Analysis and Description of ABC Submission to NIST SRE 2018

Oldrich Plchot, Pavel Matejka, Ondrej Novotny, Anna Silnova, Johan Rohdin, Mireia Diez, Ondrej Glembek, Lukas Burget, Martin Karafiat, Lucas Ondel, Frantisek Grezl, Niko Brummer, Patrick Kenny, Jahangir Alam, Gautam Bhattacharya, Themos Stafylakis, Jan Profant, Josef Slavicek, Michal Klco, Alicia Lozano-Diez

NUANCE
BRNO UNIVERSITY OF TECHNOLOGY
CRIM
UAM UNIVERSIDAD AUTONOMA DE MADRID
Omilia Conversational Intelligence

December 16th, Athens, NIST SRE 2018 workshop

Overview

- Data, calibration and fusion strategy
- Analysis with x-vectors on CMN2
- PLDA model adaptation
- WGAN adaptation for x-vector training
- 16KHz system - VAST
- Conclusions

Data, fusion and calibration

- For x-vector training we used
  - Data from previous evaluations
  - Voxceleb 1,2
  - MIXER6
  - Fisher English
- We no longer honored the PRISM split of data into train/test
- CMN2 and VAST conditions treated separately
  - VAST systems were trained on dev part of voxceleb 1,2 only (16KHz system)
    - we used SITW core-core and core-multi to monitor our performance
    - we also looked at the results on the very small VAST dev set
    - finally we **fused and calibrated** on VAST dev
  - CMN2 systems were trained on all available telephone data
    - we used both SRE18 dev and SRE16 eval to monitor our performance
    - we **fused and calibrated** on SRE18 DEV
- Simple logistic regression for calibration
  - systems were pre-calibrated before the fusion and the fusion itself was also re-calibrated
  - We chose a small target prior to cover wide range of operating points (0.005)
- We performed a **generative fusion** via MMFBG

Comparison - CMN2 (all trials equal)

SRE18 EVAL - all trials scored equally:

System	**EER[%]**	**minCprim**	**actCprim**
TFX_Xvec (HTPLDA)	7.76	0.57	0.57
TFX_Xvec (GPLDA)	7.71	0.52	0.53
Kaldi_Big_Xvec (GPLDA)	7.77	0.54	0.54
Fusion	**6.56**	**0.49**	**0.49**
Ivector	15.36	0.83	0.85

Comparison - VAST (all trials equal)

SRE18 EVAL - all trials scored equally:

System	**EER[%]**	**minCprim**	**actCprim**
1 Kaldi_Xvec_16k_Adapt	11.85	0.42	0.53
2 Kaldi_Xvec_16k	12.22	0.44	0.45
3 CRIM_noAdapt_PLP	12.74	0.61	0.68
Fusion	**11.44**	**0.44**	**0.53**
Fusion 2+3	11.70	0.46	0.48
Ivector_16k_PLP_BN	15.26	0.59	0.60

Analysis with X-vectors

- Same HTPLDA backend except for the TF model, where we apply additional LDA
- Steps to obtain JHU results: 1) remove Fisher, MIXER6 and SRE12, 2) Use Voxceleb 1+2 in original short chunks format, 3) apply GSM AMR codec on Voxceleb 1+2, 4) increase the number of archives from 140 to 900 with about half size -> 3x more examples per speaker

