INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

THUEE System for NIST SRE19 CTS Challenge

Ruyun Li[1], Tianyu Liang[2*], Dandan Song[3], Yi Liu[1], Yangcheng Wu[1], Can Xu[2], Peng Ouyang[3], Xianwei Zhang[2], Xianhong Chen[2], Weiqiang Zhang[2], Shouyi Yin[1†] and Liang He[2†]*

[1]Institute of Microelectronics, Tsinghua University, Beijing, China
[2]Department of Electronic Engineering, Tsinghua University, Beijing, China
[3]TsingMicro Co. Ltd. Beijing, China

liry17@mails.tsinghua.edu.cn, ty_liang@mail.tsinghua.edu.cn, yinsy@tsinghua.edu.cn, heliang@mail.tsinghua.edu.cn

Abstract

In this paper, we present the system that THUEE submitted to NIST 2019 Speaker Recognition Evaluation CTS Challenge (SRE19). Similar to the previous SREs, domain mismatches, such as cross-lingual and cross-channel between the training sets and evaluation sets, remain the major challenges in this evaluation. To improve the robustness of our systems, we develop deeper and wider x-vector architectures. Besides, we use novel speaker discriminative embedding systems, hybrid multi-task learning architectures combined with phonetic information. To deal with domain mismatches, we follow a heuristic search scheme to select the best back-end strategy based on limited development corpus. An extended and factorized TDNN achieves the best single-system results on SRE18 DEV and SRE19 EVAL sets. The final system is a fusion of six subsystems, which yields EER 2.81% and minimum cost 0.262 on the SRE19 EVAL set.

Index Terms: speaker recognition, speaker embedding, deep neural network, NIST SRE, multi-task learning

1. Introduction

Speaker recognition is to determine whether a specified target speaker is speaking or not in a given segment of speech. Organized by the National Institute of Standards and Technology (NIST), the series of Speech Recognition Evaluations (SRE) usually reflect the most advanced technology in the field of speaker recognition. Similar to SRE18, SRE19 evaluation sets consist of two separate parts: (i) narrowband Tunisian Arabic conversational telephone speech (CTS) collected outside North America and (ii) wideband English recordings drawn from the Video Annotation for Speech Technology corpus (VAST). Due to different collection conditions and languages between training data and evaluation data, domain mismatches remain the major challenges. Unconstrained public and proprietary data can be used to narrow the performance degradation caused by domain mismatches. Besides, a leaderboard on the part of the evaluation dataset is offered to motivate the development of various systems [1].

In this paper, we present our submissions to the CTS challenge. According to the results of previous SREs, deep speaker embedding, such as x-vector [2], has already replaced i-vector [3] and becomes the state-of-the-art method in the field of text-independent speaker verification. Our speaker recognition systems are based on x-vector. To improve the robustness of the x-vector extractor, we increase the amount and diversity of training data through data augmentation. Powered by advanced computational resources, and large-scale speech datasets, we also train sufficiently deep speaker models to fit very complicated distributions in the speaker feature space, such as ETDNN [4] and FTDNN [5, 6]. An extended factorized TDNN (EFTDNN), trained on 5-fold data augmentation, is our best single system with a much deeper and wider architecture. Besides, we use novel speaker discriminative embedding systems, hybrid multi-task learning architectures combined with automatic speech recognition (ASR), including hybrid multi-task [7] and c-vector [8]. They take advantage of phoneme features and context information.

Given the limited development data, we follow a heuristic search scheme to mitigate the mismatch between training sets and evaluation sets. Following the extraction of x-vector, Gaussian probabilistic linear discriminant analysis (GPLDA)[9] scoring, domain adaptation, and adaptive symmetric score normalization (AS-Norm)[10] are adopted. We also explore the local pairwise linear discriminant analysis (LP-LDA)[11] / multiobjective optimization training of probabilistic linear discriminant analysis (Mot-PLDA)[12] back-end. All the subsystems are fused and calibrated using the BOSARIS toolkit [13], which learns a scale and a bias for each system by logistic regression approach.

This paper is organized as follows: Sec.2 discusses related literature in SRE. Sec.3 and Sec.4 describes the dataset and architectures for individual subsystems. Sec.5 introduces the back-end solutions for SRE19. Sec.6 reports experimental results obtained with each single system and the fusion systems on the SRE18 DEV and SRE19 EVAL sets. Finally, we conclude our work with a future direction in Sec.7.

