TJU System Description to Short-duration Speaker Verification (SdSV) Challenge 2020

Ruiteng Zhang[1], *Jianguo Wei*[2]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Computer College, Qinghai Nationalities University, Xining, China

email@address

Abstract

This paper describes the submission of Tianjin University team (Team48) to the Short-duration Speaker Verification (SdSV) Challenge 2020. The challenge is focused on the problem of performance degradation of short-duration speaker verification. In this work, we investigate different deep neural networks architectures with multiple large marge softmax losses to solve the task. The primary system achieved 2.69% ERR and 0.1122 minDCF08 on the Task 2 evaluation set respectively.

Index Terms: speaker verification, SdSV, x-vector

1. Introduction

This document mainly describes the Tianjin University (TJU) team submissions for Task 2 of Short-duration Speaker Verification (SdSV) Challenge 2020 [1]. The SdSV Challenge 2020 concentrates on the speaker verification (SV) task of short-duration, and it expects that deep neural networks will play a key role. The challenge has two separate tasks: text-dependent (TD) and text-dependent (TI) speaker verification in a short duration scenario.

To increase the capacity of networks well capture short-duration information of speakers, our submitted systems are all based on Deep Neural Networks (DNNs). In this paper, we have three contributions to increase the performance of TI-SV on a short-duration scenario. First, we adopted the speaker augmentation to resolve the problem of low-resource on the challenge dataset. Then, a novel architecture of x-vector-based transfer-learning models is proposed to enhance the discriminations of speaker-verification networks. Finally, a series of large margin softmax losses are used to optimizer our models.

The rest of this document is organized as follows: Section 2 describes the setup for our systems. A description of our end-to-end-based models is given in Section 3. The results and analysis are presented in the last section.

2. Experimental Setup

2.1. Training data

For TI-SV, we used DeepMine [2] (Task 2 Train Partition), the train-clean-100 part and train-clean-360 part of LibriSpeech [3], VoxCeleb1 [4] dev part, and VoxCeleb2 [5] dev part to train models. The VoxCeleb1 and VoxCeleb2 dev part total have 7205 speakers and more than 1.2 million speeches. The train-clean-100 part and train-clean-360 part of LibriSpeech have 251 and 921 speakers, respectively. DeepMine (Task 2 Train Partition) has 588 speakers and 85,764 utterances. The configuration of these databases are summarized in Table 1.

Table 1: *The configuration of databases.*

Data Set	# Speakers	# Utterances
DeepMine (Task 2 Train Partition)	588	85,764
VoxCeleb1 dev	1,211	148,642
VoxCeleb2 dev	5,994	1,092,009
Libirispeech (train-clean-100)	251	28,539
Libirispeech (train-clean-360)	921	104,014

2.2. Training sample selection

VoxCeleb2 is extracted from videos uploaded to YouTube with real-world scenarios. Although the real-world scene can help neural networks obtain more robust capacity, an excessively noisy environment affects the transfer learning model to learn information on the target databases. To make our transform learning models more robust on DeepMine, we built a TDNN with AMM-Softmax to find an excessively hard sample and remove them.

2.3. Augmentations

The number of training speakers is a significantly important factor for the good performance of speaker-verification networks. We adopted speaker augmentation [6] to make additional target speakers to train the end-to-end systems, helps them obtain an accurate speaker-discriminative feature representation. For example, the VoxCeleb1 dev part has 1,211 speakers, speaker augmentation with speed factor is 0.9 and 1.1 produces 2,422 additional target speakers. In this work, we investigated the effect of different speed factors (0.9 to 1.1 and 0.8 to 1.2.) on the performance of speaker verification systems.

We built two huge datasets for pre-training speaker verification: The first dataset named "Vox1_Simple-Vox2_DeepMine_aug2-all", which includes three sub-datasets with speaker augmentation (speed factors: 0.9 to 1.1), the VoxCeleb1 dev part, the simple VoxCeleb2 dev part, and DeepMine (Task 2 Train Partition). Its total has 15,440 speakers. The second dataset called "Vox1_Vox2_DeepMine-aug2", which has the VoxCeleb1 dev part, the VoxCeleb2 dev part, and the DeepMine (Task 2 Train Partition) with speaker augmentation (speed factors: 0.9 to 1.1). There are 8,969 speakers in the second dataset.