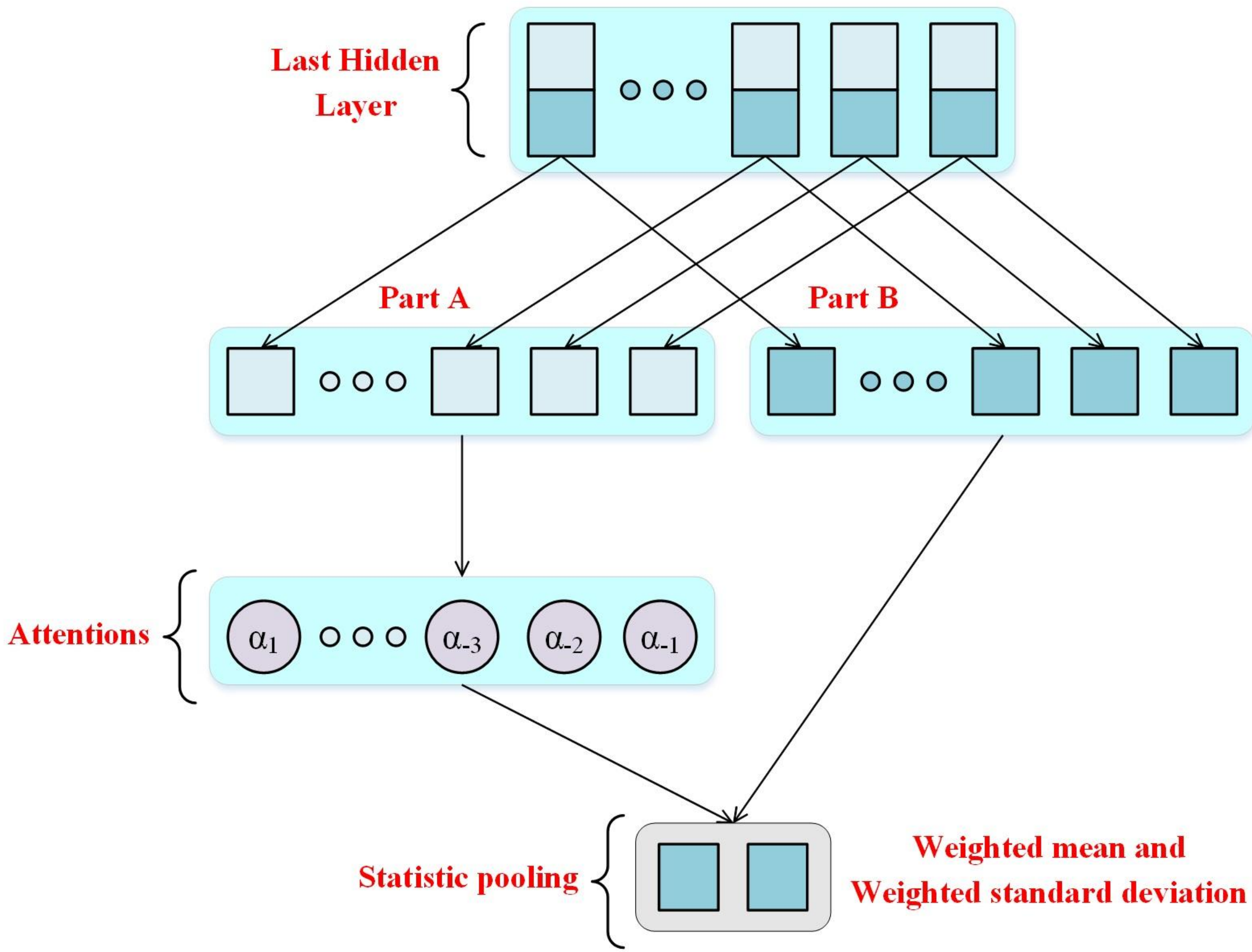
Training data	Topology	Comments	sre18_dev_cmn2		sre18_eval_cmn2	
			minC	EER, %	minC	EER, %
SRE4-8, 12, MIXER6-tel, Fisher English, Voxceleb1+2 concat, All Switchboards	Kaldi baseline		0.50	7.40	0.52	8.53
SRE4-8, 12, MIXER6-tel, Fisher English, Voxceleb1+2 concat, All Switchboards	TF implementation with attention, almost the same architecture as baseline	Our submission	0.46	5.96	0.50	7.45
SRE4-8, 12, MIXER6-tel, Fisher English, Voxceleb1+2 concat, All Switchboards	JHU architecture (COE)		0.45	6.12	0.49	7.49
SRE4-10, Voxceleb1+2 orig, All Switchboards	JHU architecture (COE)	JHU network	0.31	4.89	0.42	6.12
SRE4-10, Voxceleb1+2 orig, All Switchboards	JHU architecture (COE)	Our replicate of JHU network	0.32	4.80	0.41	5.90