2. Relation to prior work

Deep speaker embedding learning has been dominating the field of text-independent speaker verification. Very deep x-vector architectures - Extended TDNN [4] and Factorized TDNN [5], and ResNets [6] - obviously outperform shallower x-vectors. Our explorations are based on those x-vector systems. To improve the robustness of our systems, we construct deeper and wider x-vectors. Besides, we use angular margin loss (AM-softmax) as the training criterion for ETDNN, which was proposed for face recognition [14] and introduced to speaker verification in [15]. AM-softmax has more stringent requirements for correct classification when $m \geq 2$ (an integer that controls the angular margin), which generates an angular classification margin between embeddings of different classes. To increase the diversity of our subsystems, we employ hybrid multi-task learn-

*equal contribution
†corresponding authors

http://dx.doi.org/10.21437/Interspeech.2020-1245

Table 1: *Corpora used in individual system*

Model	Training set	Data Augmentation	LDA/PLDA	PLDA-adapt	AS-Norm
EFTDNN	SRE + SWB II&Cell + Voxceleb + CH&CF	5-folds	SRE + SRE18Eval	SRE + SRE18Eval	SRE18_unlabel
Multi-task & C-vector	GMM-HMM & ASR: Fisher&SWB I	-	SRE + SRE18Eval	MIXER6 + SRE16 + SRE18Eval + SRE18_unlabel + CH&CF	SRE18_unlabel
	x-vector: SRE + SWB II&Cell + Voxceleb	2-folds			
ETDNN	SRE + SWB II&Cell + Voxceleb	3-folds	SRE + SRE18Eval	SRE + SRE18Eval	SRE18_unlabel
FTDNN & ResNet	SRE + SWB II&Cell + Voxceleb	2-folds	SRE + SRE18Eval	SRE + SRE18Eval	SRE18_unlabel

Table 2: *Datasets Notations*

Dataset	Corpora	#Spk
SRE	SRE04-10 + SRE16 + MIXER6	5209
SWB II&Cell	Switchboard II phase 2 & 3 + Switchboard Cellular Part 1 & 2	2594
Voxceleb	Voxceleb 1 & 2	7363
CH&CF	Callhome + Callfriend	3238
Fisher&SWB I	Fisher + Switchboard I	3800

ing architectures combined with phonetic information. [16] proposed collaborative joint learning of automatic speech and speaker recognition based on multi-task recurrent neural network models. In this paper, we use Multi-task [7] and C-vector architectures [8] which combines phonetic information with x-vector [2]. Besides, back-end strategies are optimized to deal with domain mismatches [17, 18].

3. Dataset

For submissions to NIST SRE19, there are extensive training sets available for system development under the open condition. According to our experimental selection, Table 1 summarizes the data configuration corresponding to different frameworks. In different systems, training sets (i.e., SRE, SWBD, Voxceleb) are post-augmented by different folds by convolving with far-field Room Impulse Responses (RIRs) or adding noise from the MUSAN corpus [19]. The speaker filtering criterion is applied to training sets, which excludes recordings with less than 400 frames and speakers with fewer than 8 recordings. Table 2 defines the dataset notations and lists the number of filtered speakers for each dataset. Kaldi x-vector recipe [1] is used for this processing.

4. Speaker embedding systems

In order to utilize the complementarity of various systems, six general speaker recognition frameworks are explored for the submission of SRE19, namely ETDNN, FTDNN, EFTDNN, ResNet, Multi-task and C-vector. Their detailed configurations are described in this section.

4.1. ETDNN/AMS system

ETDNN/AMS system is an extended version of TDNN based on AM-softmax loss. Compared with the ETDNN in [20], our ETDNN has a wider context and interleaving dense layers between each two TDNN layers. The architecture is the same as the ETDNN architecture in [6], except that the context of layer 5 of our system is t-3:t+3 instead of t-3, t, t+3. The x-vector is extracted from layer 12 before the ReLU non-linearity. Besides, AM-softmax with $m = 0.15$ is used instead of the traditional softmax loss. It is easier to train and generally performs better than angular softmax, which will be proved in Section 6.