We also constructed two in-domain datasets for training our in-domain model named "Vox1_DeepMine-aug4" and "DeepMine-aug4". The "Vox1_DeepMine-aug4" includes DeepMine (Task 2 Train Partition) with speaker augmentation (speed factors: 0.8 to 1.2) and the VoxCeleb1 dev part while the "DeepMine-aug4" include DeepMine (Task 2 Train Partition) with speaker augmentation (speed factors: 0.8 to 1.2).

Table 2: *The architecture of E-TDNN*

Layer	Layer Type	Context	Size
1	Frame1	t-2:t+2	512
2	Frame2	t	512
3	Frame3	t-1:t+1	512
4	Frame4	t	512
5	Frame5	t-1:t+1	512
6	Frame6	t	512
7	Frame7	t-2:t+2	512
8	Frame8	t	512
9	Frame9	t	512
10	Frame10	t	1536
11	Statistics Pooling	Full-seq	3072
12	Segment1	-	512
13	Segment2	-	512
14	AAM-Softmax	-	#speakers

Table 3: *Comparison of performance of our primary system and the x-vector baseline on the progress dataset of the challenge.*

Evaluation Set	EER (%)	minDCF08	EER (%)	minDCF08
progress	2.68	0.1122	10.67	0.4319
progress-male	2.20	0.1007	8.31	0.3216
progress-female	2.98	0.1191	11.73	0.4419
progress-EN	3.03	0.1388	9.61	0.4111
progress-FA	2.17	0.0805	6.17	0.2949
progress-EN-male	2.59	0.1200	7.04	0.2759
progress-EN-female	3.29	0.1492	10.55	0.4387
progress-FA-male	1.66	0.0703	4.24	0.2053
progress-FA-female	2.50	0.0868	6.59	0.2950
progress-Farsi-TC-vs-IC-subset	4.28	0.1448	3.91	0.1744

Table 4: *Comparison of performance of our primary system and the x-vector baseline on the evaluation dataset of the challenge.*

Evaluation Set	EER (%)	minDCF08	EER (%)	minDCF08
evaluation	2.69	0.1118	10.67	0.4324
evaluation-male	2.22	0.1001	8.26	0.3239
evaluation-female	2.97	0.1188	11.71	0.4430
evaluation-EN	3.03	0.1380	9.58	0.4118
evaluation-FA	2.19	0.0808	6.14	0.2962
evaluation-EN-male	2.57	0.1188	7.11	0.2770
evaluation-EN-female	3.30	0.1490	10.58	0.4374
evaluation-FA-male	1.70	0.0719	4.25	0.2062
evaluation-FA-female	2.51	0.0863	6.53	0.2953
evaluation-Farsi-TC-vs-IC-subset	4.30	0.1461	3.91	0.1725

2.4. Feature

In the training and testing stage, we adopted 161-dimensional spectrograms as the speech feature to fed into speaker-verification networks.

2.5. Loss Functions

AM-Softmax. The additive margin (AM-Softmax) [7] loss incorporates an addictive cosine margin to Softmax loss. The formulation of AM-Softmax loss is as follow.

$$L_{AAM} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(cos\theta_{y,i}-m)}}{e^{s(cos\theta_{y,i}-m)}+\sum\limits_{j\neq y_i}^{N}e^{s(cos\theta_{j,i})}}, \quad (1)$$

where N is the number of training speakers, m is the additive cosine margin, and s is a scaling factor.

AM-Softmax. The additive angular margin (AAM-Softmax) [8] loss introduces an additive angular margin in Softmax loss. The equation of AAM-Softmax loss is:

$$L_{AAM} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(cos(\theta_{y,i}+m))}}{e^{s(cos(\theta_{y,i}+m))}+\sum\limits_{j\neq y_i}^{N}e^{s(cos\theta_{j,i})}}, \quad (2)$$

where N is the number of training speakers, m is the additive angular margin, and s is a scaling factor.