Tensorflow X-vector implementation

Difference of Tensorflow and Kaldi baseline while the overall topologies are the same:

- Using CNN instead of TDNN
- Using LRelu instead of Relu
- Adding L2-Regularization to the segment level part of the network to prevent from over-fitting
- Using attention mechanism by doubling the size of the last hidden layer before pooling and using half-part of it for calculating the attentions (weights) and another half part for calculating weighted mean and standard deviation. Flowchart in the next slide.
- Source codes are available online in github: https://github.com/hsn-zeinali/x-vector-kaldi-tf

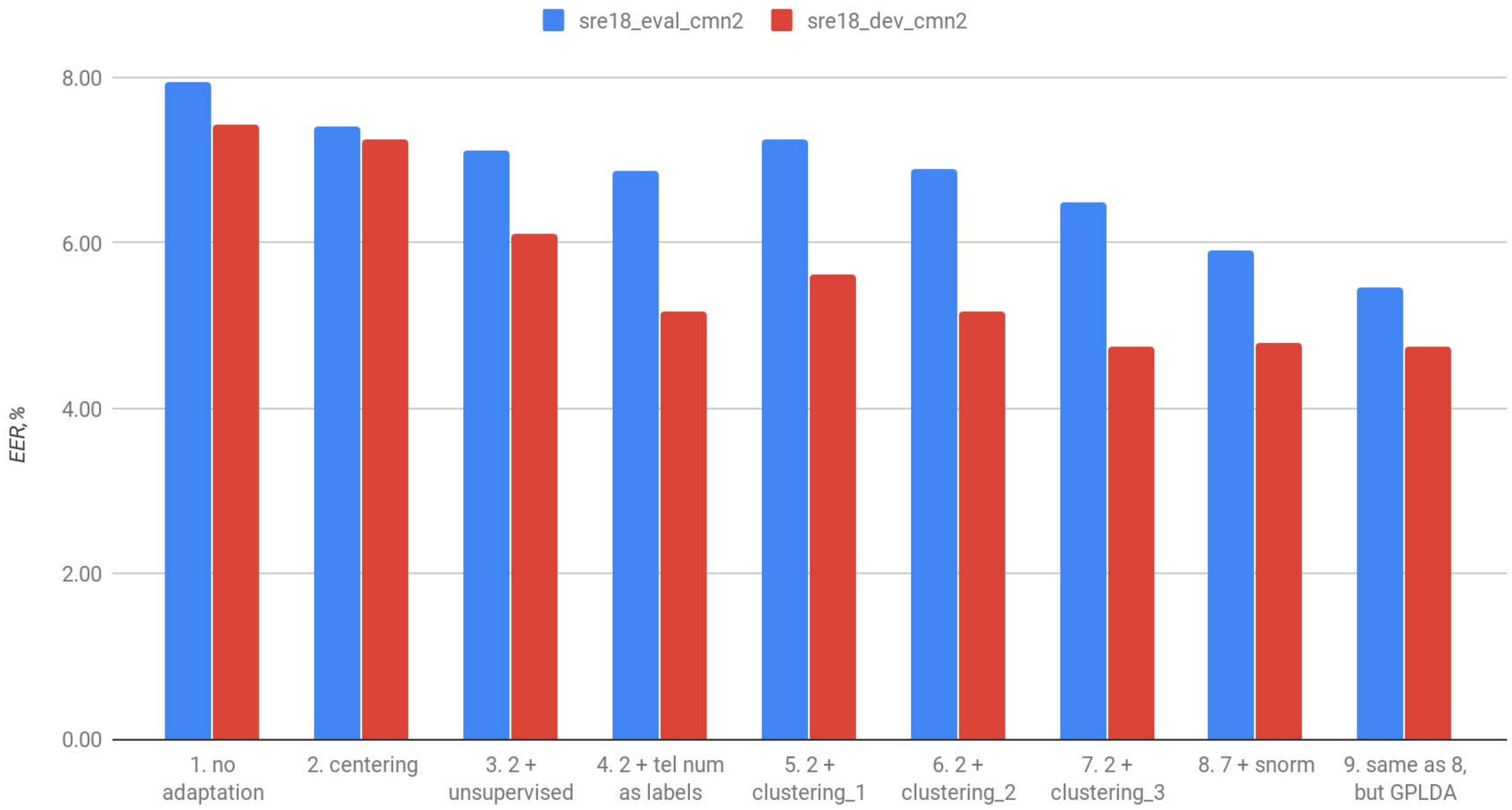
Adding attention to the network



Adaptation details

- Unsupervised
  - Kaldi-style adaptation
  - Excess of the covariance of adaptation data is equally distributed between within- and across-class covariance matrices of PLDA model
- Model interpolation
  - Small-scale model is trained on adaptation data, then within- and across- class covariances of original and adaptation model are interpolated
  - For training adaptation model, we used telephone number labels or labels obtained by clustering
  - Clustering is done by sampling from P(L | X, ϴ), where L, X and ϴ are the labels, data and model parameters respectively
  - We tried 3 options of selecting model parameters ϴ
    - clustering_1: ϴ is fixed to the parameters of the non-adapted model
    - clustering_2: ϴ is fixed to the parameters of the model with unsupervised adaptation
    - clustering_3: ϴ is fixed to the parameters of the model adapted using phone number labels

Model adaptation



16KHz system - VAST (single best system)

- X-vector based architecture from Kaldi baseline
- Training data (all recordings from session were concatenated into single one with 1 second of silence between every recording):
  - 16k VoxCeleb1 Development set
  - 16k VoxCeleb2 Development set
- more augmentations (512K vs 128K) compared to original X-vector recipe
- 9 epochs instead of 3
- Slightly extended context of time-delaying layers

