BUT+Omilia System Description
VoxCeleb Speaker Recognition Challenge 2020

Niko Brummer, Lukáš Burget, Ondřej Glembek, Pavel Matějka, Ladislav Mošner, Ondřej Novotný, Oldřich Plchot, Johan Rohdin, Anna Silnova, Themos Stafylakis, Shuai Wang, and Hossein Zeinali

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czechia
Omilia - Conversational Intelligence, Athens, Greece
{matejkap,iplchot}@fit.vutbr.cz, tstafylakis@omilia.com

1. Introduction

In this report, we describe the submission of Brno University of Technology (BUT) and Omilia team to the VoxCeleb Speaker Recognition Challenge (VoxSRC) 2020 Track 1 and Track 2. Submitted systems for both Fixed and Open conditions are a fusion of 3 ResNet systems.

2. Experimental Setup

2.1. Training data, Augmentations

For all fixed systems, we used development part of VOXCELEB-2 dataset [1] for training. This set has 5994 speakers spread over 145 thousand sessions (distributed in approx. 1.2 million speech segments). For training DNN based embeddings, we used original speech segments together with their augmentations. The augmentation process was based on the Kaldi recipe[1] and it resulted in additional 5 million segments belonging to the following categories:

- Reverberated using RIRs[2]
- Augmented with Musan[3] noise
- Augmented with Musan music
- Augmented with Musan babel

2.2. Validataion datasets

We use the development data provided by the organizers[4] named VoxSRC-20 to monitor our performance.

2.3. Input features

We use different features for several systems with 40, 64 and 80 mel-filter banks with 16kHz sampling frequency, frequency limits 20-7600Hz, 25ms frame length. Features are subjected to short time mean normalization with a sliding window of 3 seconds.

3. DNN based Systems

All Deep Neural Network (DNN) based embeddings used Energy-based VAD from Kaldi SRE16 recipe[5]. For this challenge, we use the speaker embedding ResNet architectures analyzed below.

3.1. ResNet34 wide

This is a wide version of the 34-layer ResNet, that employs a modified version of Squeeze-and-Excitation (SE) [2, 3]. 80-dimensional filter-banks are used as acoustic features, while Kaldi-style and spectral augmentations are applied. The speaker embeddings are 256-dimensional and the loss is the angular additive margin with scale equal to 30 and margin increasing from 0.1 to 0.3 [4]. The sizes of the feature maps are 64, 128, 256 and 256 for the 4 ResNet blocks, while standard average pooling with both mean and std features is employed for squeezing the temporal dimension (attentive pooling yielded slight degradation). The SE layers are added to the first 2 ResNet blocks only, after experimentation with several configurations. The network and the training scheme have certain similarities with those proposed in [5]. We use stochastic gradient descent with momentum equal to 0.9 and initial learning rate equal to 0.2, which we halve after 15 failed epochs (note that each epoch corresponds to 400 iterations and not to the whole training set).

The implementation is based on TensorFlow ([6]) using a single 12GB-memory GPU and the model training takes about 4 days. In order to fit a minibatch of 256 examples in a single GPU, we split it into 16 "microbatches" of 16 examples each and we applied gradient accumulation.

3.2. ResNet152 fbank64

In our system, 256 dimensional speaker embeddings were extracted from a 152-layer ResNet [7] DNN. The network was trained on the development part of the Voxceleb 2 dataset (5994 speakers in 145k sessions), cut into 2-second chunks and augmented with noise, as described in [8] and available as a part of the Kaldi-recipe collection [9]. As input, we used 64-dimensional filter-banks extracted from the original 16 kHz audio with a window size of 25 ms and a 10 ms shift.

The loss function used for training the DNN was CosFace [10], with scaling parameter s set to 32 and margin parameter m linearly increased from 0.05 to 0.3 throughout the whole period of training. We ran 1 epoch (i.e., passing all training data once) of stochastic gradient descent optimization, throughout which we exponentially decreased the learning rate from 10^{-1} to 10^{-6}. Note that we scaled the learning rate by the number of parallel jobs to compensate for the dynamic range of the accumulated gradients, in our case by 3. The momentum and weight decay were kept constant at 0.9 and $5 \cdot 10^{-4}$, respectively. The batch size was set to 128, however, training on 3 GPUs in parallel virtually tripled the batch size. Also, due to large mem-

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

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

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

[4] https://github.com/a-nagrani/VoxSRC2020

[5] We did not find a significant impact on performance when using different VAD within the DNN embedding paradigm and it seems that a simple VAD from Kaldi performs very well for DNN embedding in various channel conditions.

ory requirements, the gradients were computed over 2 "micro-batches" of size 64 after which the update step was taken. Note that care needs to be taken while using this approach in connection with any model that uses batch-normalization (as our ResNet model does) as batch-norm statistics may get biased with decreasing batch size.

3.3. ResNet152 fbank40

Second ResNet 152 system was trained with the same settings as in section 3.2 with few differences.

