UNISOUND System for VoxCeleb Speaker Recognition Challenge 2023

Yu Zheng[1], Yajun Zhang[1], Chuanying Niu[1], Yibin Zhan[1], Yanhua Long[2], Dongxing Xu[1]

[1]Unisound AI Technology Co., Ltd., Beijing, China

[2]SHNU-Unisound Joint Laboratory of Natural Human-Computer Interaction, Shanghai Normal University, Shanghai, China

Presenter：Yu Zheng

August 20，2023

Overview

Dataset

Data augmentation

Feature

System

Architectures

- ResNet
- RepVGG
- Pooling Layer
- Loss Function

Backend

- CMF
- AS-Norm
- QMF & Fusion

Training

Results

Conclusions

Dataset

We only used VoxCeleb2-dev as training data for both Track 1 & Track 2

Data augmentation

- Firstly, we adopted a 3-fold speed augmentation to generate extra twice speakers. Each speech segment was perturbed by 0.9 and 1.1 factor based on the SoX speed function.

- Secondly, we used RIRs and MUSAN to create extra four copies of the training utterances and the data augmentation process was based on the Kaldi.

Finally we obtained 17,982 speakers with 16,380,135 utterances.

Dataset

Features

- We extracted 80-dimensional log Mel filter bank with energy using Kaldi toolkit.
- The window size was 25 ms with a 10 ms frame shift.
- No voice activity detection (VAD) was applied.
- Chunks of features were mean-normalized before fed into the network.

System

Architectures

- Large-scale ResNet[1] and RepVGG[2] architectures as **backbones.**
- Multi-query multi-head attention (MQMHA) [3]**pooling layer** was attached after.
- A fully connected feed-forward layer with 512 dimensions is added after the pooling layer as **embedding layer**.
- **Loss function** is an AM-Softmax or AAM-Softmax[4].

[1] 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.
[2] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, “Repvgg: Making vgg-style convnets great again,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13 733–13 742.
[3] M. Zhao, Y. Ma, Y. Ding, Y. Zheng, M. Liu, and M. Xu, “Multiquery multi-head attention pooling and inter-topk penalty for speaker verification,” CoRR, vol. abs/2110.05042, 2021.
[4] Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4690–4699.

System

ResNet

- Each block was stacked with bottlenecks.
- The strides of blocks was (1, 2, 2, 2).
- Number of channels of blocks was (64, 128, 256, 512).

Table 1: *ResNets architecture*

Name	Layers of each block	Params(M)
ResNet101	3, 4, 23, 3	42.49
ResNet152	3, 8, 36, 3	58.13
ResNet242	3, 10, 64, 3	89.98
ResNet314	3, 16, 82, 3	111.77
ResNet518	6, 32, 128, 6	181.24

Including the MQMHA and embedding layers, the largest model, ResNet518 has a total of **227.46M** parameters.

System

RepVGG

We select RepVGG-B1 with 64 base channels.

Pooling layer

- MQMHA was used in each system. And the number of **query** was set to **4** while the number of **head** was **16**.
- We only used **standard deviation** in the pooling layer of the ResNet systems.

Loss function

- AM-Softmax and AAM-Softmax were used in different stages of training.
- The Sub-Center[1] method was introduced, and the number of center was set to 3.
- We also used the Inter-TopK[2]

[1] J. Deng, J. Guo, T. Liu, M. Gong, and S. Zafeiriou, "Sub-center arcface: Boosting face recognition by large-scale noisy web faces," in European Conference on Computer Vision. Springer, 2020, pp. 741–757.
[3] M. Zhao, Y. Ma, Y. Ding, Y. Zheng, M. Liu, and M. Xu, "Multiquery multi-head attention pooling and inter-topk penalty for speaker verification," CoRR, vol. abs/2110.05042, 2021.