VAST - analysis with diarization

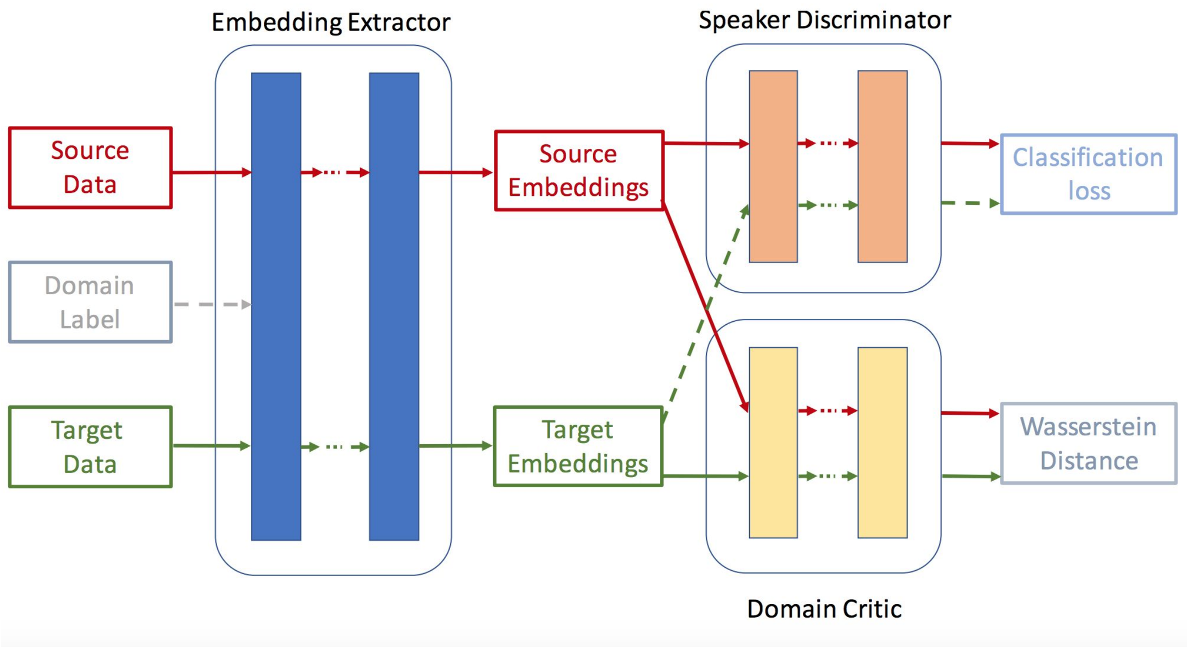
- Impact of the diarization

	minDCF0.05	minDCFsre16	EER [%]
DIARIZATION			
sitw_core-multi_eval	0.169	0.286	2.93
sre18_dev_vast	0.370	0.370	5.48
sre18_evl_vast	0.428	0.636	11.95
NO DIARIZATION			
sitw_core-multi_eval	0.212	0.327	4.59
sre18_dev_vast	0.342	0.556	3.64
sre18_evl_vast	0.454	0.631	11.13

- Results of our SITW i-vector system [EER %]: 7.34 (fusion)

WGAN adaptation for x-vector training

- Adversarial adaptation is applied on x-vectors
- X-vector extractor is the “generator”
- Supervised vs unsupervised adaptation is explored
- Language information (english / non-english) is used as side information in the TDNN
- Wasserstein loss is used in the discriminator (critic)



WGAN adaptation results

	Development set			Evaluation set		
	EER	DCF0.01	DCF0.005	EER	DCF0.01	DCF0.005
Baseline	9.528	0.615	0.670	10.011	0.629	0.699
Sup	9.208	0.603	0.650	9.589	0.615	0.688
Adv	9.668	0.637	0.678	10.347	0.626	0.690
Adv+Sup	8.008	0.583	0.634	8.889	0.593	0.667
Adv+Lan+Sup	**7.892**	**0.552**	**0.597**	**8.878**	**0.585**	**0.653**

- Adversarial adaptation on its own deteriorates the performance
- The combination of adversarial and supervised adaptation is effective
- Adding language information helps
- Better than PLDA adaptation on development set but not on evaluation set
- Difficult to combine WGAN adaptation and PLDA adaptation

Conclusions

- X-vectors outperforming i-vectors both in telephone and microphone conditions
