The NICT submission for short-duration speaker verification challenge 2020

Peng Shen, Xugang Lu, Hisashi Kawai

National Institute of Information and Communications Technology, Japan

peng.shen@nict.go.jp

Abstract

In this paper, we describe the NICT speaker verification system for the text-independent task of the short-duration speaker verification (SdSV) challenge 2020. We firstly present the feature preparation. Then, x-vector-based front-ends band back-ends are introduced. For front-ends, we evaluated TDNN, extended TDNN (E-TDNN), factorized TDNN (F-TDNN), TDNN-LSTM, CNN-TDNN, and ResNet network configurations. For objective loss function, besides softmax function, we investigated angular softmax, additive angular margin softmax, and additive margin softmax functions. For back-ends, probabilistic linear discriminant analysis (PLDA), simplified PLDA, Cosine similarity, and neural network-based PLDA are investigated and explored. Finally, a greedy fusion method was used to obtain the final score for submission. Experimental results showed that our primary fusion yielded minDCF of 0.074 and EER of 1.50 on the evaluation subset, which was the 2nd best result in the text-independent speaker verification task.

Index Terms: speaker verification, short duration, SdSV challenge

1. Introduction

In this paper, we describe the NICT submission to the Short-duration Speaker Verification (SdSV) Challenge 2020. The main goal of the SdSV challenge 2020 is to evaluate new technologies for text-dependent and text-independent speaker verification in a short duration scenario. It is the first challenge with a broad focus on systematic benchmark and analysis on varying degrees of phonetic variability on short-duration speaker recognition. The SdSV challenge 2020 includes two tasks, where task 1 is defined as speaker verification in text-dependent mode: given a test segment of speech and the target speaker's enrollment data, automatically determine whether a specific phrase and the test segment was spoken by the target speaker. Task 2 is speaker verification in text-independent mode: given a test segment of speech and the target speaker enrollment data, automatically determine whether the test segment was spoken by the target speaker. Our work focus on the text-independent task, i.e., task 2.

In this work, we firstly investigated the influence of the data duration mismatch on front-ends and back-ends. Then, we built x-vector-based speaker embedding systems as front-ends. We investigated time-delay neural networks (TDNN) [1] and extended TDNN (E-TDNN) [2], factorized TDNN (F-TDNN) [3], TDNN followed with long short-term memory (TDNN-LSTM) recurrent neural networks, convolutional neural network TDNN (CNN-TDNN), and residual networks (ResNet). Besides softmax function, we investigated angular softmax (ASoftmax) [4], additive angular margin softmax (ArcSoftmax), and additive margin softmax (AMSoftmax) functions [5]. After speaker embeddings were extracted, several probabilistic linear discriminant analysis (PLDA)-based back-ends and Cosine similarity were used for scoring. Then, adaptive symmetric score normalization (AS-Norm) [6] was used to produce well-calibrated speaker verification scores. Finally, a greedy fusion method was used to obtain the final score for submission.

2. Datasets and feature extraction

2.1. Training data

The SdSV challenge 2020 is a fixed training condition task where the system should only be trained using a designated set. The fixed training set consists of VoxCeleb 1 and 2, LibriSpeech, and DeepMine datasets. DeepMine data for task 2 are the in-domain training data contains text-independent Persian utterances from 588 speakers. Non-speech data is allowed for data augmentation purposes. Other public or private speech data and task 1 in-domain data for task 2 training are forbidden.

2.2. Enrollment and test data

The enrollment data in task 2 consists of one to several variable-length utterances. The net speech duration for each model is roughly 3 to 120 seconds (after applying an energy-based VAD). Each trial in the evaluation contains a test utterance and a target model. The duration of the test utterances varies between 1 to 8 seconds. The whole set of trials is divided into two subsets: a progress subset (30%), and an evaluation subset (70%). The progress subset is used to monitor progress on the leaderboard. The evaluation subset is used to generate the official results at the end of the challenge.

