training data.

The remainder of this paper is organized as follows. Section 2 shows our end-to-end speaker embedding system, including deep learning network, loss function and domain adaptation techniques. In Section 3, we introduce the setup of data usage, data augmentation and training details. Then the experimental development and evaluation results are reported and analysed in Section 4. Lastly, we give a conclusion and make a summary in Section 5.

2. System Descriptions

2.1. Network structure

In our work, we choose an Extended Time-Delay Neural Network (E-TDNN) as the framework to classify speakers. Compare with the classic x-vector framework, the E-TDNN is much deeper and comprises more context in the frame level. As table 1 shows, acoustic features are feed to the network, then high-representation of frames are extracted by several TDNN layers. At the same time, we put some residual links[19] among the frame-level layers to keep some low-rank information. Our previous work has shown that LSTM[20] layer can lead success in language recognition task[21], so an RNN layer is connected with the TDNN part. Then we make a pooling operation so

[1]This part is finished after the mid-term deadline (May 1^{st}), so we only report the development results of these experiments

that we can get the utterance level features from frame level. Batch normalization and ReLU non-linearity are deployed after each layer. And finally, we project the utterance level features to a classifier by two fully connected layers. The output of the E-TDNN is the posterior probabilities of the training speakers. In the evaluation phase, the speaker embeddings are extracted from the layer 14.

Table 1: *E-TDNN Framework with RNN layers*

ID	Layer Type	Input-node	Output-dim	Res Links
1	TDNN	t-2:t+2	512	
2	Dense	t	512	
3	TDNN	t-2,t,t+2	512	
4	Dense	t	522	
5	TDNN	t-3,t,t+3	512	3
6	Dense	t	512	
7	TDNN	t-4,t,t+4	512	2,4
8	Dense	t	512	
9	Dense	t	512	4,6,8
10	RNN	t	512	
11	Dense	t	1500	
12	Polling	T	3000	
13	Dense	T	512	
14	Dense	T	512	
15	Dense	T	Num.Spks.	

2.2. Angular based speaker embedding leaning

Softmax cross entropy is the most widely used loss function for classification tasks. And much improvements has been made in face recognition task. Modified softmax uses $\|\mathbf{x}_i\|$ as weights and can it be written as:

$$L_{modified} = -\frac{1}{N}\sum_{i=1}\log\frac{e^{\|\mathbf{x}_i\|cos(\theta_{y_i,i})}}{\sum_{j=1}e^{\|\mathbf{x}_i\|cos(\theta_{j,i})}} \tag{1}$$

Based on this equation, different margins are introduced. And we choose additive margin softmax (AM-softmax)[22] as loss function to optimize the speaker classifier:

$$L_a = -\frac{1}{N}\sum_{i=1}\log\frac{e^{\|\mathbf{x}_i\|(cos(\theta_{y_i,i})-m)}}{e^{\|\mathbf{x}_i\|(cos(\theta_{y_i,i})-m)}+\sum_{j\neq y_i}e^{\|\mathbf{x}_i\|cos(\theta_{j,i})}} \tag{2}$$

In real implementation of speaker verification tasks, annealing technique is used in the training process:

$$f_{y_i} = \frac{\lambda\|\mathbf{x}_i\|cos(\theta_{y_i,i}) + \|\mathbf{x}_i\|cos(m\theta_{y_i,i})}{1+\lambda} \tag{3}$$

where f_{y_i} is the y_ith output logit given embedding $\mathbf{x}_i$, and λ is gradually reduced during training processing. And we use a more straight forward alternative in all the experiments:

$$L_{speaker} = (1-\lambda^{'})L_{modified} + \lambda^{'}L_a, \tag{4}$$

where we gradually increase $\lambda^{'}$ in first several epochs to gradually shift the loss from modified softmax to AM-softmax loss.

