The SpeakIn System Description for CNSRC2022

Yu Zheng, Yihao Chen, Jinghan Peng, Yajun Zhang,
Min Liu, Minqiang Xu

SpeakIn Technologies Co. Ltd., ShangHai, China

SPEAK
国音智能 IN

Overview

1. Datasets
2. Systems description

   ResNet, RepVGG, ECAPA-TDNN
3. Training Strategy
4. Results and conclusions

Dataset

SPEAK IN 国音智能

Training dataset

Task 1 SV *fixed track*: ONLY *CN-Celeb.T* was used;

Task 1 SV *open track* and Task 2 SR *open track*: in total, there are 245497 speakers in this dataset. The datasets used for training included:

- Our internal large scale corpus.
- VoxCeleb 1+2.
- CN-Celeb.T.

For *CN-Celeb.T*, firstly we adopted a 3-folded speed augmentation to generate extra twice speakers. Each speech segment in this dataset was perturbed by 0.9 or 1.1 factor based on the SoX speed function.

SPEAK 国音智能 IN

Augmentation

Normal kaldi-based data augmentation is implemented

- Adding reverberation: artificially reverberation using a convolution with simulated RIRs from the AIR dataset.
- Adding music: taking a music file (without vocals) randomly selected from MUSAN, trimmed or repeated as necessary to match duration, and added to the original signal (5-15dB SNR).
- Adding noise: MUSAN noises were added at one second intervals throughout the recording (0-15dB SNR).
- Adding Babble: MUSAN speech was added to the original signal (13-20dB SNR).

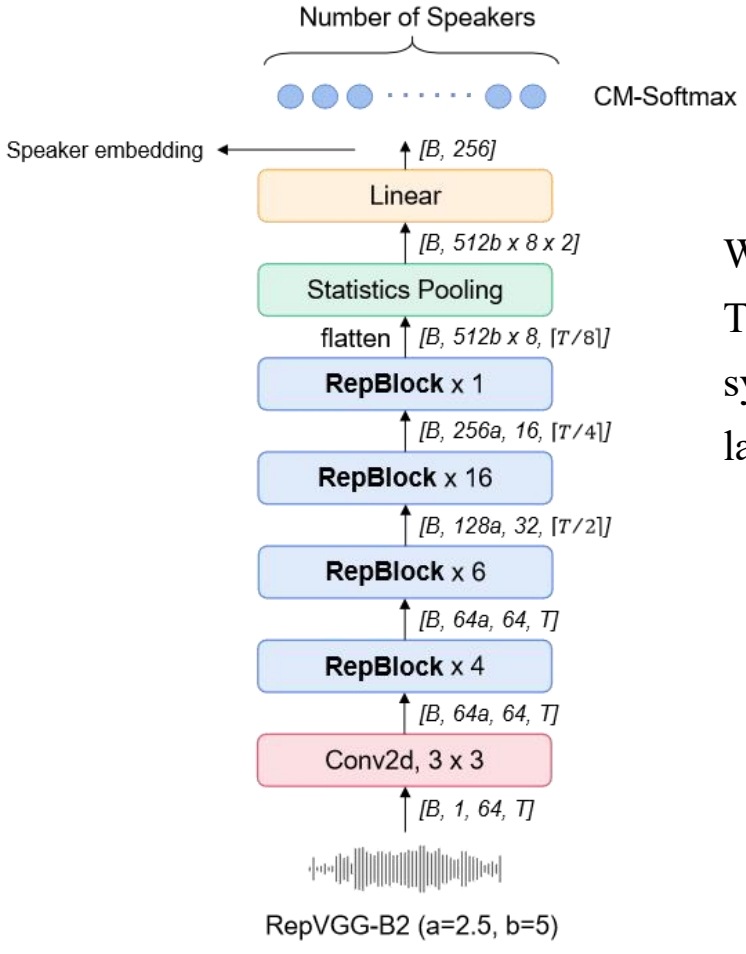
After augmentation, the amount of data is 5 times the original data

SPEAK
国音智能 IN

Feature

- We extracted both 81-dimensional log Mel filter bank energies based on Kaldi.
- The window size is 25 ms, and the frame-shift is 10 ms.
- without extra voice activation detection (VAD).
- All features were cepstral mean normalized in both our training modes.