[1] https://github.com/kaldi-asr/kaldi/blob/master/egs/sre16/v2

Table 3: *Factorized TDNN x-vector architecture*

Layer	Layer type	Context factor1	Context factor2	Skip conn. from layer	Size	Inner size
1	TDNN	t-2:t+2			512	
2	FTDNN	t-2,t	t, t+2		1024	256
3	FTDNN	t	t		1024	256
4	FTDNN	t-3, t	t, t+3		1024	256
5	FTDNN	t	t	3	1024	256
6	FTDNN	t-3, t	t, t+3		1024	256
7	FTDNN	t-3, t	t, t+3	2,4	1024	256
8	FTDNN	t-3, t	t, t+3		1024	256
9	FTDNN	t-3, t	t, t+3	4,6,8	1024	256
10	Dense-ReLU	t	t		2048	
11	Pooling(mean+stddev)	full-seq			4096	
12	Dense-ReLU				1024	
13	Dense-ReLU				1024	
14	Dense-Softmax				N. spks.	

4.2. FTDNN system

Factorized TDNN (FTDNN) has achieved great success in SRE18 [6]. By factorizing the weight matrix of each TDNN layer into the product of two low-rank matrices, the number of network parameters can be reduced. The first of these factors is constrained to be semi-orthogonal. It is assumed that the semi-orthogonal constrain will help to retain the main information while projecting from a higher dimension to the low-rank dimension. The factorization and skip-connection in FTDNN can effectively solve the gradient vanishing problem. In this paper, we follow the FTDNN described in [6] and make two subtle differences. Firstly, our FTDNN system is trained on AM-softmax. Another modification is a higher speaker embedding dimension, which is considered to preserve more speaker information. We use 1024 nodes instead of 512 nodes in layer 12 and 13. The 1024-dimensional x-vector is extracted from layer 12 before the ReLU non-linearity.

4.3. Extended FTDNN/AMS system

Inspired by the success of ETDNN and FTDNN, we explore a deeper and wider version of the FTDNN [5], called extended factorized TDNN (EFTDNN). The architecture is summarized in Table 4. Dense layers are interleaved between FTDNN layers. More hidden layers and wider temporal representation enable the network to model complicated frame-level representation of utterances. Besides, "factorizing the convolution" and "3-stage splicing" are utilized instead of basic factorizing, which factorizes the TDNN layer into a feed-forward layer multiplied by a convolution. In the EFTDNN, we have a factorized TDNN layer with a constrained 2×1 convolution to dimension 256, followed by another constrained 2×1 convolution to dimension 256, followed by a 2×1 convolution back to the hidden-layer dimension (e.g., 1024). The dimension now goes from, $1024 \rightarrow 256 \rightarrow 256 \rightarrow 1024$, within one layer. The effective temporal context of this setup is, of course, wider than the original TDNN layer, due to the extra 2×1 convolution. The x-vector is extracted from the layer 22 prior to the Relu

Table 4: *Extended factorized TDNN x-vector architecture*

Layer	Layer Type	Context factor1	Context factor2	Context factor3	Skip conn, from layer	Size	Inner size
1	TDNN-ReLU	t-2 : t+2				512	
2	Dense-ReLU	t				512	
3	FTDNN-ReLU	t-3, t-1	t-1, t+1,	t+1, t+3		1024	256
4	Dense-ReLU	t				1024	
5	FTDNN-ReLU	t	t	t		1024	256
6	Dense-ReLU	t				1024	
7	FTDNN-ReLU	t-5,t-2	t-2, t+1	t+1, t+4		1024	256
8	Dense-ReLU	t				1024	
9	FTDNN-ReLU	t	t	t	5	1024	256
10	Dense-ReLU	t				1024	
11	FTDNN-ReLU	t-5, t-2	t-2, t+1	t+1, t+4		1024	256
12	Dense-ReLU	t				1024	
13	FTDNN-ReLU	t-5,t-2	t-2, t+1	t+1, t+4	3, 7	1024	256
14	Dense-ReLU	t				1024	
15	FTDNN-ReLU	t-5, t-2	t-2, t+1	t+1, t+4		1024	256
16	Dense-ReLU	t				1024	
17	FTDNN-ReLU	t	t	t	7, 11, 15	1024	256
18	Dense-ReLU	t				2048	
19	Dense-ReLU	t				2048	
20	Dense-ReLU	t				2048	
21	Pooling (mean+stddev)	full-seq				2×2048	
22	Dense-ReLU					1024	
23	Dense-ReLU					1024	
24	Dense-Softmax					N spks.	

non-linearity.

4.4. ResNet system

ResNet architecture is also inspired by x-vector [20] where the TDNN layers are replaced by a residual network with 2D convolutions (ResNet34) [21]. The network architecture contains: a front-end ResNet34, a statistic pooling layer, and a feed-forward network. Specifically, we adopt the angular softmax loss (A-softmax) [22, 23], which learns angularly discriminative features by generating an angular classification margin between embeddings of different classes.