2.3. Attention mechanism

Traditionally, we make the time dimension pooling operation by computing the mean and standard deviation. To let the network focus on the speaker-specific features and ignore the influence of other elements, example for depth, channels or domains. We apply attention methods[6] instead of the original statistic pooling layer. The weight of each frame is defined as:

$$e_t = \boldsymbol{W_2}ReLU(\boldsymbol{W_1}\boldsymbol{h_t}) \tag{5}$$

$$\alpha_t = \frac{exp(e_t)}{\sum_{\tau}^{T}exp(e_\tau)} \tag{6}$$

where $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$ are two matrices, $\boldsymbol{h_t}$ is the frame level feature. After a softmax function, we can calculate the weight of each frame. Subsequently, we define the weighted mean $\widetilde{\boldsymbol{u}}$ and the weighted standard deviation $\widetilde{\boldsymbol{\sigma}}$ as follows[23]:

$$\widetilde{\boldsymbol{u}} = \sum_{t}^{T}\alpha_t\boldsymbol{h}_t \tag{7}$$

$$\widetilde{\boldsymbol{\sigma}} = \sqrt{\sum_{t}^{T}\alpha_t\boldsymbol{h}_t\cdot\boldsymbol{h}_t - \widetilde{\boldsymbol{u}}\cdot\widetilde{\boldsymbol{u}}} \tag{8}$$

After getting these attention statistics, concatenation of mean and standard deviation is feed to the next part as utterance level representations.

2.4. Domain adaptation methods

We explore two types of domain adaptation methods to transfer the system to perform better in the target domain.

2.4.1. One-class classification (OCC)

To reduce domain divergence, we can force the input from different dataset into a new nominal domain, which is called a one-class classification problem[24]. A fully connected (FC) layer and a sigmoid activate function are employed sequentially to the layer 13 in Table1. And the OCC loss can be written as:

$$L_{OCC} = -\frac{1}{N}\sum_{i=1}\log(p^0(x_i)) \tag{9}$$

where $p^0(x_i)$ is the probability that whether the feature x_i belongs to the new nominal domain.

2.4.2. Speaker-independent branch (SIDB)

After attention mechanism, we assume that the attention map builds a projection to the speaker specific space. So the rest part may contain other information such as domain[25], we call this part as Speaker-independent branch. And the speaker embedding and the speaker-independent feature can be represented as:

$$F_x^{em} = A_x\otimes F_x, F_x^{sp} = (1-A_x)\otimes F_x \tag{10}$$

where A_x is the attention map, $\otimes$ is the calculating operation of mean and standard deviation, and F_x is the frame-level features. The F_x^{sp} will then link to a domain classifier by a FC layer with cross entropy loss:

$$L_{DSB} = -\frac{1}{N}\sum_{i=1}\log(p(F_{x_i}^{sp})) \tag{11}$$

To ensure that the two features are totally mutual exclusive and independent, we add a soft orthogonal constraint loss written as follow:

$$L_{orth} = -\frac{1}{N}\sum_{i=1}\frac{F_{x_i}^{em}\cdot F_{x_i}^{sp}}{\|F_{x_i}^{em}\|_2^2\|F_{x_i}^{sp}\|_2^2} \quad (12)$$

As F_x^{sp} learns speaker-independent elements (domain label in this paper) better, the network can eliminate less useless information from F_x^{em}. So the F_x^{em} will only contains speaker-dependent features. Because all the training samples has speaker labels, we can jointly optimize the loss function as follows:

$$L_{total} = L_{speaker} + \gamma L_{DA} \quad (13)$$

where L_{DA} is domain transfer loss representing L_{OCC} or $L_{DSB} + L_{orth}$ and γ is a hyper-parameter.

2.5. Score normalization

We utilize Adaptive Symmetric Score Normalization (AS-Norm)[26, 27] to FFSVC results from all trials after cosine similarity scoring as a domain adaptation method. Through comparative experiments, we conclude that adaptive S-norm2 performs best in the indicator DCF_{min} and we choose the FFSVC development set as the adaptive cohort. We select X closest files (most positive scores) to the enrollment/test utterance as $\mathcal{E}_e^{top}$ or $\mathcal{E}_t^{top}$, and the cohort scores based on such selections are defined as:

$$\begin{aligned} S_e(\mathcal{E}_t^{top}) &= \{s(e,\varepsilon)|\forall\varepsilon\in\mathcal{E}_t^{top}\} \\ S_t(\mathcal{E}_e^{top}) &= \{s(t,\varepsilon)|\forall\varepsilon\in\mathcal{E}_e^{top}\} \end{aligned} \quad (14)$$