System

Backend CMF score calibration

$$s_{AB} = \frac{1}{N}\sum_{i=1}^{N}\cos(x_i, y) = \frac{y}{N \cdot ||y||}\sum_{i=1}^{N}\frac{x_i}{||x_i||} \quad (1)$$

$$c = \sum_{i=1}^{N}\frac{x_i}{||x_i||} \quad (2)$$

$$s_{AB} = \frac{y}{N \cdot ||y||} \cdot c = \frac{1}{N}||c|| \cdot cos(y, c) \quad (3)$$

$$CMF_B = \frac{1}{N}||c|| \quad (4)$$

- Inspired by segment scoring[1], we proposed a consistency-aware score calibration method which used **Consistency Measure Factor(CMF)** scaling score.
- Suppose y is a embedding of audio A,(x_1, x_2, ..., x_N) are the embeddings of N segments from audio B which cut into N crops.
- Eq.(4) is the definition of CMF

[1] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in Interspeech 2018. ISCA, sep 2018. [Online]. Available: https://doi.org/10.21437%2Finterspeech. 2018-1929

System

CMF score calibration

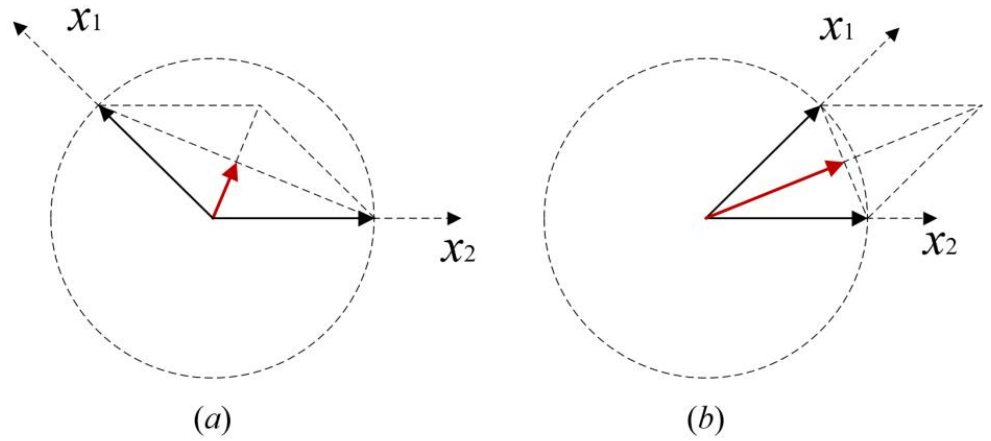

Figure 1: ***The connection between CMF and segment embeddings***. *Suppose $N = 2$ and the dimension of x_i is 2, and the red arrow represents $\frac{1}{N}c$. (a) When the angle between x_1 and x_2 is large, the modulus length of $\frac{1}{N}c$ is smaller; (b) When the cosine distance between x_1 and x_2 is large, the modulus length of $\frac{1}{N}c$ is large.*

- CMF reflects the degree of **consistency** or dispersion of the embeddings.
- Larger value of CMF indicates that the distribution of vectors is more concentrated.
- To some extent, CMF reflects the stability of audio voiceprint.

System

CMF score calibration

- Problem with segment scoring: when the segment length is shorter, it may be not friendly to judge the similarity, but it may increase the discrimination of CMF.
- To fix this problem, we only use **CMF as a scale to calibrate score** as Eq.(5)

$$score_{A,B} = CMF_A \cdot CMF_B \cdot cos(emb_A, emb_B) \qquad (5)$$

- For VoxCeleb1-test and VoxSRC23-dev, segment-length was 400, overlap was 200.
- For VoxSRC23-test, segment-length was 200 and overlap was 100.

System

AS-Norm[1]

- We selected the original VoxCeleb2 dev dataset without any augmentation.
- Embeddings were averaged speaker-wise.
- Top-400 highest scores are selected to calculate mean and standard deviation for normalization.

QMF and fusion

- Qualities of QMF[2]: speech duration, imposter mean based on AS-Norm, and magnitude of embeddings
- Audio with duration longer than **5s** was considered as long audio. We took the audio clipped from **2s to 5s** as the short audio.
- The ratio of target to nontarget is 1:1
- We fused the single system scores after AS-Norm, then used QMF to calibrate the fused score.

