INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

State-of-the-art Speaker Recognition for Telephone and Video Speech: the JHU-MIT Submission for NIST SRE18

Jesús Villalba[1], *Nanxin Chen*[1], *David Snyder*[1,2], *Daniel Garcia-Romero*[2], *Alan McCree*[2], *Gregory Sell*[2], *Jonas Borgstrom*[3], *Fred Richardson*[3], *Suwon Shon*[4], *François Grondin*[4], *Réda Dehak*[5], *Leibny Paola García-Perera*[1], *Daniel Povey*[1,2], *Pedro A. Torres-Carrasquillo*[3], *Sanjeev Khudanpur*[1,2], *Najim Dehak*[1]

[1]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, US
[2]Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore, MD, US
[3]MIT Lincoln Laboratory, Lexington, MA, US
[4]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, US
[5]LSE-EPITA, Villejuif, France

jvillalba@jhu.edu

Abstract

We present a condensed description of the joint effort of JHU-CLSP, JHU-HLTCOE, MIT-LL., MIT CSAIL and LSE-EPITA for NIST SRE18. All the developed systems consisted of x-vector/i-vector embeddings with some flavor of PLDA back-end. Very deep x-vector architectures–Extended and Factorized TDNN, and ResNets– clearly outperformed shallower x-vectors and i-vectors. The systems were tailored to the video (VAST) or to the telephone (CMN2) condition. The VAST data was challenging, yielding 4 times worse performance than other video based datasets like Speakers in the Wild. We were able to calibrate the VAST data with very few development trials by using careful adaptation and score normalization methods. The VAST primary fusion yielded EER=10.18% and Cprimary=0.431. By improving calibration in post-eval, we reached Cprimary=0.369. In CMN2, we used unsupervised SPLDA adaptation based on agglomerative clustering and score normalization to correct the domain shift between English and Tunisian Arabic models. The CMN2 primary fusion yielded EER=4.5% and Cprimary=0.313. Extended TDNN x-vector was the best single system obtaining EER=11.1% and Cprimary=0.452 in VAST; and 4.95% and 0.354 in CMN2.

1. Introduction

The National Institute of Standards and Technology (NIST) regularly conducts speaker recognition evaluations (SRE) to assess the state-of-the-art of the technology [1]. These evaluations focus on the speaker detection task, i.e., given one or more enrollment recordings and a test recording, we need to decide whether the enrollment speaker is also present in the test. Along the years, NIST has been increasing the difficulty of the evaluation conditions. First SRE campaigns were only centered on telephone conversational speech [2, 3]. In SRE08-12, NIST introduced far-field microphone interview speech [4, 5, 6]. SRE16 brought significant changes [7]. Although it focused again on telephone speech; for the first time, the data was non-English speech collected outside North America. This was a major difficulty since the training data was mainly English speech collected in the US. Just a small amount of unlabeled adaptation data was provided to correct distribution shift due to language and channel mismatch. For SRE18 [8], NIST decided to maintain the non-English condition. This time, they selected Arabic language collected in Tunisia through PSTN and VoIP networks. Furthermore, NIST added a new condition including speech from amateur Internet videos (VAST) [9]. As consequence, VAST spans a wide range of quality levels, including noise, reverberation and other artifacts that complicate speaker verification. These recordings usually contain multiple speakers so diarization was required to isolate the speaker of interest.

In this paper, we analyze the JHU-MIT submission to NIST SRE18. This is the joint effort of teams at Johns Hopkins CLSP and HLTCOE, MIT Lincoln Laboratory, MIT CSAIL and LSE-EPITA. All our systems consisted of a neural network (a.k.a. x-vector) [10] or i-vector [11] embedding followed by some form of PLDA [12] back-end. We explored several types of x-vectors differing in network topology and pooling methods. We tested TDNN [7, 10], E-TDNN [13], factorized TDNN [14], and ResNet (2D convolutions) [15] topologies. We also tested mean plus standard deviation; learnable dictionary encoder (LDE) [16] and multi-head attention pooling. We adapted the back-ends to the video condition or to the Arabic telephone condition. Primary submissions were a fusion of Extended TDNN, Factorized TDNN and ResNet x-vectors; while the best single system was JHU-HLTCOE E-TDNN x-vector. These systems can be considered the current state-of-the-art in text-independent speaker recognition technology.