Then the adaptive S-norm2 (AS-Norm2) is:

$$\widetilde{s}(e,t) = \frac{1}{2}\left(\frac{s(e,t)-\mu[S_e(\mathcal{E}_t^{top})]}{\sigma[S_e(\mathcal{E}_t^{top})]} + \frac{s(e,t)-\mu[S_t(\mathcal{E}_e^{top})]}{\sigma[S_t(\mathcal{E}_e^{top})]}\right) \quad (15)$$

3. Experimental setup

3.1. Dataset preparing

We set up three data sets for training. FFSVC20 challenge training database includes 120 speakers and each speaker has 3 visits. The recordings from five recording devices for each utterance are provided for training, including one close-talk microphone, one 25cm distance cellphone, and three randomly selected microphone arrays (4 channels per array). For this set, although each speaker has a large number of utterances, the number of speakers is a small amount. Therefore, we introduce Vox-celeb1&2 set (SLR49), which covers 7323 speakers and the language is English. Furthermore, we add CN-celeb set (SLR82) with 1000 speakers to train them jointly. The language of FFSVC and CN-celeb is Chinese Mandarin. We combine FFSVC and Voxceleb as combineI set (7443 speakers), and combine FFSVC, Voxceleb and CN-celeb as combineII set (8443 speakers).

The given development set is split into two sub-sets. We use the official development trials to report the development results. And we use the rest as score normalization corpus. For each task, we randomly choose 400 utterances from score normalization corpus and set the top number to 200 for subsequent adaptive AS-norm2 operation.

3.2. Data augmentation

The data augmentation method we used is adding MUSAN[28] dataset and room impulse responses (RIRs) to the raw training data. And we randomly sampled 1500 augmentation utterances (2s-4s) for each speaker. Energy-based voice activate detection (VAD) is applied. After these operations, 30-dimensional Mel-frequency cepstral coefficient (MFCC) is extracted as the input acoustic feature with a frame-length of 25 ms, mean-normalized over a sliding window of 3 seconds.

3.3. Implementation Details

Stochastic Gradient Descent (SGD) with weight decay=5e-4 and momentum=0.9 is used as the optimization method. We use PyTorch platform to training the network, and other operations are implemented in Python. Each model is trained for 7 epochs with an initial learning rate of 0.01. The learning rate is gradually decreased to 0.0001. In the back-end modelling, we use the cosine similarity as a scoring method, which gives the cosine similarity of the two utterance embeddings. For the FFSVC, the primary measure metric is defined as minimun detection cost function (minDCF), which has same form as NIST SRE:

$$C_{det} = C_{miss}\times P_{miss}\times P_{tar} + C_{fa}\times P_{fa}\times(1-P_{tar}) \quad (16)$$

where the parameters C_{miss}, C_{fa} and P_{tar} are setting as 1.0, 1.0 and 0.01. Through equal error rate (EER) and C_{llr} is provided as auxiliary metrics, we select the sub-model which performs best in primary measure metric (minDCF).

4. Results and analysis

4.1. Comparative Experiment

To find the best backbone for the FFSVC task, we first conduct a series of experiments with different network architectures. As show in Table2. We can first observe that adding more layers to the original 5-layer x-vector truly make sense, the deeper framework has a stronger ability to extract frame-level features from the input acoustic features, and AS-Norm2 improves minDCF a lot as a special domain adaptation method. Although the speaker verification task does not consider the content of each utterance and the sequence will be pooling at the time dimension, adding an RNN layer can still improve the performance by 15% in EER and minDCF. Specially, we conducted a comparative experiment on LSTM and BiLSTM, and the latter shows a better performance in both FFSVC task and Vox task. Thus we choose the best structure as our backbone to carry out subsequent experiments.

