IDIAP RESEARCH REPORT

IDIAP SUBMISSION TO THE NIST SRE 2019 SPEAKER RECOGNITION EVALUATION

Seyyed Saeed Sarfjoo Srikanth Madikeri
Mahdi Hajibabaei Petr Motlicek Sébastien Marcel

Idiap-RR-15-2019

NOVEMBER 2019

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

IDIAP SUBMISSION TO THE NIST SRE 2019 SPEAKER RECOGNITION EVALUATION

Seyyed Saeed Sarfjoo, Srikanth Madikeri, Mahdi Hajibabaei, Petr Motlicek, and Sébastien Marcel

Idiap Research Institute, Martigny, Switzerland
{ssarfjoo, msrikanth, mahdi.hajibabaei, petr.motlicek, marcel}@idiap.ch

ABSTRACT

Idiap has made a submission to the conversational telephony speech (CTS) challenge of the NIST SRE 2019. The submission consists of six speaker verification (SV) systems: four extended TDNN (E-TDNN) and two TDNN x-vector systems. Employment of various training sets, loss functions, adaptation sets and extracted speech features is among the main differences of the submitted systems. Domain adaptation is represented by a supervised method (developed using a limited data) with transfer learning of the batch norm layers. Mean shift and covariance estimation of batch norm allows to map the target domain to the source domain, alleviating the problem of over-fitting on the adaptation data. The back-end of all the systems is represented by the conventional Linear Discriminant Analysis (LDA) projection followed by Probabilistic LDA (PLDA) scoring for inference. The PLDA was also adapted unsupervisedly using the unlabelled part of the NIST SRE 2018 set. In addition, training the LDA and PLDA using in-domain data was investigated. The entire system was built around the Kaldi toolkit.

1. INTRODUCTION

Our systems are developed based on the x-vector framework [1]. The back-end remains the same across all the submitted systems. Two versions of x-vector where five or ten frame-level layers are applied before the statistical pooling layer are developed [2]. Here we call these two architectures TDNN and E-TDNN, respectively. In this report, we introduce new supervised adaptation method for limited amount of in-domain data. Under this condition, instead of transfer learning of all the weights, batch norm layers will be adapted to the target domain. Two parameters of the batch norm β and γ shift the mean of the represented features and estimate the covariance of the data to map the limited target domain to the source domain. For increasing the discriminability of the extracted features, some SV systems employ additive margin softmax (AMSoftmax) [3]. Applying feature normalization [4] is investigated in one of the E-TDNN SV systems. Short-time gaussianization (STG) [5] for feature extraction is investigated in another E-TDNN SV system.

Applying AMSoftmax on E-TDNN architecture is described in Section 2. Domain adaptation using batch norm transfer learning is described in Section 3. E-TDNN SV system with feature normalization is shown in Section 4. E-TDNN SV system with STG features is given in Section 5. The results on the development and evaluation sets are provided in Section 6.

2. SV SYSTEMS WITH AMSOFTMAX

Here, two SV systems using AMSoftmax are developed. The systems are mostly based on the x-vector implementation described in [1] and [2].

The large margin softmax loss can be written as:

$$L_{LMS} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s.\psi(\theta_{y_i})}}{e^{s.\psi(\theta_{y_i})}+\sum_{j=1,j\neq i}^{C}e^{s.cos(\theta_j)}}, \quad (1)$$

where $cos(\theta_j)$ is the angle between j-th column of weights in the output layer and the input of the last layer, s is the scaling factor which causes convergence, and $\psi(\theta_{y_i})$ is an angle function which is defined as:

$$\psi(\theta_{y_i}) = cos(m_1\theta_{y_i} + m_2) - m_3, \quad (2)$$

where, m_1, m_2, and m_3 are individual coefficients for angular softmax (ASoftmax), additive angular margin softmax (ArcSoftmax) and additive margin softmax (AMSoftmax) losses, respectively [3].

Here, we performed some experiments with ArcSoftmax and AMSoftmax with different margins. For regularization, L_2 regularization was applied on each CNN layer. The x-vectors obtained for each speech utterance are centered, and projected using LDA [6]. LDA of dimension 150 was used, based on tuning the parameters on the development set. After the dimensionality reduction, the x-vector representations are length-normalized [7] and modeled by PLDA [8]. For score normalization, although adaptive s-norm [9] showed significant improvement in NIST SRE 2016 set [10], based on the result on development set of NIST SRE 2018, S-norm was used as score normalization method.

