BUT System for VoxCeleb Speaker Recognition Challenge 2019

Hossein Zeinali, Shuai Wang, Anna Silnova, Pavel Matějka and Oldřich Plchot

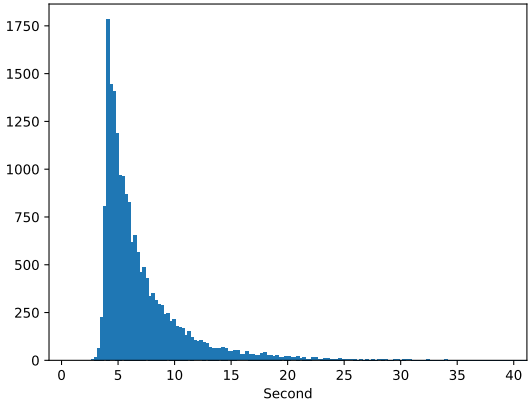
Brno University of Technology, Faculty of IT, IT4I Centre of Excellence, Czechia

September 13, 2019

VoxCeleb Speaker Recognition Challenge

- Text independent speaker verification using 16K data in two separate tracks:
  - **Fixed:** Using only development set of VoxCeleb2 for training.
  - **Open:** Using any data for training is allowed.
- Evaluation data:
  - Number of utterances: 19148
  - 208008 trials (contains 83 duplicate trials!)

- Duration histogram after removing silences using Kaldi VAD



Training Data and Augmentations

- Fixed condition:
  - Only development set of VoxCeleb2
  - 5994 speakers
  - distributed in approx. 1.2 million speech segments
- Open condition:
  - Development set of VoxCeleb1 and VoxCeleb2 - 7146 speakers
  - LibriSpeech dataset- 2338 speakers
  - DeepMine dataset - 1735 speakers
- Augmenting using standard Kaldi recipe
  - Reverberated using RIRs
  - Augmented with Musan noise
  - Augmented with Musan music
  - Augmented with Musan babel
- Development datasets: cleaned versions of Voxceleb1-O, Voxceleb1-E, and Voxceleb1-H

The used x-vector topology

Layer	Standard E-TDNN (5.8M)		BIG E-TDNN (20M)	
	Layer context	(Input) × output	Layer context	(Input) × output
frame1	$[t-2, t-1, t, t+1, t+2]$	$(5 \times K) \times 512$	$[t-2, t-1, t, t+1, t+2]$	$(5 \times K) \times 1024$
frame2	$[t]$	512×512	$[t]$	1024×1024
frame3	$[t-2, t, t+2]$	$(3 \times 512) \times 512$	$[t-4, t-2, t, t+2, t+4]$	$(5 \times 1024) \times 1024$
frame4	$[t]$	512×512	$[t]$	1024×1024
frame5	$[t-3, t, t+3]$	$(3 \times 512) \times 512$	$[t-3, t, t+3]$	$(3 \times 1024) \times 1024$
frame6	$[t]$	512×512	$[t]$	1024×1024
frame7	$[t-4, t, t+4]$	$(3 \times 512) \times 512$	$[t-4, t, t+4]$	$(3 \times 1024) \times 1024$
frame8	$[t]$	512×512	$[t]$	1024×1024
frame9	$[t]$	512×1500	$[t]$	1024×2000
stats pooling	$[0, T]$	1500×3000	$[0, T]$	2000×4000
segment1	$[0, T]$	3000×512	$[0, T]$	4000×512
segment2	$[0, T]$	512×512	$[0, T]$	512×512
softmax	$[0, T]$	$512 \times N$	$[0, T]$	$512 \times N$

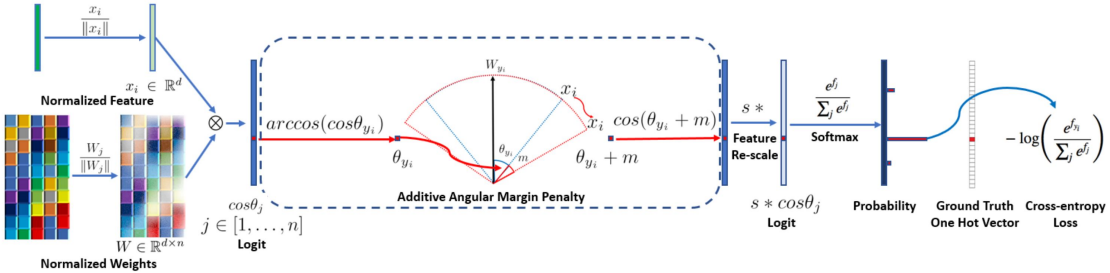
Res-E-TDNN: Standard E-TDNN with few residual connections in frame part and 768 outputs instead of 512. Parameters: 11.4M

ResNet34 topology (6M) - "r-vector"

Layer name	Structure	Output
Input	–	$40 \times 200 \times 1$
Conv2D-1	3×3, Stride 1	$40 \times 200 \times 32$
ResNetBlock-1	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$, Stride 1	$40 \times 200 \times 32$
ResNetBlock-2	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 4$, Stride 2	$20 \times 100 \times 64$
ResNetBlock-3	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 6$, Stride 2	$10 \times 50 \times 128$
ResNetBlock-4	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 3$, Stride 2	$5 \times 25 \times 256$
StatsPooling	–	5×256
Flatten	–	2560
Dense1	–	256
Dense2 (Softmax)	–	N

Fine-tuning networks with additive angular margin (AAM) loss

- Fine-tune a pre-trained DNN using SoftMax
- Remove all layers after embedding layer and fine-tune the whole network using AAM loss.