The rest of the paper is organized as follows. Section 2 describes the training, development and evaluation data. Section 3 describes the acoustic features and VAD. Section 4 discusses the x-vector variants. Section 5 describes the PLDA back-ends. Section 6 describes the diarization. Section 7 summarizes the calibration, fusion and submissions. Section 8 presents and analyzes the results. Finally, Section 9 shows the conclusions.

2. Datasets

2.1. Evaluation data

NIST SRE18 consisted of two conditions. On the one hand, we had telephone speech in Tunisian Arabic recorded in Tunisia from the *Call My Net 2* corpus (CMN2). Given that most training data available is English recorded in the US, this condition is most challenging. On the other hand, we had speech from internet videos extracted from the VAST corpus. These are amateur videos so a wide range of acoustic conditions may be expected. Also, videos may contain multiple speakers, so diariza-

http://dx.doi.org/10.21437/Interspeech.2019-2713

tion is needed to isolate the target speaker. In the enrollment side, ground truth diarization marks were provided.

2.2. Training data

The datasets used for training included Switchboard phase1-3 and cellular1-2; NIST SRE04-10 as prepared by the SRE16 Kaldi recipe[1]; NIST SRE12 telephone data (SRE12-tel) and phone-calls recorded through far-field microphone (SRE12-micphn); MIXER6 telephone (MX6-tel) and microphone phone-calls (MX6-micphn); VoxCeleb 1 and 2 where we concatenated the segments belonging to the same original video into a unique segment (VoxCelebCat); and Speaker in the Wild dev core (single speaker segments) (SITW-dev-core). Using concatenated VoxCeleb helps to balance the weight of each video in the x-vector training and avoids including within-session variability in the within-class covariance of the PLDA.

We built 8 kHz and 16 kHz versions of our systems. For the 8 kHz systems, the HLTCOE team trained x-vectors using Switchboard, SRE04-10 and VoxCelebCat; and the rest of teams–denoted from now on as CLSP-MIT– used all the above datasets. Datasets originally at 16 kHz were downsampled to 8 kHz. For the 16 kHz systems, the HLTCOE team used just VoxCelebCat while the CLSP-MIT teams also used microphone data form SITW-dev-core, MIXER6 and SRE12. In total, we used around 13K speakers for 8kHz and 7.5K speakers for 16 kHz. This data was augmented with noise, babble and music from the MUSAN corpus[2]; and reverberation from the Aachen impulse response database (AIR)[3]. JHU-HLTCOE used MX6-micphn instead of MUSAN to create babble noise and it also used codecs on VoxCeleb to simulate GSM phone encoding[4].

To train PLDA for the telephone condition, we took the NIST SRE telephone utterances from the x-vector training lists ($\sim$4.5K speakers). For the video condition, we took just the 16 kHz utterances($\sim$7K speakers).

Other datasets were used for back-end adaptation and score normalization. They are SITW-dev-core; SITW-dev-test-diarized (Segments obtained from diarizing the SITW dev multi-speaker recordings); SRE18-dev-unlabeled (Tunisian Arabic data with telephone number labels but not speaker labels); and SRE18-dev-VAST-diarized (Segments obtained from diarizing the SRE18 development VAST data). SRE18-dev-unlabeled was used for centering, PLDA adaptation and score normalization for the telephone condition. SITW-dev-core plus SITW-dev-test-diarized, denoted as SITW-dev-diar, were used to center the SITW eval set; and SITW-dev-diar plus SRE18-dev-VAST-diarized, denoted as SITW-SRE18-dev-diar, were used to center SRE18 VAST and for score normalization of the video condition.

2.3. Development data

The development datasets were used to train fusion and calibration; and measure performance. For the CMN2 condition, we used the development set provided by the organizers. For the VAST condition, the development set provided by the organization was too small (only 270 trials) to provide reliable performance estimation. Also, there were only around 2-3 false alarm errors at the $P_{\mathcal{T}} = 0.05$ operating point, which was not enough to train calibration. Thus, we decided to use the SITW eval core-multi condition, which also consists of speech from video and also requires diarization on the test side.