2.1. Datasets

Majority of training data is in English comprising telephone, microphone, and audio from video recordings. All wide-band

audio recordings were downsampled to 8 kHz. For training the x-vector model, Switchboard dataset (SWBD)[1], main NIST dataset (SRE)[2], and Voxceleb dataset (VCELEB)[3] were used. SWBD contains Switchboard 2 Phases 1, 2, and 3 as well as Switchboard Cellular parts 1, and 2. In total, the SWBD dataset contains about 28 K recordings from 2.6 K speakers. The SRE dataset consists of NIST SREs corpora from 2004 to 2010 along with Mixer 6, which gives in total about 63 K recordings from 4.4 K speakers. VCELEB contains data from Voxceleb 1, and 2. Both datasets consist of videos from celebrity speakers. Voxceleb 1 consists of 153'516 utterances from 1'251 speakers and Voxceleb 2 consists of 1'128'246 utterances from 6'112 speakers.

To increase the amount and diversity of the existing training data, SRE and SWBD datasets were augmented with additive noise and reverberation. For reverberation and noise, RIR, and MUSAN datasets were used, respectively. RIR is the collection of room impulse responses measured in the different room sizes. The MUSAN dataset, consists of over 900 noise samples, 42 hours of music from various genres and 60 hours of speech from twelve languages. Both MUSAN and RIR datasets are freely available[4]. The strategy for augmenting the data is similar to the ideas mentioned in an original x-vector paper [1]. In addition to clean speech samples, the augmented version of the speech samples mixed with some noise, randomly chosen from four different categories, is added to the training dataset. These noise categories contain *babble, music, noise,* and *reverb* which are speech, music, noise, and room impulse response, respectively. In the first three categories, the selected noises from MUSAN dataset are added to the original speech in different SNR levels. In the last category, the training recording is artificially reverberated via convolution with simulated RIRs.

2.2. Experimental Setup

After down-sampling the speech data to 8 kHz, 23 dimensional mel frequency cepstral coefficients (MFCCs) were extracted with 25 ms window of speech data with 10 ms frameshift. Band-pass filtering was applied between 20 to 3700 Hz. Log of energy was added to the feature vector and these features were mean-normalized over a sliding window of up to 3 seconds. Energy-based voice activity detection (VAD) was used to removing the non-speech frames. For training the x-vector, chunk size of speech frames were chosen between 200 to 400 frames. For training the model from extracted features, the Tensorflow code was applied[5]. Here, in the network architecture, instead of TDNN layers, CNN layers were applied. Because the number of parameters in TDNN architecture is smaller than E-TDNN one, we did not applied dilation in this architecture and kernel size of the first three layers was set with values of 5, 5, and 7, respectively. However for E-TDNN architecture, similar to [2], dilation was set to 2, 3, and 4 in the third, fifth, and seventh layers, respectively. For tuning the margin of AMSoftmax and ArcSoftmax, we performed some experiments with 0.1, 0.15, and 0.2 margins. Based on the initial results, 0.15 margin indicated the best performance. In extraction time, chunk size of 100 seconds (10'000 frames) with minimum size of 250 ms was used, while for longer utterances, the average x-vector from input chunks was computed.

In these experiments, as the VCELEB dataset contains more than 1.2 M utterances, we did not perform data augmentation. The x-vector system was trained on combination of VCELEB and augmented version of SWBD and SRE datasets. First, we trained the PLDA classifiers on augmented version of SRE and for adapting to the target domain, we performed PLDA adaptation using Bayesian maximum a *posteriori* (MAP) estimation on test part of evaluation set of SRE 2018. However, we realized that training the LDA and PLDA with in domain data which is augmented version of evaluation set of SRE 2018 will perform better on development set of SRE 2018. The development set of SRE 2018 was used for initial evaluations, selecting the score normalization method, and calibration.