2.3. Data preparation and feature extraction

We followed the training data preparation of the baseline x-vector system supplied by the SdSV challenge organizer. We firstly combined the VoxCeleb, LibriSpeech, and DeepMine in-domain data as the x-vector extractor training data. Then, data augmentation (additive noise, music, babble, and reverberation) as described in [7] was used on the whole training data. Because of this task focused on short-duration test data, to reduce the duration mismatch of the training data and test data, we picked up examples with 2 seconds (200 frames) for network training.

Three types of acoustic features were applied, i.e., the Mel-frequency cepstral coefficient (MFCC), perceptual linear predictive cepstrum (PLP), a log Mel-filter bank (FBANK). MFCC features were computed using 30 Mel-filter banks. The PLP analysis computed 20-order PLP-cepstra. FBANK features were estimated using 40 and 60 Mel-filter banks. The feature extraction was progressed with a frame window of 25 ms and a shift of 10 ms. The frames of silence and low signal-to-noise ration were removed with an energy-based voice activity detection (VAD) after doing feature extraction.

3. Speaker embedding front-ends

The model for extracting speaker embedding representations consists of three modules: a frame-level feature extractor, a statistics pooling layer, and utterance-level representation layers. In this work, by fixing the statistics pooling layer and utterance-level representation layers, we investigated the frame-level feature extractor with several neural networks for extracting the speaker embedding and different settings of the objective loss function.

3.1. Network

TDNN is the most commonly used for x-vector extraction [1]. Our TDNN network includes three time-delay layers and two fully connected layers. There are 512 channels except for the last one, which has 1500 channels. The kernel sizes are 5, 3, and 3; and dilation factors are 1, 2, and 3 for time-delay layers, respectively.

An extended TDNN architecture (E-TDNN) has been shown its effectiveness for extracting x-vectors [8]. Compared with TDNN, E-TDNN consists of one more time-delay layer and three fully-connected layers. The new fully connected layers are inserted into every two time-delay layers. The kernel sizes are 5, 3, 3, and 3; and dilation factors are 1, 2, 3, and 4, respectively. Therefore, the temporal context of E-TDNN is wider than that of TDNN. And E-TDNN has more parameters.

In our ResNet configuration, we replaced the TDNN network with a ResNet34 network [9]. A channel average pooling was applied to the output of the final layer of the ResNet. The dimension of the average pooling was 512. Then, the statistic pooling and utterance-level representation were processed.

We also borrowed some effective networks from speech recognition tasks. For example, the factorized TDNN (F-TDNN) [3] showed its effectiveness on many speech recognition tasks than the conventional TDNN network. We evaluated F-TDNN [1], TDNN-LSTM [2] and TDNN-LSTM with attention [3], and CNN-TDNN [4] networks [13]. In the speech recognition task, these networks had an L2 regularization setting to overcome overfitting, in this task, we removed the regularization setting.

3.2. Objective loss function

In conventional speaker embedding training, softmax-based categorical cross-entropy is commonly used as the objective loss function. In this work, besides softmax, we also evaluated angular softmax (ASoftmax) [4], additive angular margin softmax (ArcSoftmax) and additive margin softmax (AMSoftmax)-based functions [5].

4. Back-ends

With the extracted embedding vectors, we firstly applied in-domain global mean subtraction on training, enrollment, and test data. Then, linear discriminant analysis (LDA) was used to select the most speaker relevant feature and reduce the dimension of the original x-vector. To further reduce the variabilities between training data and testing data, in-domain whitening was applied before classifier. The in-domain whitening calculated the mean and covariance of the in-domain data and applied them to whiten the test data. Finally, length normalization was applied to the speaker discriminant vectors.

