System description of Team05 for SdSV Challenge 2020

Anonymous Author[1]

[1]Anonymous Affiliation

author@mail.com

Abstract

In this report, we describe the submission of Team05 to the SdSV Challenge 2020.

Index Terms: speaker recognition, speaker embeddings

1. Introduction

The SdSV Challenge 2020 includes two tasks: text-independent and text-dependent speaker verification. Since our solutions for both tasks are based on text-independent systems, first, we will focus on the text-independent scenario and discuss the text-dependent case towards the end of this report.

2. Experimental setup

2.1. Training data

For both tasks we used all available training data including VoxCeleb1 [1], VoxCeleb2 [2], LibriSpeech [3] datasets and the corresponding training parts of the DeepMine dataset [4] (due to challenge rules we used text-dependent part of DeepMine for Task 1 and text-independent part for Task 2) . Specifically, we combined the development parts of VoxCeleb1 and VoxCeleb2 datasets which have 1152 and 5994 speakers, respectively. This results in over a million speech segments in total. Then we added 2338 speakers from the LibriSpeech dataset. Finally, we included a random subset of the training partition of the DeepMine dataset which contains 90% of all speakers.

2.2. Development data

We used the trial list consisting of $37,720$ pairs provided on the Voxceleb1 website[1]. These trials include speakers from the test part of the VoxCeleb1 dataset. Also we created a list of $200,000$ trials from our test split of DeepMine dataset which includes 10% of available speakers.

3. System description

Our approach is based on recent advances in the field of speaker recognition where deep neural networks (DNNs) play an important role.

3.1. Pre-processing

We followed the Voxceleb recipe[2] from Kaldi for training DNNs used to extract utterance-level speaker embeddings. We added the following augmentations to the original speech segments:

- Reverberation using RIRs[3]
- Additive noise: Musan[4] noise
- Additive noise: Musan music
- Additive noise: Musan babble

Each type of noise was added with different SNRs as it was done in the Kaldi recipe. We did **not** apply any voice activity detection (VAD) to the input signals. As can be seen from the evaluation metrics, VAD was not a critical ingredient for building a relatively accurate speaker verification system for this data.

3.2. Features

We used two feature representations of audio signal to train different speaker embedder networks: 30-dimensional MFCCs and 40-dimensional log filterbanks. Our implementation is based on the open-source `python_speech_features`[5] package. Both features were extracted from signal frames of 25ms length with 10ms shift. Frequency limits were set to 20-7600 Hz. Both features were mean normalized within the entire feature sequence.

3.3. Embedding extractors

We used two different topologies of neural networks to extract speaker embeddings. The first one is the well-known x-vector topology proposed in [5]. We used the same network architecture as in the aforementioned Kaldi recipe. The second one is based on the ResNet34 topology from [6] which outperformed the x-vector approach in the recent VoxCeleb SRC Challenge 2019 [7]. It uses statistical pooling which accumulates mean and standard deviation statistics for the frame-level outputs to get a fixed-dimensional utterance-level representation. We compared a few alternative ResNet architectures including ResNet50 and found that the one used in [8] yields the best performance on the development data. We selected it for our experiments. Our implementations are based on the PyTorch framework [9].

Each network was trained with all available training data and then fine-tuned on the training split of the DeepMine dataset. We found that including DeepMine data to the training set leads to slightly better performance compared to using only VoxCeleb and Librispeech at the training stage.

We followed the two-stage training strategy described in [8]. First, the network was trained with the standard Softmax loss. Second, on the fine-tuning stage, the additive angular margin loss (further referred to as AAM-Softmax) was used after removing all the layers following the embedding layer. We used the AAM-Softmax loss with scale $s = 30$ and margin $m = 0.2$.