3. DOMAIN ADAPTATION USING BATCH NORM TRANSFER LEARNING

Recently, for alleviating the language mismatch problem, several domain adaptation techniques were proposed [11, 12, 13, 14]. In face recognition field, it has been shown that high level CNN layers are potentially domain independent and can be used for extracting the embedding and modeling the target identities [15]. On the other hand, low-level CNN layers can be seen as domain-specific features and adaptation of these domain-specific units (DSUs) allows to map from the target to the source domain.

In our work, TDNN and E-TDNN architectures were used for adaptation. In these architectures, batch normalization is applied after every CNN or dense layer. This batch normalization can be defined as:

$$h(x) = \beta_i + \gamma_i . \frac{g(W_i \times x) - \mu_i}{\sigma_i}, \qquad (3)$$

where β is the batch normalization offset, γ is batch normalization scale, W is the kernel of CNN layers, g is the nonlinear function which is applied to the convolution, usually ReLU, μ and σ are the accumulated mean and standard deviation of the current batch. In the back-propagation step, two variables γ and β are updated.

[1] LDC2018E48

[2] Including LDC2009E10 and LDC2012E09

[3] http://www.robots.ox.ac.uk/ vgg/data/voxceleb

[4] http://www.openslr.org

[5] Partially the code from https://github.com/mycrazycracy/tf-kaldi-speaker was used in this implementation

Algorithm 1: Training Strategy Given a Pre-trained CNN-based Model θ, Loss Function $\mathcal{L}$ and the Number of Layers to be Adapted n_{layers}. θ_t is Split Between the CNN Kernel parameter W and the Batch Normalization Parameters Including Offset β and Scale γ.

Data: θ, $\mathcal{L}$, n_{layers}
Result: θ_t
$\theta_t = \theta[: n_{layers}]$; `// Domain Spec. Units`
$\theta_s = \theta[n_{layers} :]$; `// Domain Indep. Units`
while *has_data* **do**
  batch = get_batch() ;
  $\frac{\partial \mathcal{L}}{\partial \theta_t}$ = forward_backward(batch, θ, θ_t, $\mathcal{L}$) ;
  $\theta_t[\beta] = \theta_t[\beta] - \eta \frac{\partial \mathcal{L}}{\partial \theta_t}[\beta]$;
  $\theta_t[\gamma] = \theta_t[\gamma] - \eta \frac{\partial \mathcal{L}}{\partial \theta_t}[\gamma]$;
end

3.1. Datasets

For pre-training the TDNN and E-TDNN models, similar datasets as in the Section 2 were used. As adapting the DSUs is the supervised adaptation method, CMN2 part of the evaluation set of NIST SRE 2018 was used as development set. This set contains 188 unique speakers with 13'451 segments. For increasing the variability of the adaptation set, data augmentation was applied. The method for augmentation was similar to the Section 2, however for increasing the size of adaptation set, we did not sub-sample from the augmented data. In this condition, the adaptation set contains 67'255 segments.

3.2. Experimental Setup

The experiment setup for training the TDNN and E-TDNN models were similar to the Section 2. For adaptation, last layer of the pre-trained model was replaced by fully connected layer with output size of the number of speakers in the adaptation set. For regularization, dropout layer with 40% dropout rate was applied before the final output layer. For investigating the effect of adapting the DSUs, first we adapt all the W, β, and γ parameters. In this condition, just adapting the first layer slightly improved the performance and adapting more layers caused over-fitting on the small adaptation set[6]. In addition, adapting from the first layers indicated better performance with respect to adapting from the last layers. This observation satisfied the hypothesis that low-level CNN layers are domain-specific. For alleviating the over-fitting problem on limited adaptation data, we adapt the β, γ, and combination of β and γ parameters. Adaptation of the combination of β and γ parameters showed the best performance and adapting the first four layers showed the best result. In this condition, with mean shift and scaling the covariance, the represented features of the target domain mapped to the source domain. Based on these results, language mismatch between source and target domains is more complex to be modeled in one single input layer, however for modeling the language mismatch, deeper input layers are more informative than the final layers. In addition, mean shift and covariance estimation will help to adapt the target domain with limited amount of data.

Similar to Section 2 LDA and PLDA are trained using evaluation set of SRE 2018. The development set of SRE 2018 was used for selecting the score normalization method and calibration.