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

[2] http://www.openslr.org/resources/17

[3] http://www.openslr.org/resources/28

[4] http://www.3gpp.org/ftp/Specs/archive/26_series/26.073/26073-800.zip

3. Feature extraction

x-Vectors Systems based on time delay networks used 23 MFCC for 8KHz; and 30 (HLTCOE) or 40 MFCC (CLSP-MIT) for 16 KHz. Systems based on ResNets used 23 and 40 log-Mel filter-banks for 8 and 16 KHz respectively. i-vector systems added first and second derivatives to the MFCC. Features were short-time centered before silence removal with a 3 seconds sliding window. Most systems used Kaldi energy VAD. Only 16 KHz systems based on F-TDNN x-vectors used a neural network VAD based on [17]. The system was trained on NIST SRE10 corpus with added noise and reverberation.

4. x-Vector embeddings

Neural network embeddings (a.k.a. x-vectors) are obtained using a neural network trained to classify the speakers in the training set [7, 10]. x-Vector networks are divided into three parts. First, an encoder network extracts frame level representations from the acoustic features. This is followed by a global temporal pooling layer that produces a single vector per utterance. Finally, a feed forward classification network processes the pooling vector to produce speaker class posteriors. Typically in the evaluation phase, the x-vector is obtained from the first affine transform after the pooling layer, while the last layers of the network are discarded. Different x-vector systems are characterized by different encoder architectures; pooling methods and training objectives. Categorical cross-entropy is the usual x-vector objective but we also tested angular softmax loss [18]. Angular softmax has stronger requirements for correct classification, which generates an angular classification margin between embeddings of different classes [16, 19].

4.1. Encoder Networks

4.1.1. TDNN

Time delay networks are the ones used in most x-vector papers [10] and was our baseline system. It was composed of two time-delay layers (a.k.a 1D dilated convolutions) and two fully connected layers. All layers had 512 channels except the last one, which had 1500 channels. Time delay layers had kernel sizes 5, 3 and 3; and dilation factors 1, 2 and 3 respectively.

4.1.2. E-TDNN

The Extended TDNN architecture (E-TDNN) [13] has slightly wider temporal context w.r.t. the previous TDNN (due to and extra time-delay layer), and interleaves dense layers in between the convolutional layers (equivalent to the 1x1 convolutions used in computer vision architectures). In summary, E-TDNN had 1 time-delay layer with kernel 5 and 3 layers with kernel 3. Dilation factors were 1, 2, 3 and 4 respectively. Each time-delay layer was followed by a fully connected layer.

4.1.3. F-TDNN with skip connections

The factorized TDNN (F-TDNN) [14], reduces the number of parameters of the network by factorizing the weight matrix of each TDNN layer into the product of two low-rank matrices. The first of those factors is constrained to be semi-orthogonal, which helps to assure that we do not lose information when projecting from the high dimension to the low-rank dimension. The authors of [14] found that; instead of factorizing the TDNN

layer into a convolution times a feed-forward layer; it is better to factorize the layer into two convolutions with half the kernel size. For example, instead of using context (-2, 0, 2) in the first low-rank factor and no context in the second factor, it is better to use context (-2,0) in the first factor and (0, +2) in the second factor. We also introduced skip connections between the low-rank interior layers of the F-TDNN. The prior layers were concatenated to the input of the current layer, instead of added like in ResNet [15].

In summary, our F-TDNN consisted of a TDNN layer of kernel size 5 and 512 channels; 8 F-TDNN layers with 1024 channels and internal dimension 256; and a fully connected layer with dimension 2048. The kernel sizes for F-TDNN layers are (3,1,3,1,3,3,3,1), and the dilation factor is 3 for all of them except the first one, which is 2. Layer 5 receives skip connections from layer 3; layer 7 from layers 2 and 4; and layer 9 from layers 4, 6 and 8.

4.1.4. ResNet 2D

TDNN layers are replaced by a residual network with 2D convolutions. We used a residual network with 34 layers (ResNet34) as described in [15]. This was implemented in Pytorch while the others were implemented in Kaldi.