System Description



We used ResNet, RepVGG, ECAPA-TDNN in our system. Take individual system RepVGG as an example, our system is composed of backbone, pooling, embedding layers and loss function.

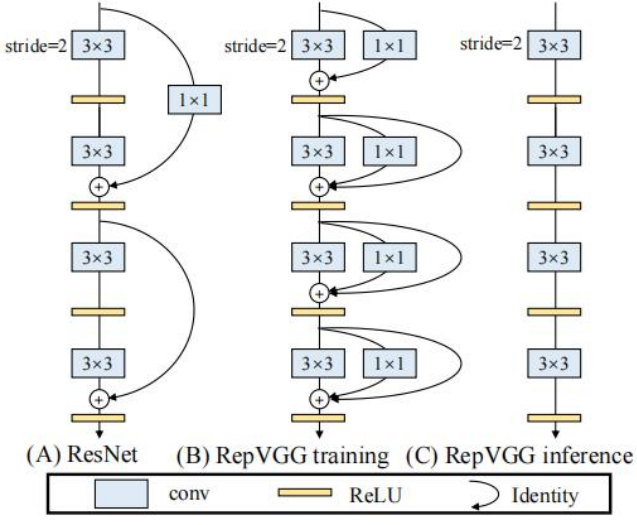
ResNet

We used ResNet-34, ResNet-74 and ResNet-101. The block to build resnet was bottleneck. The base channel was 64. The block numbers of resnet74 and resnet101 were (3, 4, 14, 3) and (3, 4, 23,3).

Layer name	ResNet-34	ResNet-74	ResNet-101
conv1	3×3, 64, stride1	3×3, 64, stride1	3×3, 64, stride1
conv2	3×3,64 3×3,64 ×3, stride1	1×1,64 3×3,64 ×3, stride1 1×1,256	1×1,64 3×3,64 ×3,stride1 1×1,256
conv3	3×3,64 3×3,64 ×4, stride2	1×1,128 3×3,128 ×4, stride2 1×1,512	1×1,128 3×3,128 ×4, stride2 1×1,512
conv4	3×3,64 3×3,64 ×6, stride2	1×1,256 3×3,256 ×14, stride2 1×1,1024	1×1,256 3×3,256 ×23, stride2 1×1,1024
conv5	3×3,64 3×3,64 ×3, stride2	1×1,512 3×3,512 ×3, stride2 1×1,2048	1×1,512 3×3,512 ×3, stride2 1×1,2048

RepVGG

The repvgg[1] structure is effective in speaker recognition. We select RepVGG-A2 as our backbones. The model adopt 64 base channels.



Name	Layers of each stage	a	b
RepVGG-A0	1, 2, 4, 14, 1	0.75	2.5
RepVGG-A1	1, 2, 4, 14, 1	1	2.5
RepVGG-A2	1, 2, 4, 14, 1	1.5	2.75
RepVGG-B0	1, 4, 6, 16, 1	1	2.5
RepVGG-B1	1, 4, 6, 16, 1	2	4
RepVGG-B2	1, 4, 6, 16, 1	2.5	5
RepVGG-B3	1, 4, 6, 16, 1	3	5

RepVGG models defined by multipliers *a* and *b*
Eg: [64*a*, 128*a*, 256*a*, 512*b*]

[1]Miao Zhao, Yufeng Ma, Min Liu, and Minqiang Xu, “The speakin system for voxceleb speaker recognition challange 2021,” arXiv preprint arXiv:2109.01989, 2021.

Pooling

In addition to statistics pooling layer, multi-query multi-head attention pooling mechanism layer (MQMHA) [1] was used to aggregate along the time across.