[6]The best results are reported here. Layer-by-layer adaptation results can be shared if requested

4. E-TDNN SYSTEM WITH FEATURE NORMALIZATION

As a competing system to previously described E-TDNN, we developed E-TDNN exploiting feature normalisation. Similar to the NIST SRE 2019 baseline, TDNN of Snyder *et al.* [2] was used for extracting x-vectors.

4.1. Datasets

For E-TDNN-FM, we used VoxCeleb 1 and 2 datasets [16, 17] for training. During training, each utterance was augmented, considering various samples (music, noise, babble and room impulse response). Furthermore, we did not discard any augmented sample through sub-sampling.

4.2. Experimental Setup

We normalized the length of features in penultimate layer of TDNN to 100 because it improved the validation accuracy. We experimented with two scheme for training backends. First, we trained LDA and PLDA backends with augmented x-vectors of VoxCeleb dataset and adapted the PLDA using evaluation set of NIST SRE 2018. In the other setting, we trained the LDA and PLDA using x-vectors of CMN2 part of NIST SRE 2018 evaluation set. Given that there are 188 unique speakers in CMN2 evaluation set of NIST SRE 2018, we decreased the dimensionality of the LDA and PLDA from 250 to 188 in second set of experiments.

5. E-TDNN SYSTEM WITH STG FEATURES

Another competing system considers E-TDNN architecture with STG features.

STG had been consistently shown to alleviate channel effects in i-vector based speaker verification systems. Thus, we experimented with training a E-TDNN system by applying short-term Gaussianization (STG) on 20 dimensional MFCC

features [18]. Such features were earlier used in our experiments in [19, 20].

5.1. Datasets

The following datasets were used to train the E-TDNN system: Fisher, SRE04 to 10 and SRE16 evaluation set. Only speakers with 6 or more examples were included during training. The same set was used to train LDA and PLDA models.

5.2. Experimental Setup

The SRE18 evaluation set was used to adapt the PLDA models and score normalization parameters for AS-norm.

6. EXPERIMENTS

In this section, we report our results on the CMN2 part of the development set of NIST SRE 2018 available for system optimization. In addition, we report our fusion results on the evaluation set of NIST SRE 2019. We also report the time taken to evaluate each trial on an average.

6.1. System Performance

As mentioned above, all systems are evaluated on the test set provided with NIST SRE 2018 development. The same test set is used to tune the results, tune the fusion weights and calibrate our systems. The results are presented in Table 1. TDNN-AM and E-TDNN-AM are the systems when AMSoftmax is applied on the TDNN and E-TDNN architectures. TDNN-AM-BNAD and E-TDNN-AM-BNAD are the results from the proposed batch norm adaptation on top of TDNN and E-TDNN systems, respectively. E-TDNN-FN is the E-TDNN system result with feature normalization and E-TDNN-STG is E-TDNN system result when STG features were used in training. Based on the observed results except min_C for E-TDNN-AM-BNAD, the proposed batch norm adaptation technique significantly improved the SV performance. In terms of Equal Error Rate (EER), the adaptation models relatively improved the TDNN and E-TDNN SV by 9.8 and 7.0 %, respectively. In this condition, E-TDNN-AM-BNAD showed the best performance in individual SV systems. Each SV system is calibrated before the final score fusion. Linear combination was used for fusing the scores. For evaluation set of SRE 2019, the fusion score is reported. In this set, with respect to development set of SRE 2018, we observed better EER and worse min_C performance. The reason for this observation needs more investigation.

6.2. Processing Requirements

The infrastructure used to train TDNN and E-TDNN systems contains 16 GPU GeForce GTX 1080 Ti with 11 GB memory per GPU. The probing is done on CPU, Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz with a memory of 32 GB. The extraction of TDNN and E-TDNN x-vectors for enrollment and probing is done on CPU, Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz, with a total memory of 32 GB. The execution time of TDNN x-vector extraction process in a single thread when computed only on detected speech is of 14.43 times faster than real time (FRT). For the whole recordings including silence, it would be 16.4 FRT using 1.5 GB of memory. For E-TDNN x-vector, processing in a single thread is 7 FRT using 2 GB of memory. x-vector averaging time for enrollment and scoring time is negligible with respect to the x-vector extraction time.