4.2. Pooling Methods

The basic x-vector framework just compute the mean and standard deviation to obtain a single vector per utterance. Meanwhile, the learnable dictionary encoder (LDE) [20, 16] assumes that frame level representations are GMM distributed in C clusters and it learns a dictionary with the centers of those clusters. The component posteriors are obtained as, $w_{t,c} = \frac{\exp(-\frac{1}{2}s_c\|\mathbf{x}_t-\boldsymbol{\mu}_c\|^2+b_c)}{\sum_{c=1}^{C}\exp(-\frac{1}{2}s_c\|\mathbf{x}_t-\boldsymbol{\mu}_c\|^2+b_c)}$ where s_c is an isotropic precision; and b_c includes the log-weight and log-normalizing constant of the Gaussian. Then, we compute a component dependent mean $\mathbf{e}_c$ and concatenate all $\mathbf{e}_c$ to obtain a super-vector, which has the same role as the super-vector mean in i-vectors. This super-vector is projected to a lower dimension to obtain the final embedding. This projection has the same role as the total variability matrix in i-vectors.

We also tried *multi-head attention*, which is similar to LDE but it normalizes the frame weights to sum up to one in the time dimension, not in the GMM component dimension. It intends to find the most important frames in the sequence.

5. Back-ends

The back-ends consisted of LDA dimension reduction to 200, centering, whitening, length normalization, PLDA and score normalization. We tuned different back-ends for the SITW/VAST condition and the CMN2 condition.

5.1. SITW/VAST

HLTCOE used full-rank PLDA while CLSP-MIT used simplified PLDA (150 eigenvoices). It was trained on data originally at 16 kHz as described in Section 2.2. For HLTCOE, centering was calculated given equal weight to the SITW-dev-diar and SRE18-VAST-dev-diar sets. For CLSP-MIT, centering for SITW dev/eval was calculated on SITW-dev-diar. Meanwhile, the centering for VAST was MAP adapted from SITW-dev-diar to SRE18-dev-VAST-diar with relevance factor $r = 14$. We used diarization on the test recordings to obtain single speaker segments. We scored the enrollment segment against all the diarization segments and selected the maximum score.

We observed better alignment between SITW and VAST dev score distributions when using adaptive score normalization (S-Norm) [12]. Thus, we expected to obtain better calibration on the VAST eval using S-Norm, which was finally true. We used adaptive S-Norm with SITW-SRE18-dev-diar as cohort. HLTCOE used the 10% top cohort segments; CLSP-MIT used 500 top cohort segments for SITW eval and 120 for VAST.

5.2. CMN2

HLTCOE used the heavy-tail PLDA in [21]–no length normalization was needed. CLSP-MIT used SPLDA or discriminative PLDA (DPLDA). It was trained on SRE telephone data as described in Section 2.2. On SRE18 CMN2, we used the centering computed on the SRE18 unlabeled data. We adapted the SPLDA to the SRE18 unlabeled data in two steps. First, we adapted SPLDA using the telephone numbers in the meta-data as speaker labels. Second, we used the adapted SPLDA to apply agglomerative clustering (AHC) to the SRE18 unlabeled segments and obtain new speakers labels. Those labels were used to adapt again the PLDA. The number of speakers for AHC was tuned based on the SRE18 CMN2 dev Cprimary. The within-class and between-class covariances of the adapted model were a weighted sum of the out-of-domain $\mathbf{S}_{\text{out}}$ and in-domain $\mathbf{S}_{\text{in}}$ covariances. $\mathbf{S}_{\text{adapt}} = \alpha\mathbf{S}_{\text{in}} + (1-\alpha)\mathbf{S}_{\text{out}}$, with $\alpha = 0.3$ for HLTCOE and $\alpha = 0.6$ for CLSP-MIT. We used adaptive S-Norm using SRE18 unlabeled as cohort. HLTCOE used the top 20% cohort segments and CLSP-MIT used 400 cohort segments to compute the normalization parameters of each trial.

6. Diarization

For diarization of the video data, we used a similar setup to the Kaldi x-vector callhome diarization recipe[5], which is based on [22]. We used E-TDNN (HLTCOE) or F-TDNN (CLSP-MIT) x-vector to compute embeddings using a sliding window with 1.5 seconds length and 0.75 seconds shift. We scored all x-vectors in a given recording against each other and applied AHC on the score matrix. CLSP-MIT tuned the stopping threshold for AHC to optimize performance on the SITW eval sets. The HLTCOE team, in order to eliminate the AHC threshold, assumed that there were never more than $K = 3$ speakers in an utterance, and perform clustering K times, with $k \in \{1, 2, \ldots, K\}$ clusters each time. Then, we use all the segments from those K clustering in the final PLDA scoring [13].