4.5. Multi-task system

In text-independent speaker verification systems such as x-vector, the effect of phonetic information on the speech signal has been mostly ignored. In fact, the characteristics of a speaker are different on different phonemes. If these characteristics could be further fine-grained on phonemes, the speaker could be better represented. In this paper, we implement a hybrid multi-task learning system, which combines the x-vector network with an ASR network [7]. This ASR network share TDNN layers with the x-vector, while retaining frame-level information. The multi-task system can extract more delicate speaker features, specifically phonetically-aware speaker representations. During the training process, phonetic and speaker examples are merged into different mini-batches. Finally, the speaker embedding is extracted from the hidden layer behind the pooling layer of the x-vector network.

4.6. C-vector system

Although phonetic information is helpful for speaker recognition, there is a trade-off between the frame-level phoneme discrimination and the segment-level speaker discrimination. The c-vector network [8] combines the hybrid multi-task learning and phonetic adaptation to learn the shared parts of phonetic information and speaker information. The hybrid multi-task learning tries to learn more effective information from phonetic content, while phonetic adaptation tries to suppress the negative effect of phonetic content. The architecture of the c-vector network is shown in Figure 1.

First, an ASR model with a bottleneck layer is pre-trained. Then the output of the bottleneck layer is merged into the x-vector network of hybrid multi-task learning as an auxiliary vector. During the training of a hybrid multi-task learning network, the pre-trained ASR network is no longer updated. The processes of training and embedding extraction are the same as the aforementioned Multi-task system in Section 4.5. A GMM-HMM is also required to do phonetic alignment for training datasets as Section 4.5.

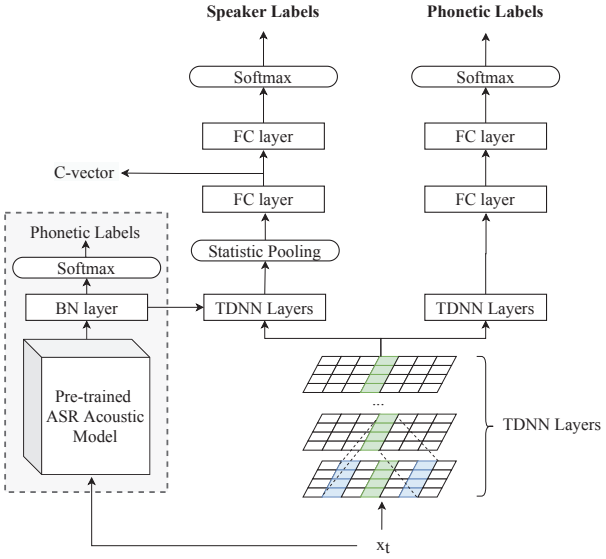

Figure 1: *C-vector architecture*

Table 5: *Comparison between LDA and LP-LDA on the SRE18 DEV dataset*

System	LDA / LP-LDA	AS-Norm	SRE18 DEV	
			EER(%)	min-DCF
EF-TDNN	LDA	×	4.22	0.330
	LP-LDA		4.86	0.291
	LDA	√	3.67	0.196
	LP-LDA		4.29	0.255

5. Back-end scoring strategies

The same length-normalization and centralization/mean subtraction strategy is applied to each subsystem. LDA is applied to reduce the dimension of the embedding, and then Gaussian PLDA with a full co-variance residual noise term is trained on the speaker discriminant features. Then domain adaptation is implemented with interpolation of out-of-domain PLDA and in-domain PLDA [24]. Unsupervised speaker clustering is performed on the SRE18 unlabeled set to get the labels for in-domain PLDA training. After that, the scores of any given enrollment-test pair are calculated as the log-likelihood ratio on the PLDA model. Also, adaptive symmetric score normalization (AS-Norm) is used when generating the PLDA score, with an adaptive cohort selection scheme followed by top score selection.