Table 2: *System performance on FFSVC task2 development set and vox1 test set, training with combineI set*

Model	FFSVC Dev		Vox1 Test	
	minC	EER	minC	EER
x-vector(original)	0.885	9.48%	0.231	2.30%
x-vector(AS-Norm2)	0.809	9.41%	0.231	2.30%
ETDNN	0.741	8.05%	0.185	1.92%
LSTM-ETDNN	0.659	6.39%	**0.142**	1.61%
BiLSTM-ETDNN	**0.625**	**6.20%**	0.165	**1.48%**

4.2. System results on FFSVC development sets

Table3 illustrates the detailed results on different speaker verification tasks in FFSVC development sets. In general, systems

Table 3: *System Performance on FFSVC Development set, Att means attention mechanism, OCC means one class classification loss, DSB means domain specific branch and (II) means the system is trained by combine(II) set, AS-Norm2 is appiled to all the systems*

System		Task1		Task2		Task3		Vox1 Test	
ID	Model	minDCF	EER	minDCF	EER	minDCF	EER	minDCF	EER
1	x-vector	0.723	8.52%	0.809	9.41%	0.659	6.61%	0.321	2.30%
2	BiLSTM-ETDNN	0.651	6.29%	0.625	6.20%	0.570	4.99%	0.165	1.48%
3	BiLSTM-ETDNN+Att	**0.633**	6.01%	0.629	6.21%	**0.546**	4.68%	0.170	1.57%
4	BiLSTM-ETDNN (II)	0.648	**6.00%**	**0.612**	6.13%	0.559	4.77%	0.166	1.53%
5	BiLSTM-ETDNN+Att+OCC	0.666	6.33%	0.629	6.23%	0.548	**4.62%**	0.166	1.62%
6	BiLSTM-ETDNN+Att+SIDB	0.642	6.29%	0.661	**6.03%**	0.549	4.90%	**0.134**	**1.46%**
7	Submitted Fusion System (2∼4)	0.578	5.27%	0.555	5.21%	0.511	4.14%	0.136	**1.34%**
8	New Fusion System (2∼6)	**0.569**	**5.07%**	**0.554**	**5.06%**	**0.489**	**3.97%**	**0.119**	1.35%

Table 4: *Results of Submitted Fusion System (2∼4) on FFSVC Evaluation set (mid-term results)*

Task1		Task2		Task3	
minDCF	EER	minDCF	EER	minDCF	EER
0.695	7.25%	0.693	6.61%	0.625	7.00%

perform better in the text-independent task (FFSVC Task2) than text-dependent task (FFSVC Task1) with a single microphone array. It may because we directly transfer the text-independent trained system to the text-dependent task without any text constraint fine-tuning. Since task3 has more test utterances than task1 and task2, so all systems perform best in this task. It is interesting that by comparing system 2 and system 3, we can observe that replacing the statistic pooling by attention pooling improves the results in text-dependent tasks. We analyze the reason may be that the attention mechanism focuses more on speaker-specific features and it is less constrained by utterance content. System 4 has the same architecture as system 2, but training with 8443 speakers (combineII). As results, adding more data can slightly improve performance in all tasks.

Experiments of system 5 and system 6 are tested after the mid-term deadline, so the evaluation set results is not reported. These two experiments evaluate the performance of domain adaptation methods. The results demonstrate that OCC does not help the network in the FFSVC development tasks. We guess it is because that the OCC loss is arranged on speaker embedding. While narrowing the distance of domains, it also shortens the distance of speaker embeddings. Maybe the method makes sense in unsupervised situations, but it influences the speaker verification performances. And adding SIDB can is an efficient way in Vox1 test set, the improvement of minDCF is more than 30%. But it is a pity that the system does not perform as good after score normalization. According to the raw data, the SIDB has improved the minDCF and EER, but the improvement by AS-Norm is not as obvious as other systems. Probably adding more FFSVC training data can make SIDB work better.