Loss function

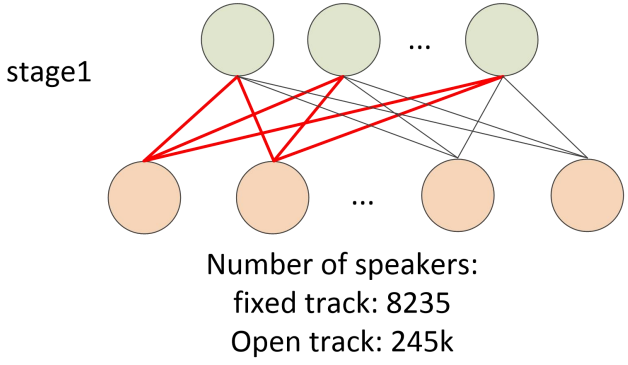
AMsoftmax, AAMsoftmax and CMsoftmax were used. the subcenter method [2] was introduced to reduce the influence of possible noisy samples.

$$L = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{s[\cos(\theta_{y_i}+m_1)-m_2]}}{e^{s[\cos(\theta_{y_i}+m_1)-m_2]}+\sum_{i\neq y_i}e^{s\cos\theta_j}}$$

[1] Miao Zhao, Yufeng Ma, Yiwei Ding, Yu Zheng, Min Liu, Minqiang Xu, Multi-query multi-head attention pooling and Inter-topK penalty for speaker verification, ICASSP2022

[2] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou, “Sub-center arcface: Boosting face recognition by large-scale noisy web faces,” in *European Conference on Computer Vision*. Springer, 2020, pp. 741–757.

Training Strategy



Stage1: training from scratch

- All data are used for training
  (eg: fixed track,3-fold speed aug 8235 spks)
- Chunksize is 200
- margin of AMsoftmax is 0.2
- Number of subcenter is 3

Stage2: domain adaptation & large-margin refinement

- Domain Adaptation(open track ONLY)
- Drop out the speed augmented part from the training set.
- Refine the network with these datasets. Extract the weight of the 2745 speakers in basic model as the initial weight of the current classifier.
- Turn off subcenter. Choose the weights with the largest vector length as the ”dominated center” and discard the other two for each id.
- Chunksize increases to 1000, margin of AAMsoftmax is 0.8.

Results and conclusions

Task 1 SV *fixed track*

5*System	CN-Celeb.E		CN-Celeb.E (submean asnorm)	
	eer	minc	eer	minc
ResNet34	7.8231	0.3755	7.4571	0.3518
Ecapa-tdnnL	8.8257	0.4086	8.6623	0.3914
ResNet74	7.7274	0.3785	7.3162	0.3475
RepVGG_A2	7.7387	0.3681	7.4233	0.3460
ResNet101	6.2518	0.3619	6.1335	0.3358
fused	-	-	5.9530	0.3185

Taking resnet34 as an example, the table shows the improvement of system results by refinement parameter adjustment, submean and asnorm.

System	CN-Celeb.E	
	eer	minc
ResNet34 baseline	8.7356	0.4179
+refinement(sc=3,chunk600,margin0.5)	8.8933	0.3898
+refinement(sc=1,chunk600,margin0.5)	7.7556	0.3859
+refinement(sc=1,chunk1200,margin0.8)	7.8231	0.3755
++submean+asnorm	7.4571	0.3518

Results and conclusions

Task 1 SV *open track*

2*System	CN-Celeb.E		CN-Celeb.E (submean asnorm)	
	eer	minc	eer	minc
RepVGG_A2	5.8519	0.2925	6.0321	0.2711
ResNet101	5.0577	0.2574	5.0183	0.2418
fused	-	-	4.6630	0.2384

Task 2 SR

2*System	SR.eval
	mAP
RepVGG_A2	0.5203
ResNet101	0.6106

Post-evaluation

system	eer minc	
	leaderboard	emb avg
ResNet34	7.4571 0.3518	6.9333 0.3338
Ecapa-tdnnL	8.6623 0.3914	7.8175 0.3603
ResNet74	7.3162 0.3475	6.6291 0.3123
RepVGG A2	7.4233 0.3460	6.7868 0.3234
ResNet101	6.1335 0.3358	**5.5477 0.3001**
fuse	5.9530 0.3185	**5.5252 0.2914**

system	eer minc	
	leaderboard	emb avg
RepVGG_A2	6.0321 0.2711	5.1929 0.2558
ResNet101	5.0183 0.2418	**4.4832 0.2315**
fuse	4.6630 0.2384	**4.2749 0.2268**

Results and conclusions

- 3-folded speed augmentation, and adaptive score normalization are effective.
- Domain mismatch degrades the performance, in-domain data for fine-tuning is useful.
- Long duration and large margin can improve system performance.
- Sub-center makes the model more robust.

SPEAK
国音智能 IN

THANKS