To deal with domain mismatches, we follow a heuristic search scheme among back-end strategies. We explore the LP-LDA [11] / MotPLDA [12] back-end. According to Table 5, when AS-Norm is not used, LP-LDA achieves a better minDCF than LDA. When using AS-Norm, the EER and DCF of LP-

Table 6: *System performance on SRE18 DEV and SRE19 EVAL sets. AS-Norm is used in all subsystems.*

System	Whiten	SRE18 DEV		SRE19 EVAL	
		EER(%)	min-DCF	EER(%)	min-DCF
ETDNN	×	3.95	0.222	3.37	0.296
FTDNN	√	4.28	0.258	3.58	0.331
EFTDNN	√	**3.67**	**0.196**	**3.16**	**0.290**
ResNet34	×	4.02	0.253	3.38	0.306
Multi-task	√	4.35	0.276	4.00	0.346
C-vector	×	3.92	0.252	3.94	0.340
Final fusion: six subsystems	-	3.45	0.176	2.81	0.262
Fusion except EFTDNN	-	3.52	0.191	2.94	0.273
Fusion except ETDNN	-	3.39	0.183	2.88	0.271
Fusion except FTDNN	-	3.46	0.173	2.82	0.260
Fusion except Multi-task	-	3.46	0.174	2.78	0.262
Fusion except C-vector	-	3.49	0.178	2.79	0.262
Fusion except ResNet34	-	3.49	0.172	2.85	0.266
Fusion: ETDNN + EFTDNN	-	3.64	0.176	2.86	0.265
Fusion: ETDNN + EFTDNN + ResNet34	-	3.48	0.175	2.79	0.260

LDA and LDA are significantly improved, but LDA turns out to be superior. This may be due to the conflict between LP-LDA and AS-Norm in the data selection process. Furthermore, when we replace PLDA with MotPLDA [12], MotPLDA shows a lower DCF. However, further investigation is required on how to perform MotPLDA adaptation. Although LDA/PLDA itself performs worse than LP-LDA/MotPLDA, their combination with adaptation and AS-Norm makes a significant improvement. The LP-LDA/MotPLDA back-end deserves a future investigation.

6. Results and analysis

In this section, we report the results of the individual systems and their fusions over the entire SRE18 DEV and SRE19 EVAL sets. By comparing the performance of different fusion systems, we study the complementarity between individual systems. Besides, the DET curves of individual systems and the final fusion system are shown in Figure 2.

As shown in the first block of Table 6, EFTDNN and ETDNN are significantly superior to other systems on both SRE18 DEV and SRE19 EVAL. FTDNN, ETDNN, and EFTDNN are all TDNN x-vector systems. By comparing their performance, we find that more training data and deeper models are helpful. This demonstrates that x-vector is data-driven. In terms of x-vector, the interleaving of dense layers in between the convolutional layers significantly improves the performance. Besides, according to Figure 3, AM-softmax is proven to be more stable and faster than A-softmax on large-scale x-vector architectures (i.e, EFTDNN). This is because AM-softmax has a more flexible angular margin at the beginning of network training.

In the first block of Table 6, another obvious observation is that hybrid multi-task learning systems appear to be inferior to other individual systems. However, it is noticeable that the ASR networks of Multi-task and C-vector are trained using Switchboard and Fisher sets, which are spoken in English, while SRE18 and SRE19 sets are spoken in Tunisian Arabic. These two languages have completely different phonemes. In such a situation where languages are extremely mismatched between training and evaluation sets, these two hybrid multi-task learning systems still perform well enough.

As shown in the second block of Table 6, in general, the contribution of different individual systems to the fusion system is positively related to its performance. On SRE18 DEV, the fusion of the ETDNN, EFTDNN, and ResNet improved over the ResNet alone, by 13.4% in EER, and the fusion of all six systems was slightly better. Although FTDNN performs well itself, its role in the fusion system is not so important because of the strong correlation between TDNN x-vector systems. As a result of the observed SRE18 DEV performance, our primary system for NIST SRE19 CTS Challenge is the linear fusion of all the above six subsystems by BOSARIS Toolkit [25]. Our final submission obtains actDCF of 0.230 and 0.287 on the SRE19 EVAL Progress and Test set, respectively. It is evaluated by the primary metric provided by NIST SRE 2019.