- 40 mel filter banks
- trained on 1 GPU
- trained with 3 epochs - (passing all training data 3 times)

4. Backend

4.1. Cosine distance scoring and score normalization

The scores for all of the subsystems that we used were obtained with the cosine distance scoring. Prior to scoring the embeddings were centered with the mean of the embeddings extracted from VoxCeleb2 development set. The same set was used as a cohort for score normalization. We created the cohort by averaging x-vectors for each speaker; it consisted of 5994 speaker models.

All systems used adaptive symmetric score normalization (adapt S-norm) which computes an average of normalized scores from Z-norm and T-norm [11, 12]. In its adaptive version [12, 13, 14], only part of the cohort is selected to compute mean and variance for normalization. Usually X top scoring or most similar files are selected; we set X to 100 for all experiments.

4.2. Fusion

4.2.1. Fixed condition

As we did not have any data to train the fusion on for fixed condition and we were optimizing for calibration-insensitive minDCF, we performed the fusion by computing the weighted average of the scores of the three selected systems. The weights were hand-picked based on the performance of the individual systems. The corresponding weights were 0.6, 0.3 and 0.1 for systems 8, 4 and 5 from Table 1, respectively.

4.2.2. Open condition

In the open condition, we performed the calibration and fusion via logistic regression which was trained on a random subset of 50 thousand trials from the voxceleb_H condition to obtain scaling and weight factors for individual systems. Additionally, we shortened all segments in this voxceleb_H trial set in such a way that if the segment was longer than 1.5s, we took half of it. Individual systems were first pre-calibrated and afterwards fused. We have observed similar results on MinDCF metrics as using simple weighted average. The fused systems in the open condition were systems 4, 7 and 8 from Table 1.

5. Results and Analysis

In Table 1, we report results of individual subsystems and their fusions we selected for CLOSED and OPEN tracks of the evaluation. We report minimum Decision Cost Function metric (minDCF) and Equal Error Rate [%] (EER). The operating point of the minDCF is set to $p_{tar} = 0.05$ which is the same as primary evaluation metric.

6. Acknowledgements

The work was supported by Czech Ministry of Interior project No. VI20152020025 "DRAPAK", Czech Science Foundation under project No. GJ17-23870Y, Czech National Science Foundation (GACR) project NEUREM3 No. 19-26934X, and by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

7. References

[1] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 1086–1090.

[2] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[3] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos 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.

[5] Daniel Garcia-Romero, Gregory Sell, and Alan McCree, "Magneto: X-vector magnitude estimation network plus offset for improved speaker recognition," in *Speaker Odyssey*, 2020.

[6] Hossein Zeinali, Luka Burget, Johan Rohdin, Themos Stafylakis, and Jan Honza Cernocky, "How to improve your speaker embeddings extractor in generic toolkits," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6141–6145.

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

[8] 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.

[9] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[10] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[11] P. Kenny, "Bayesian speaker verification with heavy–tailed priors," keynote presentation, Proc. of Odyssey 2010, June 2010.

[12] Pavel Matějka, Ondřej Novotný, Oldřich Plchot, Lukáš Burget, Mireia Sánchez Diez, and Jan Černocký, "Analysis of score normalization in multilingual speaker recognition," in *Proceedings of Interspeech 2017*. 2017, pp. 1567–1571, International Speech Communication Association.

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

Table 1: *Results of the systems in the fusion for the VoxSRC 2020 challenge.*

	System	**VoxSRC 2020 dev**		**Voxceleb O**		**Voxceleb H**		**Voxceleb E**	
		minDCF	EER	minDCF	EER	minDCF	EER	minDCF	EER
1	ResNet34 fbank 80 - cos.dist	0.196	3.97	0.084	1.35	0.140	2.45	0.088	1.41
2	ResNet34-SE fbank 80 - cos.dist	0.178	3.51	0.077	1.16	0.124	2.14	0.076	1.17
3	ResNet34 fbank 80 - cos.dist + asnorm	0.177	3.67	0.072	1.25	0.124	2.21	0.078	1.29
4	ResNet34-SE fbank 80 - cos.dist + asnorm	0.161	3.25	0.067	1.05	0.112	1.94	0.067	1.09
5	ResNet152 fbank 40 - cos.dist + asnorm	0.166	3.13	0.058	0.86	0.114	1.88	0.070	1.06
6	ResNet152 fbank 64 - PLDA	0.182	3.70	0.066	0.97	0.124	2.22	0.077	1.23
7	ResNet152 fbank 64 - cos.dist	0.174	3.34	0.068	0.90	0.119	2.00	0.077	1.23
8	ResNet152 fbank 64 - cos.dist + asnorm	1.145	2.92	0.056	0.78	0.101	1.74	0.064	0.96
	Fusion CLOSED - 4 + 5 + 8	0.134	2.73	0.057	0.87	0.92	1.58	0.050	0.73
	Fusion OPEN - half - 4 + 7 + 8	0.134	2.62	0.050	0.72	-	-	0.056	0.86

[14] Yaniv Zigel and Moshe Wasserblat, "How to deal with multiple-targets in speaker identification systems?," in *Proceedings of the Speaker and Language Recognition Workshop (IEEE-Odyssey 2006)*, San Juan, Puerto Rico, June 2006.