7. ACKNOWLEDGEMENT

This work was partially supported by (1) the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833635 (project ROXANNE: Real time network, text, and speaker analytics for combating organised crime, 2019-2022), and by (2) the SNF project, ODESSA.

8. REFERENCES

[1] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *Submitted to ICASSP*, 2018.

[2] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5796–5800.

[3] Yi Liu, Liang He, and Jia Liu, "Large margin softmax loss for speaker verification," in *Proc. INTERSPEECH*, 2019.

[4] Gautam Bhattacharya, Jahangir Alam, and Patrick Kenny, "Deep speaker recognition: Modular or monolithic?," *Proc. Interspeech 2019*, pp. 1143–1147, 2019.

[5] Bing Xiang, Upendra V Chaudhari, Jiři Navrátil, Ganesh N Ramaswamy, and Ramesh A Gopinath, "Short-time gaussianization for robust speaker verification," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2002, vol. 1, pp. I–681.

[6] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," 2011, vol. 19(4), pp. 788–798, IEEE Tran. on Audio, Speech and Language Processing.

Table 1. Results on the development set of NIST SRE 2018 and evaluation set of NIST SRE 2019 datasets for all systems presented as provided by the NIST toolkit. EER: Equal Error Rate, min_C: minimum Decision Cost Function, act_C: actual Decision Cost Function.

System	SRE18 Dev			SRE19 Eval		
	EER (%)	min_C	act_C	EER (%)	min_C	act_C
TDNN-AM	5.88	0.355	0.365	-	-	-
TDNN-AM-BNAD	5.30	0.333	0.340	-	-	-
E-TDNN-AM	5.25	0.319	0.329	-	-	-
E-TDNN-AM-BNAD	4.88	0.317	0.325	-	-	-
E-TDNN-FN	6.12	0.391	0.414	-	-	-
E-TDNN-STG	8.95	0.574	0.579	-	-	-
Fusion	4.42	0.251	0.260	4.0	0.349	0.365

[7] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," August 2011, pp. 249–252, In Proc. of Interspeech.

[8] Sergey Ioffe, "Probabilistic linear discriminant analysis," in *Computer Vision–ECCV 2006*, pp. 531–542. Springer, 2006.

[9] Douglas E Sturim and Douglas A Reynolds, "Speaker adaptive cohort selection for tnorm in text-independent speaker verification," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* IEEE, 2005, vol. 1, pp. I–741.

[10] Pavel Matejka, Ondrej Novotnỳ, Oldrich Plchot, Lukáš Burget, and JH Cernockỳ, "Analysis of score normalization in multilingual speaker recognition," in *Proceedings of Interspeech*, 2017.

[11] Wei Xia, Jing Huang, and John HL Hansen, "Cross-lingual text-independent speaker verification using unsupervised adversarial discriminative domain adaptation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5816–5820.

[12] Weiwei Lin, Man-Wai Mak, Youzhi Tu, and Jen-Tzung Chien, "Semi-supervised nuisance-attribute networks for domain adaptation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6236–6240.

[13] Gautam Bhattacharya, Joao Monteiro, Jahangir Alam, and Patrick Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6226–6230.

[14] Chunlei Zhang, Shivesh Ranjan, and John Hansen, "An analysis of transfer learning for domain mismatched text-independent speaker verification," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 181–186.

[15] Tiago de Freitas Pereira, André Anjos, and Sébastien Marcel, "Heterogeneous face recognition using domain specific units," *IEEE Transactions on Information Forensics and Security*, 2018.

[16] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[17] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[18] Jason Pelecanos and Sridha Sridharan, "Feature warping for robust speaker verification," 2001, pp. 213–218, In Proc. of Speaker Odyssey.

[19] Petr Motlicek, Subhadeep Dey, Srikanth Madikeri, and Lukas Burget, "Employment of subspace gaussian mixture models in speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4445–4449.

[20] Srikanth Madikeri, Subhadeep Dey, Marc Ferras, Petr Motlicek, and Ivan Himawan, "Idiap submission to the nist sre 2016 speaker recognition evaluation," Tech. Rep., Idiap, 2016.