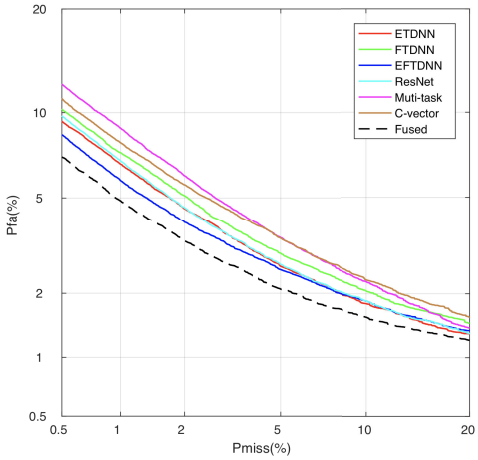

Figure 2: *DET curves for different systems on SRE19 EVAL*

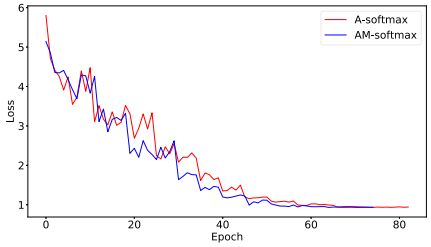

Figure 3: *AM-softmax vs A-softmax on EFTDNN*

7. Conclusion

In this paper, we present the components and analyze the results of the THUEE systems for the NIST 2019 SRE CTS Challenge. Different acoustic features, various front-end modeling methods, and heuristics searching among various backends are implemented. The fusion shows that different subsystems are complementary to each other at the score level. If there are sufficient memory and computation resources, deeper speaker models trained on larger training sets are helpful, and AM-softmax is proven to be more stable and faster than A-softmax on large-scale x-vector architectures. Besides, speaker embedding systems combined with phonetic information achieves competitive performance on speaker recognition. To mitigate the performance degradation caused by domain mismatches, adaptation and score normalization strategies for the LP-LDA / MotPLDA back-end deserves further investigation.

8. References

[1] "Nist 2019 speaker recognition evaluation," https://www.nist.gov/system/files/documents/2019/07/22/2019_nist_speaker_recognition_challenge_v8.pdf, 2019.

[2] 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)*. IEEE, 2018, pp. 5329–5333.

[3] 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.

[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)*. IEEE, 2019, pp. 5796–5800.

[5] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks." in *Interspeech*, 2018, pp. 3743–3747.

[6] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, and etc, "The jhu-mit system description for nist sre18," in *NIST Speaker Recognition Evaluation Workshop*, 2018.

[7] Y. Liu, L. He, J. Liu, and M. T. Johnson, "Speaker embedding extraction with phonetic information," in *INTERSPEECH*, 2018, pp. 2247–2251.

[8] ——, "Introducing phonetic information to speaker embedding for speaker verification," *EURASIP Journal on Audio, Speech, and Music Processing*, accept.

[9] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[10] P. Matejka, O. Novotnỳ, O. Plchot, L. Burget, M. D. Sánchez, and J. Cernockỳ, "Analysis of score normalization in multilingual speaker recognition." in *INTERSPEECH*, 2017, pp. 1567–1571.

[11] L. He, X. Chen, C. Xu, J. Liu, and M. T. Johnson, "Local pairwise linear discriminant analysis for speaker verification," *IEEE Signal Processing Letters*, vol. 25, no. 10, pp. 1575–1579, 2018.

[12] L. He, X. Chen, C. Xu, and J. Liu, "Multi-objective optimization training of plda for speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6026–6030.

[13] N. Brümmer and E. D. Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *Eprint Arxiv*, 2013.

[14] 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, July 2018.

[15] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," in *INTERSPEECH*, 2019, pp. 2873–2877.

[16] Z. Tang, L. Li, D. Wang, R. Vipperla, Z. Tang, L. Li, D. Wang, and R. Vipperla, "Collaborative joint training with multitask recurrent model for speech and speaker recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 25, no. 3, pp. 493–504, 2017.

[17] K. A. Lee, V. Hautamaki, T. Kinnunen, H. Yamamoto, K. Okabe, V. Vestman, J. Huang, G. Ding, H. Sun, A. Larcher *et al.*, "I4u submission to nist sre 2018: Leveraging from a decade of shared experiences," *arXiv preprint arXiv:1904.07386*, 2019.

[18] S. Novoselov, A. Gusev, A. Ivanov, T. Pekhovsky, A. Shulipa, G. Lavrentyeva, V. Volokhov, and A. Kozlov, "Stc speaker recognition systems for the voices from a distance challenge," *arXiv preprint arXiv:1904.06093*, 2019.

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

[20] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *INTERSPEECH*, 2017, pp. 999–1003.

[21] 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.

[22] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

[23] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *arXiv preprint arXiv:1804.05160*, 2018.

[24] D. Garcia-Romero and A. McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4047–4051.

[25] N. Brümmer and E. de Villiers, "The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF," *arXiv e-prints*, Apr. 2013.