[1] http://www.robots.ox.ac.uk/~vgg/data/voxceleb/meta/veri_test.txt

[2] https://github.com/kaldi-asr/kaldi/blob/master/egs/voxceleb/v2/

[3] http://www.openslr.org/resources/28/rirs_noises.zip

[4] http://www.openslr.org/17/

[5] https://pypi.org/project/python_speech_features/

We did not find any considerable difference between this strategy and the annealing procedure proposed in [10] where the margin parameter of the loss function is gradually increased during training.

We used chunks of 200 frames for training both embedder networks. These chunks were obtained by random cropping of the training segments. In the testing stage, embeddings were extracted from the full-length feature sequences without any cropping.

The x-vector network was trained using Adam [11] optimizer while the ResNet network was trained using SGD with momentum=0.9 and weight decay=0.0001. We stopped training when EERs computed on the development data stopped decreasing.

3.4. Backend

We used different backends for the x-vector and ResNet based embeddings.

In the former case we used linear discriminant analysis (LDA) without dimensionality reduction together with the probabilistic LDA (PLDA) based scoring [12]. Prior to applying LDA, embeddings were centered with the center computed from the training split of the DeepMine dataset. The backend pipeline was trained on the embeddings extracted from the original speech segments without any augmentation.

For the ResNet based embeddings we used cosine similarity based scoring followed by the score normalization. Specifically, adaptive s-norm [13, 14] with 300 top-scoring impostors was applied to the raw cosine similarities. The impostor cohort was created by averaging embeddings for each speaker in the training split of the DeepMine dataset.

We found that LDA-PLDA based scoring does not help when embeddings are extracted with ResNet. At the same time, applying the regularized version of WCCN (RWCCN) [15] to the raw embeddings improves the system performance. In our implementation of RWCCN we added identity matrix with a small coefficient to the estimated within-class covariance matrix. While, in principle, this coefficient can be estimated on the development data we simply used 0.01 in all our experiments. Table 1 summarizes these results.

Backend	Training data	EER, %	minDCF
LDA-PLDA	VoxCeleb2	3.27	0.1646
LDA-PLDA	DeepMine	3.15	0.1610
RWCCN, cos, s-norm	DeepMine	2.02	0.0963

Table 1: *Comparison of different backends for embeddings extracted with ResNet34. Performance metric are borrowed from the challenge Leaderboard.*

We were unable to get considerable improvement by doing fusion of the ResNet and the x-vector based sub-systems, therefore we decided to replace the latter one by another ResNet.

Our primary system consists of two ResNet34 embedding networks where the first one was trained using all available data while the second one was trained on on the development parts of the VoxCeleb dataset. Both systems use the same backend which includes RWCCN before cosine scoring followed by the score normalization. Score fusion was done by computing the weighted average of the scores of the selected systems where the weights were set to 3 and 1, respectively. We used the first system for the single-system submission. Its performance is shown in the last row in Table 1.

4. Text-dependent task

In contrast to the text-independent case, in the text-dependent task of the challenge one needs to design a system for joint speaker & utterance verification (SUV).

Our solution for the text-dependent scenario is based on fusing the scores from independent speaker verification (SV) and utterance verification (UV) systems. Our SV system is identical to the one used in the text-independent scenario with the only difference that the part of the DeepMine dataset corresponding to this task was used for training. To be more specific, RWCCN together with adaptive s-norm were used to obtain the final SV scores. In this task we also used a subset of DeepMine data including 10% of speakers for evaluating our systems.

We built and compared two systems for utterance verification.

The first one is a GMM-based monophone HMM system that was trained on the union of english and farsi phones from the DeepMine dataset. We have used 13-dimentional MFCC with delta and delta-delta features. The decodings of the test data that were produced with the HMM were used to generate binary scores for the list of trials.

Since this system produces only binary verification labels we used the following rule to obtain the final score:

- same phrase – keep the original SV score unchanged,
- different phrases – subtract a large constant value from the SV score

We found that this strategy is comparable to a trainable score fusion, and we used it in our experiments. Our implementation of this system was based on the Kaldi toolkit.