- There is still room to improve our systems by exploiting the new and large Voxceleb dataset
  - We can now develop nice wideband system
- Less diversity in subsystems (all X-vectors)
  - relatively easy calibration and fusion (simple LR)
  - small gains from fusion
- Adaptation with soft labels (tel. numbers)

THANK YOU

We are happy for the dataset with a lot of room for improvement and research :)

GPLDA vs HTPLDA

- For both GPLDA and HTPLDA, speaker subspace is 150-dimensional.
- For all experiments but the last one, the channel subspace has the same dimensionality as the original x-vector or x-vector after LDA.
- LDA reduces the dimensionality from 512 to 300.
- For the two covariance model LDA is applied to reduce the sizeof x-vectors to 150.
- The results are presented for SRE18 evaluation condition

	HTPLDA		GPLDA	
X-vector preprocessing	EER, %	minC	EER, %	minC
no preprocessing	6.49	0.46	9.07	0.52
LN	7.23	0.52	10.43	0.61
LDA	5.96	0.46	8.22	0.49
LN+ LDA	6.53	0.49	9.49	0.57
LDA + LN	5.84	0.46	5.68	0.47
LN + LDA +LN	5.92	0.46	5.95	0.47
LN+ LDA150+LN two covariance model	-	-	6.02	0.45

Model adaptation

- For HTPLDA backend, no preprocessing of x-vectors is done
- For PLDA backend, LDA and length normalization are applied to x-vectors as a preprocessing step

	sre18_dev_cmn2		sre18_eval_cmn2	
	minC	EER, %	minC	EER, %
1. no adaptation	0.52	7.44	0.59	7.95
2. centering	0.54	7.24	0.50	7.41
	0.50	6.11	0.51	7.12
	0.42	5.18	0.48	6.87
	0.39	5.61	0.51	7.25
	0.42	5.17	0.50	6.9
	0.37	4.75	0.46	6.49
8. 7 + snorm	0.32	4.80	0.41	5.90
9. same as 8, but PLDA	0.33	4.75	0.37	5.47