[1] W. Wang, D. Cai, X. Qin, and M. Li, “The DKU-DukeECE systems for VoxCeleb Speaker Recognition Challenge 2020,” arXiv preprint arXiv:2010.12731, 2020.
[2]J. Thienpondt, B. Desplanques, and K. Demuynck, “The IDLAB VoxCeleb speaker recognition challenge 2020 system description,” arXiv preprint arXiv:2010.12468, 2020

Training

First stage

- All VoxCeleb2 data with speed perturbation was used.
- 60 GPUs were used to train ResNet518 with 10 batch size in each GPU.
- 10 GPUs were used for other system training, batch size on each GPU was from 20 to 80 due to different model size.
- AM-Softmax with margin 0.2 and scale 35.

Second stage

- Removed the speed augmented part from the training set, and Only 5,994 classes were left.
- Changed the frame size from 200 to 600 while increased the margin from 0.2 to 0.5.
- AM-Softmax loss was replaced by AAM-Softmax loss.
- The Inter-TopK penalty was removed.
- Adopted smaller finetuning learning rate of 8e-5.

Results

Table 2: *Results on Development Sets*

Index	System	VoxCeleb1-O		VoxCeleb1-E		VoxCeleb1-H		VoxSRC23-val	
		EER(%)	$DCF_{0.01}$	EER(%)	$DCF_{0.01}$	EER(%)	$DCF_{0.01}$	EER(%)	$DCF_{0.05}$
S1	ResNet34	0.5037	0.0546	0.7123	0.0597	1.1246	0.0990	2.4956	0.1394
S2	ResNet101	0.4136	**0.0293**	0.6088	0.0450	0.9722	0.0755	2.1642	0.1210
S3	ResNet152	0.4401	0.0313	0.6047	0.0454	0.9313	0.0702	2.0121	0.1143
S4	ResNet242	0.4348	0.0370	**0.5789**	0.0413	**0.9042**	0.0666	2.0082	0.1188
S5	ResNet314	0.4454	0.0361	0.6339	0.0465	0.9701	0.0760	1.9809	**0.1072**
S6	ResNet518	**0.3712**	0.0299	0.5851	**0.0391**	0.9093	**0.0647**	**1.8678**	0.1074
S7	RepVGG-B1	0.4348	0.0368	0.6425	0.0549	1.0019	0.0793	2.2266	0.1290
Fusion	**S2∼S7**	**0.3659**	**0.0241**	**0.5651**	**0.0364**	**0.8662**	**0.0587**	**1.8327**	**0.1048**

The minDCF of our final submission is **0.0855** and the EER is **1.5880%** in VoxSRC23-test

Results

Table 3: *Backends on VoxSRC23-val*

Methods	VoxSRC23-val	
	EER(%)	$DCF_{0.05}$
ResNet101	2.8778	0.1516
+ AS-Norm	2.5463	0.1348
++ QMF	2.2655	0.1296
+ CMF	2.6555	0.1363
++ AS-Norm	2.4137	0.1259
+++ QMF	**2.1642**	**0.1210**

Table 4: *QMF and Segment score on VoxSRC23-test*

Methods	VoxSRC23-test	
	EER(%)	$DCF_{0.05}$
ResNet242 Segment score w/ 3s	2.1870	0.1126
ResNet242 Segment score w/ 2s	2.2780	0.1190
ResNet242 CMF w/ 3s	2.1490	0.1093
ResNet242 CMF w/ 2s	**2.0990**	**0.1080**
+QMF	**1.760**	**0.0993**

Conclusions

- We tried **larger model**, and got **better performance**.
- We proposed a consistency-aware score calibration method which used Consistency Measure Factor(**CMF**) scaling score, and brought **a huge performance boost** in this challenge.
- We found **only** using **standard derivation** in pooling layer **for ResNets** can get better performance.
- The final result of our system was **0.0855 minDCF and 1.5880% EER**. We achieved the **first place in Track 1** and **second place in Track 2** of VoxSRC 2023.

Thank You

Q&A