2.6. Back-end Functions

We adopted Cosine similarity, and PLDA [9] as the back-end function to calculate the similarity scores for speaker embeddings.

3. End-to-end-based Speaker Verification Systems

3.1. X-vectors

In this work, x-vector systems were extracted embedding by extended time-delay neural networks (E-TDNN). Our E-TDNN included ten TDNN layer to extract frame-level features. Then pooling layer aggregates frame-level features, followed by two fully-connected layers with ReLU activation functions, batch normalization, and a softmax output layer. This network was optimized by large margin softmax losses. Embeddings of 512-dimensional bottleneck features are extracted from the second fully-connected layer. Table 2 summarizes the architecture of E-TDNN.

Our E-TDNN with residual transformations based x-vector systems were trained on two huge datasetes (as described in 2.3), respectively. These x-vector systems were served to our transfer learning systems.

3.2. Pre-trained and Transfer-Learning Systems

Our in-domain models used the weights of layers of x-vector systems, which pre-trained on huge datasets (as described in 2.3). All weight of layers of pre-trained systems were trained and optimized on the in-domain dataset while transfer-learning models only optimized fc layers. The in-domain models were trained on "Vox1_DeepMine-aug4" and "DeepMine-aug4" (as described in 2.3).

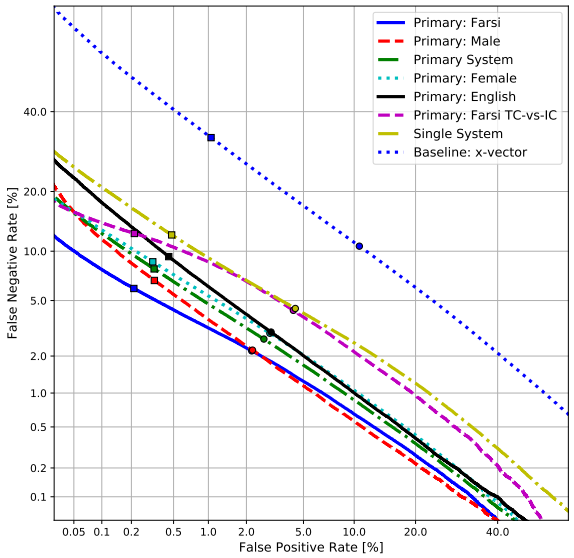

Figure 1: *DET curve of submitted system on the evaluation set.*

3.3. Two-tree Model

To solve the mismatch of pre-trained datasets and in-domain datasets, we proposed the two-tree model to increased the efficiency of weights of layers for transfer-learning models.

3.4. Fusion

We performed the fusion by computing the weighted average of the scores of selected systems.

4. Results

The challenge used minimum detection cost function from NIST SRE08 (minDCF08) [10] and Equal Error Rate (EER) as evaluation metrics. Table 3 and Table 4 show the performance of submitted systems on the progress and evaluation set of Task2, respectively. Our primary system reached 2.69% EER and 0.1118 minDCF08 on the evaluation set of Task2. Compared with baseline, on two test sets, our primary system relatively decreased minDCF08 more than 70%.

5. References

[1] K. A. Zeinali, Hossein nad Lee, J. Alam, and L. Burget, "Short-duration speaker verification (sdsv) challenge 2020: the challenge evaluation plan." arXiv preprint arXiv:1912.06311, Tech. Rep., 2020.

[2] H. Zeinali, H. Sameti, and T. Stafylakis, "DeepMine speech processing database: Text-dependent and independent speaker verification and speech recognition in Persian and English." in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 386–392.

[3] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," pp. 386–392, 2015.

[4] J. S. C. Arsha Nagrani and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[5] A. N. Joon Son Chung and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.

[6] H. Yamamoto, K. A. Lee, K. Okabe, and T. Koshinaka, "Speaker Augmentation and Bandwidth Extension for Deep Speaker Embedding," in *Proc. Interspeech*, 2019, pp. 406–410.

[7] 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.

[8] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

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

[10] A. F. Martin and C. S. Greenberg, "Nist 2008 speaker recognition evaluation: Performance across telephone and room microphone channels," in *Proc. Interspeech*, 2009, pp. 2579–2582.