7. Fusion and Calibration

Fusion and calibration was performed using linear logistic regression with the Bosaris toolkit [23]. To select the best fusion, we implemented a greedy fusion scheme. First, we calibrated all the systems and select the one with the lowest actual cost. Then, we evaluated all the two-system fusions that include that best system. Thus, we got the best two systems fusion. We fixed those two systems and then add a third system, and so on. To reduce the chances of over-fitting, we prioritized fusions with only positive weights. For VAST, we trained fusion/calibration on SITW eval-core multi. However, we observed a misalignment between the non-target score distributions of SITW and VAST dev. We tuned the weight of the VAST dev data in the centering and score-normalization to realign those distributions

[5] https://github.com/kaldi-asr/kaldi/tree/master/egs/callhome_diarization/v2

Table 1: *Results on SITW/VAST.*

System	SITW EVAL CORE		SITW EVAL CORE-MULTI		SRE18 EVAL VAST		
	EER	Min Cp	EER	Min Cp	EER	Min Cp	Act Cp
Primary	**1.53**	**0.097**	**1.82**	**0.105**	**10.18**	**0.358**	**0.431**
E-TDNN-16k (COE)	**1.99**	**0.138**	**2.26**	**0.135**	**11.11**	0.402	**0.452**
TDNN-16k	3.4	0.185	3.86	0.191	12.06	0.468	0.578
F-TDNN-16k	**1.89**	**0.124**	**2.33**	**0.135**	12.06	**0.388**	**0.474**
ResNet-LDE-16k	2.16	0.136	2.63	0.145	**10.79**	0.412	0.516
TDNN-8k	3.58	0.197	3.93	0.206	12.93	0.431	0.596
F-TDNN-8k	2.6	0.15	2.94	0.161	12.57	**0.383**	0.519
ResNet-MHAtt-8k	2.69	0.154	2.99	0.165	11.97	0.407	0.51
i-vector-8k	8.22	0.384	8.67	0.386	20.32	0.543	0.75
F-TDNN-16k w/o S-Norm	**1.61**	**0.12**	**2.01**	**0.133**	11.49	0.426	0.645

Table 2: *Results on CMN2.*

Systems	SRE18 DEV CMN2			SRE18 EVAL CMN2		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
Primary	**4.09**	**0.249**	**0.256**	4.5	0.312	0.313
Primary post-eval	4.18	0.253	0.263	**4.15**	**0.289**	**0.292**
E-TDNN-8k-HTPLDA (COE)	**4.55**	**0.298**	**0.312**	**4.95**	**0.352**	**0.354**
TDNN-8k	5.76	0.384	0.392	6.68	0.446	0.447
F-TDNN-8k	5.19	0.345	0.357	**5.14**	**0.357**	**0.359**
ResNet-MHAtt-8k-SPLDA	5.46	0.326	0.34	5.64	0.392	0.395
ResNet-MHAtt-8k-DPLDA	**5.64**	**0.319**	**0.337**	6.81	0.499	0.524
i-vector-8k	10.37	0.664	0.685	11.85	0.723	0.725

expecting to obtain a better calibration on the VAST eval, which actually worked. For CMN2, we just trained on the CMN2 dev.

The primary system for VAST fused F-TDNN 16kHz, F-TDNN 8kHz, E-TDNN 16kHz and RestNet 8kHz with multi-head attention; all using GPLDA. For CMN2, the primary system fused E-TDNN 8kHz with HTPLDA, RestNet 8kHz with multi-head attention and DPLDA and TDNN 8KHz with SPLDA. As contrastive, we submitted the best single system and the best fusions of 1, 2, 3,... systems; which we don't discuss here because of the limited space.