In order to win a better ranking, we fuse the system 2, 3 and 4 with the weight of their performance in the development tasks. The fusion system is system 6 and it is obviously that the fusion operation can learn the advantages of each single system. Finally, we fuse all the above ETDNN based systems (2∼6) as system 8. The new fusion system outperform the submitted fusion system in the three tasks. Figure 1 shows the DET plots

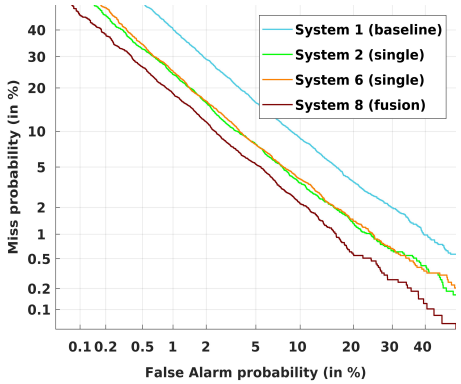

Figure 1: *DET plots for FFSVC Task2 development set with 4 systems*

with 4 systems, which can visually reflect our improvements.

4.3. Evaluation set results of submitted fusion systems

We submit the fusion system of 2,3,4 before the mid-term deadline and the result of the evaluation set is listed in Table 4. Although we can observe a big gap between development results and evaluation results, the performance progress trend on the development set is still consistent with evaluation set: our system perform best in task2, but also achieve good results in the other two tasks. The difference may be caused by the choice of score normalization cohort sets and text-dependent fine-tuning.

5. Conclusions

In this work, we propose a strengthened speaker verification system for FFSVC 2020. Followed by data augmentation, the network framework consists of TDNN, RNN and an attention layer, which shows a powerful ability in embedding extracting. In the training stage, we jointly train the model with AM-softmax loss and domain adaptation loss. In the backend scoring, we implement cosine similarity and score normalization. Our submitted fusion system achieves good results and we believe our new system can get a better rank before the final deadline by adding some domain adaptation methods and more data. However, evaluation set results are much worse than the development set and the FFSVC results are out of line with Voxceleb, which leaves a great challenge to explore.

6. References

[1] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[2] D. Cai, X. Qin, W. Cai, and M. Li, "The DKU System for the Speaker Recognition Task of the 2019 VOiCES from a Distance Challenge," in *Proc. Interspeech 2019*, 2019, pp. 2493–2497. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1435

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

[4] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[5] W. Cai, Z. Cai, X. Zhang, X. Wang, and M. Li, "A novel learnable dictionary encoding layer for end-to-end language identification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5189–5193.

[6] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Proc. Interspeech 2018*, 2018, pp. 3573–3577. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1158

[7] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5791–5795.

[8] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," in *Proc. Interspeech 2018*, 2018, pp. 3623–3627. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1545

[9] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879–4883.

[10] S. Novoselov, V. Shchemelinin, A. Shulipa, A. Kozlov, and I. Kremnev, "Triplet loss based cosine similarity metric learning for text-independent speaker recognition," in *Proc. Interspeech 2018*, 2018, pp. 2242–2246. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1209

[11] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[12] M. K. Nandwana, J. van Hout, C. Richey, M. McLaren, M. A. Barrios, and A. Lawson, "The VOiCES from a Distance Challenge 2019," in *Proc. Interspeech 2019*, 2019, pp. 2438–2442. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1837

[13] X. Qin, M. Li, H. Bu, R. K. Das, W. Rao, S. Narayanan, and H. Li, "The ffsvc 2020 evaluation plan."

[14] Z. Meng, Y. Zhao, J. Li, and Y. Gong, "Adversarial speaker verification," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6216–6220.

[15] G. Bhattacharya, J. Monteiro, J. Alam, and P. Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6226–6230.

[16] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-950

[17] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1929

[18] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," 2019.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[21] Z. Ren, G. Yang, and S. Xu, "Two-Stage Training for Chinese Dialect Recognition," in *Proc. Interspeech 2019*, 2019, pp. 4050–4054. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1522

[22] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[23] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech 2018*, 2018, pp. 2252–2256. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-993

[24] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[25] Y. Huang, P. Peng, Y. Jin, J. Xing, C. Lang, and S. Feng, "Domain adaptive attention model for unsupervised cross-domain person re-identification," *arXiv preprint arXiv:1905.10529*, 2019.

[26] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42–54, 2000.

[27] P. Matějka, O. Novotný, O. Plchot, L. Burget, M. D. Sánchez, and J. Černocký, "Analysis of score normalization in multilingual speaker recognition," in *Proc. Interspeech 2017*, 2017, pp. 1567–1571. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-803

[28] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.