Network	Feature	Loss	Back-ends	MinDCF	EER
E-TDNN	FBANK40	Softmax	PLDA	0.215	5.07
E-TDNN	FBANK40	AMSoftmax	PLDA	0.194	4.58
E-TDNN	FBANK40	AMSoftmax	NPLDA	0.139	3.10
E-TDNN	FBANK40	AMsoftmax	Fusion	0.124	2.67
E-TDNN	FBANK40	Fusion	NPLDA	0.131	2.74
E-TDNN	Fusion	AMSoftmax	NPLDA	0.124	2.64
E-TDNN	FBANK40	Fusion	Fusion	0.113	2.41
E-TDNN	Fusion	Fusion	Fusion	0.111	2.33
Fusion	Fusion	Fusion	Fusion	0.075	1.51
Primary submission (evaluation subset)				0.074	1.50

Table 1: *Fusion investigation on progress subset and results.*

The first classifier was the Gaussian PLDA [10] with a full covariance residual noise term and a full-rank eigenvoice subspace. A simplified PLDA with 150 eigenvoices was also investigated. Finally, we further investigated Cosine similarity and a neural PLDA (NPLDA) [11]. The parameters of a PLDA system were used to initialize the NPLDA model, then the parameters were trained in a backpropagation setting. We used the DeepMine training data and their augmented data as the in-domain data. The LDA dimension was selected as 150 for Cosine similarity and 200 for other classifiers.

After scoring, all trial results were subject to score normalization. We utilized adaptive symmetric score normalization (AS-Norm) [6] in our systems.

5. Fusion and calibration

We implemented a greedy fusion algorithm to obtain the final submission. Firstly, all the subsystems are evaluated to obtain minDCF and EER values. Then, the top N best subsystems are selected as the candidate list. After that, we prepare new lists by adding a new subsystem to the candidate list. The linear logistic regression with the Bosaris toolkit [12] is used to fuse and evaluate the new lists. Then, the candidate list is updated by selecting the top N best lists. The final submission is obtained when there is no further improvement. In this work, N was set to 3.

6. Results

Figure 1 shows the investigation of fusion on features, loss functions, and back-ends. From the results, we can see that different features and back-ends obtained almost the same contribution when fusion was applied. A combination of different features, loss functions, and back-ends could further improve the performance. Our primary submission was obtained using the proposed greedy fusion method. Our primary submission yielded minDCF of 0.074 and EER of 1.50 on the evaluation subset, which was the 2nd best result in the text-independent task.

[1] egs/swbd/s5c/local/chain/tuning/run_tdnn_7r.sh

[2] egs/swbd/s5c/local/chain/tuning/run_tdnn_lstm_1n.sh

[3] egs/tedlium/s5_r2/local/chain/tuning/run_tdnn_lstm_attention_bs_1b.sh

[4] egs/swbd/s5c/local/chain/tuning/run_cnn_tdnn_1a.sh

7. References

[1] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, “Deep neural network embeddings for text-independent speaker verification,” in Proc. of *Interspeech*, 2017.

[2] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, “Speaker Recognition for Multi-Speaker Conversations Using X-Vectors,” in Proc. of *ICASSP*, 2019.

[3] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohamadi, and S. Khudanpur, “Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks,” in Proc. of *Interspeech*, 2018.

[4] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “Sphereface: Deep hypersphere embedding for face recognition,” in Proc. of *CVPR*, 2017.

[5] Y. Liu, L. He, and J. Liu, “Large margin softmax loss for speaker verification,” in Proc. of *Interspeech*, 2019.

[6] P. Matjka, O. Novotn, O. Plchot, L. Burget, M. D. Snchez, and J. ernock, “Analysis of Score Normalization in Multilingual Speaker Recognition,” in Proc. of *Interspeech*, 2017.

[7] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in Proc. of *ICASSP*, 2018.

[8] D. Snyder, et al., “The JHU Speaker Recognition System for the VOiCES 2019 Challenge,” in Proc. of *Interspeech*, 2019.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. of *CVPR*, 2016.

[10] S. Ioffe, “Probabilistic linear discriminant analysis,” in Proc. of *the 9th European Conference on Computer Vision*, 2006.

[11] S. Ramoji, P. Krishnan, and S. Ganapathy, “NPLDA: A Deep Neural PLDA Model for Speaker Verification,” in Proc. of *Odyssey* 2020.

[12] N. Brummer and E. De Villiers, “The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF,” in *NIST SRE11 Speaker Recognition Workshop*, Atlanta, Georgia, USA, 2011.

[13] D. Povey, et al., “The Kaldi speech recognition Toolkit,” in Proc. of *ASRU*, 2011.