Scoring (G-PLDA or Cosine) and Normalization

Gaussian PLDA

- Centered embeddings using the mean vector of training data
- Applied LDA without dimension reduction and then did length normalization.
- Trained using 500K randomly selected utterances from original VoxCeleb2 development set (no augmentation)
- Speaker and channel subspace size were set to 312 (= 512 - 200).

Cosine distance

- Performed only for ResNet (i.e. r-vectors) with AAM
- Applied on top of centered r-vectors using the mean vector of training data
- Scores were normalized using adaptive S-Norm

Calibration and Fusion

Fixed condition

- Doing fusion by a weighted average
- Hand-picked weights based on the performance of the individual systems.
- Compensating the ranges' differences of the scores for different backends.
- The weight for ResNet with Cosine scoring is 0.4 and for TDNN with PLDA scoring is 0.1

Open condition

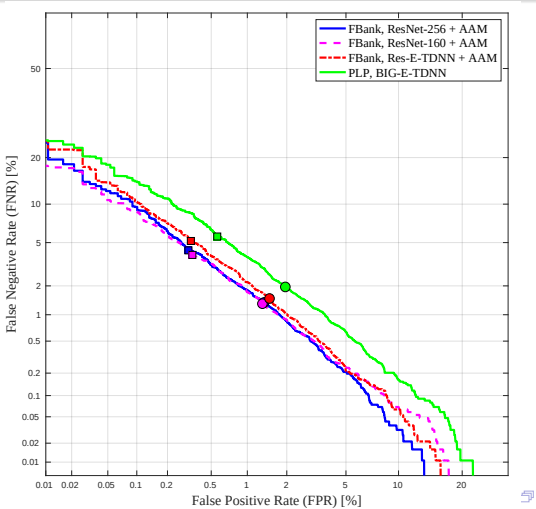
- Trained the fusion on the VoxCeleb1_O trials using logistic regression (LR)
- The scores of all systems were first pre-calibrated and then passed into the fusion.
- The output of the fusion was then again re-calibrated.

Comparison Results: MinDCF($P_{target} = 0.005$) and EER

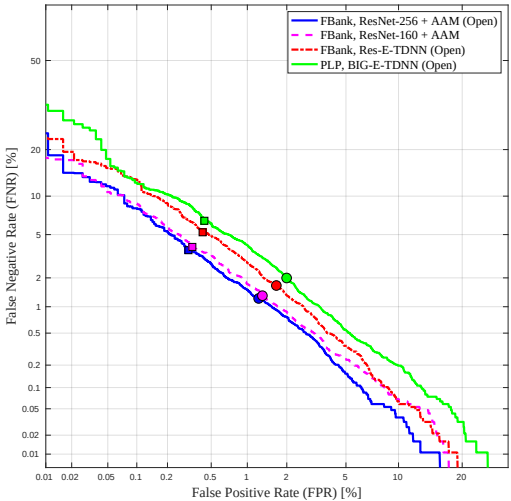
#	Fixed/Open	Acc. features	Embd NN	Backend	S-Norm	Vox1 O cleaned		Vox1 E cleaned		Vox1 H cleaned	
						MinDCF	EER	MinDCF	EER	MinDCF	EER
1	Fixed	FBANK	ResNet-256 + AAM	Cosine	Yes	0.166	1.42	0.164	**1.35**	**0.233**	**2.48**
2	Fixed	FBANK	ResNet-160 + AAM	Cosine	Yes	**0.154**	**1.31**	**0.163**	1.38	**0.233**	2.50
3	Fixed	FBANK	Res-E-TDNN + AAM	PLDA	No	0.181	1.46	0.185	1.57	0.299	2.89
4	Fixed	PLP	BIG-E-TDNN	PLDA	No	0.213	1.94	0.239	2.03	0.379	3.97
5	Open	FBANK	ResNet256 + AAM	Cosine	Yes	**0.157**	**1.22**	0.102	0.81	0.164	1.50
6	Open	FBANK	Res-E-TDNN	PLDA	No	0.195	1.65	0.170	1.42	0.288	2.70
7	Open	PLP	BIG-E-TDNN	PLDA	No	0.210	1.98	0.163	1.51	0.249	2.83
8	Fixed	Fusion 1+2+3+4 (Weighted average) (Eval EER: 1.42)				0.131	1.02	0.138	1.14	0.212	2.12
9	Open	Fusion 2+5+6+7 LR (Eval EER: 1.26)				0.118	0.96	0.098	0.80	0.160	1.51

Fusion of only two systems 1 and 3 on the evaluation set has EER of 1.49%

DET curves for Fixed conditions



DET curves for Open conditions



Conclusions

- Compare to the Voices challenge our single best system was improved about 25 % relative to our single best system for that challenge which was the overall single best system
- Our not fully tuned r-vector using ResNet (We have better models now!) considerably outperforms our tuned x-vector using E-TDNN.
- Interestingly, a simple weighted average performs as well as trained fusion for VoxSRC.
- Due to using pretty different topologies and layers, “r-vector” and “x-vector” are complementary embeddings.

Thanks for your attention!