8. Results and discussion

8.1. Evaluation results

Unless indicated otherwise in the tables, the systems used CLSP-MIT training setup, generative GPLDA and adaptive S-Norm. ResNet systems used angular softmax objective while others used cross-entropy. Table 1 presents the results for SITW/VAST. Performance was measured by EER and minimum/actual Cprimary (Cp), which is the normalized detection cost function (DCF) with target prior $P_{\mathcal{T}} = 0.05$. We omit actual cost for SITW due to space constraints and because it is nicely calibrated and it would be redundant to minimum cost. We also omit the VAST dev because it is too small to produce reliable results. For SITW eval, which was our dev set for VAST condition, we draw several conclusions. Diarization performed well since core-multi and core results were close. Best x-vector systems were around 3 times better than i-vectors. Best systems were E/F-TDNN x-vector systems, closely followed by the ResNet with LDE pooling. These deeper architectures performed around 40% better than the shallower TDNN. As expected, 16kHz systems performed better than 8kHz systems but the latter are still competitive in terms of Cprimary. The primary fusion obtained a 25% gain w.r.t. best single system. Some of these conclusions don't apply to the VAST eval, evidencing a significant mismatch between SITW and VAST. E/F-TDNN and ResNets were still better than TDNN and i-vector but the relative difference was smaller–17% and 30% in min. Cp respectively. Also, x-vectors at 16kHz and 8kHz obtained comparable minimum Cp, though systems at 16kHz were better calibrated. Fusion gain was also smaller–8% and 5% relative for min. and act. Cp. VAST calibration was good considering the big mismatch between SITW and VAST. The technique of adding VAST dev data to the centering adaptation and score normalization tuned to align SITW and VAST dev score distributions was effective. As comparison, we include F-TDNN result without S-Norm, which yields much worse calibration. However, there were still a margin of 11-19% relative between min. and act. Cp.

Table 2 presents the results for SRE18 CMN2. Cprimary is the average of DCFs at priors 0.01 and 0.005. Here, dev and eval results were very correlated and obtained good calibration. Again, E/F-TDNN results were the best closely followed by ResNet. E/F-TDNN were 50% better than i-vector and 20% better than TDNN. For the ResNet system DPLDA was better than SPLDA in dev so this system was included in the primary fusion. However, it didn't perform well on the eval due to over-fitting. By Replacing DPLDA by SPLDA in the post-eval fusion, we obtained some improvement. Post-eval fusion improved 17% w.r.t. best single system.

Due to space constraints, we don't include the results of our contrastive submissions with progressive best fusion of 2,3,... systems. Those results showed a significant gain by fusing 2 systems, but the gain of fusing 3 or more systems was marginal.

8.2. Post-eval calibration for VAST

To improve VAST calibration, we tried to transform the SITW target and non-target score distributions overlap with the SRE18 VAST dev score distributions. To do so, we adapted the mean and variance of the SITW scores (μ_{SITW}, σ^2_{SITW}) to VAST (μ_{VAST}, σ^2_{VAST}) using *maximum a posteriori*, obtaining μ_{MAP}, σ^2_{MAP}. Next, we transform the SITW scores s_{SITW} with

$$s_{\text{MAP}} = \frac{\sigma_{\text{MAP}}}{\sigma_{\text{SITW}}}(s_{\text{SITW}} - \mu_{\text{SITW}}) + \mu_{\text{VAST}} \ . \qquad (1)$$

We applied this procedure separately to the SITW target and non-target distribution. Finally, we use the adapted SITW scores to train the calibration.

Applying this method, we obtained almost perfect calibration on VAST primary, E/F-TDNN systems, with actual Cp of 0.369, 0.409 and 0.402 respectively. For the F-TDNN with S-Norm, we obtained actual Cp=0.471, which still has some gap with the min. Cp. This indicates that combining S-Norm with this calibration method was the best option.

9. Conclusions

We analyzed the JHU-MIT systems for NIST SRE18. The best single systems were very deep x-vectors based on extended and factorized TDNN architectures. Residual networks based on 2D convolutions performed close to E/F-TDNN with the advantage of having much less parameters. Shallower TDNN and i-vectors performed significantly worse. We can say that the systems presented here show the current stat-of-the-art in speaker recognition evaluations. Primary fusions obtained improvements w.r.t. single systems, although most of the gain came from the fusion of two competitive systems. We noted significant mismatch between our SITW development set and the VAST data. We showed how to obtain good calibration using very small of VAST dev data to align SITW and VAST score distributions.

10. Acknowledgements

This work is sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

11. References

[1] G. R. Doddington, "The NIST speaker recognition evaluation - Overview, methodology, systems, results, perspective," *Speech Communication*, vol. 31, no. 2-3, pp. 225–254, jun 2000.

[2] A. F. Martin and M. Przybocki, "The NIST speaker recognition evaluations: 1996-2001," in *Proceedings of Odyssey 2001 - The Speaker and Language Recognition Workshop*. Crete, Greece: ISCA, jun 2001, pp. 225–254.