The second UV system is based on a DNN with the same architecture as the x-vector extractor [5]. We trained this network to classify phrases on the DeepMine and VoxCeleb1&2 datasets. The speech segments from VoxCeleb were labeled as ‘other’ class during training. We found that including Voxceleb helps to increase classification accuracy for 10 phrases from the DeepMind dataset. Given the trained network we extract phrase embeddings and train the LDA projection with 10 classes on the training split of the DeepMine dataset. Finally, utterance verification is done by computing cosine similarities between the projected phrase embeddings. As in the previous case we obtained the (binary) hard labels by thresholding similarity scores. The threshold was determined using our development split of the DeepMine dataset. We found that this system leads to a slightly better overall performance, and we adopted it for further experiments.

We also explored an alternative approach to build a SUV system which is also based on a phrase classifier. Here we applied phrase-dependent affine transformations to the extracted speaker embeddings using the estimated phrase labels. Those transformations were jointly trained using the logistic affinity loss [16] on the binary labels aiming at segregating the ‘target-correct’ class from other types of trials: ‘target-wrong’, ‘impostor-correct’ and ‘impostor-wrong’. To this end we divided the training set into mini-batches such that each of those includes multiple instances of the same phrase uttered by the same speaker. The loss is computed using a square matrix of all pairwise cosine similarities within a mini-batch.

We used the Gaussian backend [17] to obtain the phrase labels from the LDA-projected phrase embeddings described above. This phrase classifier has accuracy above 99% on the development set.

Table 2 demonstrates the results for speaker & utterance verification with both approaches described above.

System	EER, %	minDCF
SV & UV score fusion	2.96	0.1196
Phrase-dependent projections	2.84	0.1194

Table 2: *Comparison of two strategies for obtaining SUV scores from given speaker embeddings and phrase labels. Performance metric are borrowed from the challenge Leaderboard. See the text for details.*

Our primary system consists of in independent SV subsystem based on ResNet34 and the x-vector based phrase classifier. The final SUV scores were obtained as cosine similarities between embeddings after after applying phrase-dependent projections described above.

5. References

[1] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[2] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[3] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[4] 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.

[5] 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.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[7] J. S. Chung, A. Nagrani, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, "Voxsrc 2019: The first voxceleb speaker recognition challenge," *ISCA Challenges*, 2019.

[8] H. Zeinali, S. Wang, A. Silnova, P. Matejka, and O. Plchot, "BUT system description to voxceleb speaker recognition challenge 2019," *CoRR*, vol. abs/1910.12592, 2019. [Online]. Available: http://arxiv.org/abs/1910.12592

[9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[10] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," *ArXiv*, vol. abs/1904.03479, 2019.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[12] A. Sizov, K. A. Lee, and T. Kinnunen, "Unifying probabilistic linear discriminant analysis variants in biometric authentication," in *Structural, Syntactic, and Statistical Pattern Recognition*, P. Fränti, G. Brown, M. Loog, F. Escolano, and M. Pelillo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 464–475.

[13] P. Matejka, O. Novotný, O. Plchot, L. Burget, M. D. Sánchez, and J. Cernocký, "Analysis of score normalization in multilingual speaker recognition," in *INTERSPEECH*, 2017.

[14] D. E. Sturim and D. A. Reynolds, "Speaker adaptive cohort selection for tnorm in text-independent speaker verification," in *in Proc. ICASSP*, 2005, pp. 741–744.

[15] H. Zeinali, H. Sameti, and L. Burget, "HMM-based phrase-independent i-vector extractor for text-dependent speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, pp. 1421–1435, 2017.

[16] J. Peng, R. Gu, and Y. Zou, "Logistic similarity metric learning via affinity matrix for text-independent speaker verification," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 704–709.

[17] M. McLaren, A. Lawson, Y. Lei, and N. Scheffer, "Adaptive gaussian backend for robust language identification," in *INTERSPEECH*, 2013.