[3] M. Przybocki, A. F. Martin, and A. N. Le, "NIST Speaker Recognition Evaluations Utilizing the Mixer Corpora - 2004, 2005, 2006," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 1951–1959, sep 2007.

[4] L. Brandschain, D. Graff, C. Cieri, K. Walker, and C. Caruso, "The Mixer 6 Corpus: Resources for Cross-Channel and Text Independent Speaker Recognition," in *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC10*, Valletta, Malta, may 2010, pp. 2441–2444.

[5] J. Villalba, E. Lleida, A. Ortega, and A. Miguel, "The I3A Speaker Recognition System for NIST SRE12: Post-evaluation Analysis," in *Proceedings of the 14th Annual Conference of the International Speech Communication Association, Interspeech 2013*. Lyon, France: ISCA, aug 2013, pp. 3679 – 3683.

[6] A. F. Martin, C. S. Greenberg, V. M. Stanford, J. M. Howard, G. R. Doddington, and J. J. Godfrey, "Performance Factor Analysis for the 2012 NIST Speaker Recognition Evaluation," in *Proceedings of the 15th Annual Conference of the International Speech Communication Association, INTERSPEECH 2014*, no. September. Singapore: ISCA, sep 2014, pp. 1135–1138.

[7] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Proceedings of the 18th Annual Conference of the International Speech Communication Association, INTERSPEECH 2017*. Stockholm, Sweden: ISCA, aug 2017, pp. 999–1003.

[8] NIST Speech Group, "NIST 2018 Speaker Recognition Evaluation Plan," Tech. Rep., 2018.

[9] J. Tracey and S. Strassel, "VAST : A Corpus of Video Annotation for Speech Technologies Main corpus Sub-corpora," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaky, Japan: European Language Resources Association (ELRA), may 2018, pp. 4318–4321.

[10] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors : Robust DNN Embeddings for Speaker Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*. Alberta, Canada: IEEE, apr 2018, pp. 5329–5333.

[11] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis For Speaker Verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788 – 798, may 2011.

[12] P. Kenny, "Bayesian Speaker Verification with Heavy-Tailed Priors," in *Proceedings of Odyssey 2010 - The Speaker and Language Recognition Workshop*. Brno, Czech Republic: ISCA, jul 2010.

[13] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker Recognition for Multi-Speaker Conversations Using X-Vectors," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019*. Brighton, UK: IEEE, may 2019.

[14] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohamadi, and S. Khudanpur, "Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks," in *Proceedings of the 19th Annual Conference of the International Speech Communication Association, INTERSPEECH 2018*, Hyderabad, India, sep 2018.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," dec 2015.

[16] W. Cai, J. Chen, and M. Li, "Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System," in *Odyssey 2018 The Speaker and Language Recognition Workshop*. Les Sables d'Olonne, France: ISCA, jun 2018, pp. 74–81.

[17] B. J. Borgstrom, M. S. Brandstein, and R. B. Dunn, "Improving Statistical Model-Based Speech Enhancement with Deep Neural Networks," in *IWAENC*, 2018.

[18] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep Hypersphere Embedding for Face Recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2017-Janua. IEEE, jul 2017, pp. 6738–6746.

[19] Z. Huang, S. Wang, and K. Yu, "Angular Softmax for Short-Duration Text-independent Speaker Verification," in *Interspeech 2018*. Hyderabad, India: ISCA, sep 2018, pp. 3623–3627.

[20] W. Cai, Z. Cai, X. Zhang, X. Wang, and M. Li, "A Novel Learnable Dictionary Encoding Layer for End-to-End Language Identification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary, Canada: IEEE, apr 2018, pp. 5189–5193.

[21] A. Silnova, N. Brümmer, D. Garcia-Romero, D. Snyder, and L. Burget, "Fast Variational Bayes for Heavy-tailed PLDA Applied to i-vectors and x-vectors," in *Interspeech 2018*, Hyderabad, India, 2018, pp. 72–76.

[22] G. Sell, D. Snyder, A. Mccree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge," in *Proceedings of the 19th Annual Conference of the International Speech Communication Association, INTERSPEECH 2018*, Hyderabad, India, sep 2018, pp. 2808—-2812.

[23] N. Brummer and E. De Villiers, "The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF," in *NIST SRE11 Speaker Recognition Workshop*, Atlanta, Georgia, USA, dec 2011, pp. 1–23.