INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Supervised domain adaptation for text-independent speaker verification using limited data

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

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

Abstract

To adapt the speaker verification (SV) system to a target domain with limited data, this paper investigates the transfer learning of the model pre-trained on the source domain data. To that end, layer-by-layer adaptation with transfer learning from the initial and final layers of the pre-trained model is investigated. We show that the model adapted from the initial layers outperforms the model adapted from the final layers. Based on this evidence, and inspired by the works in image recognition field, we hypothesize that low-level convolutional neural network (CNN) layers characterize domain-specific component while high-level CNN layers are domain-independent and have more discriminative power. For adapting these domain-specific components, angular margin softmax (AMSoftmax) applied on the CNN-based implementation of the x-vector architecture. In addition, to reduce the problem of over-fitting on the limited target data, transfer learning on the batch norm layers is investigated. Mean shift and covariance estimation of batch norm allows to map the represented components of the target domain to the source domain. Using TDNN and E-TDNN versions of the x-vectors as baseline models, the adapted models on the development set of NIST SRE 2018 outperformed the baselines with relative improvements of 11.0 and 13.8 %, respectively.
Index Terms: Speaker recognition, speaker verification, supervised adaptation, batch norm, transfer learning

1. Introduction

In recent years, deep neural networks (DNNs) have been successfully applied to several machine learning fields including computer vision, speech recognition, or natural language processing [1, 2, 3]. Similar to the mentioned fields, DNN-based models were investigated for text-independent SV [4, 5, 6], as well as text-dependent SV tasks [7, 8, 9]. Domain compensation is one of the recent challenges in the speaker recognition field. In the recent NIST SRE challenges, one of the main interests was a language mismatch. To alleviate the language mismatch problem, several domain adaptation techniques were recently proposed [10, 11, 12, 13, 14]. In [10], an adversarial method for unsupervised discriminative domain adaptation was proposed. For reducing the domain mismatch in i-vector and x-vector SV systems, semi-supervised nuisance attribute network (SNAN) was introduced in [11]. Instead of computing the domain variability from the dataset means, maximum mean discrepancy (MMD) was used as part of the loss function. [15], addressing the face recognition, has shown that high-level CNN layers are potentially domain-independent and can be used for extracting the embedding and modeling the target identities. On the other hand, low-level CNN layers represent domain-specific components and adaptation of these domain-specific units (DSUs) allows mapping of these components from the target to the source domain.

This paper investigates the domain adaptation problem, employing the pre-trained model on source data and adapted to the target domain using limited resources. Specifically, layer-by-layer adaptation is explored with the transfer learning from the initial and final layers of the pre-trained models. Experimental results suggest that DSUs from the initial layers were more informative for mapping the represented components from the target to the source domain. For supervised adaptation using limited amount of data in target domain, instead of applying transfer learning on all the weights, adaptation of batch norm layers to the target domain is applied. This simple yet powerful domain adaptation method showed significant improvement in image processing field [16, 15, 17]. 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. CNN-based implementation of x-vector architecture with angular margin softmax (AMSoftmax) loss is investigated to adapt DSUs. Employment of AMSoftmax loss increases the discriminability of the extracted features [18]. Two versions of x-vector implementation, with five [5] or ten [19] frame-level layers, are applied before the statistical pooling layer (denoted as TDNN and E-TDNN, respectively). An augmented version of CMN2 part of the evaluation set of NIST SRE 2018 was used as adaptation set. The adapted models on the CMN2 part of the development set of NIST SRE 2018 outperformed the non-adapted models relatively by 11.0 and 13.8 %, respectively in equal error-rate.

The rest of this paper is organized as follows: Employment of AMSoftmax on x-vector architectures is described in Section 2. Domain adaptation using batch norm transfer learning is investigated in Section 3. The experimental setup and analysis of results are given in Section 4. Finally the paper is concluded in Section 5.

2. SV systems with AMSoftmax

Here, two SV systems using AMSoftmax are developed. The systems are generally based on the x-vector implementation [5, 19].

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 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 (Arc-

http://dx.doi.org/10.21437/Interspeech.2020-2342

Softmax) and additive margin softmax (AMSoftmax) losses, respectively [18].

The experiments performed in this paper apply ArcSoftmax and AMSoftmax with different margins. Based on the initial results, AMSoftmax outperformed the ArcSoftmax, hence we report only the results from application of AMSoftmax. 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 [20]. LDA dimension was tuned on the development set. After the dimensionality reduction, the x-vector representations are length-normalized [21] and classified by PLDA [22]. For score normalization, although adaptive S-norm [23] showed significant improvement in NIST SRE 2016 [24]. However, based on the result obtained on the development set of NIST SRE 2018, S-norm was used as score normalization method.

3. Domain adaptation using batch norm transfer learning

To alleviate the over-fitting problem when applying limited adaptation data, domain adaptation experiments are performed using batch norm transfer learning. Both TDNN and E-TDNN architectures were used for adaptation, where batch normalization is applied after every CNN or dense layer. More particularly, the batch normalization can be defined as:

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

where β is the batch normalization offset, γ is batch normalization scale, W is the kernel of CNN layers, g is the non-linear 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. The E-TDNN architecture for adaptation of DSUs is shown in Figure 1.

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

With updating the β and γ parameters, the distribution of the represented features from the target domain will be mapped to the source domain.

4. Experimental setup and results

In this section, we describe the datasets and performance of the SV systems. For adaptation, CMN2 part of the evaluation set of NIST SRE 2018 was used. Performance of the adaptation on the x-vector SV systems with TDNN or E-TDNN architectures is investigated on CMN2 part of the development set of NIST SRE 2018 and evaluation set of NIST SRE 2019 datasets. In addition, we report the system fusion results on the evaluation set of NIST SRE 2019.

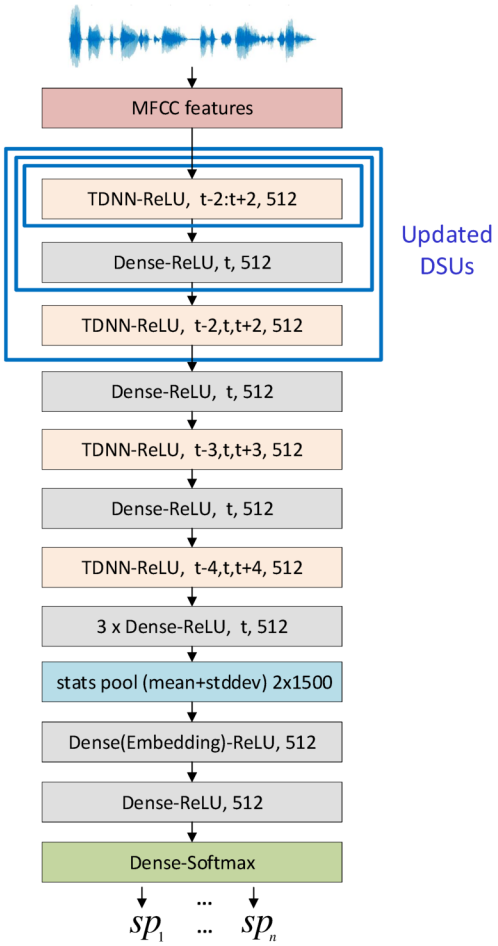

Figure 1: *E-TDNN architecture for training the embeddings extractor. DSUs are updated from the initial layers.*

4.1. Datasets

The majority of training data is in English comprising telephone, microphone, and audio from video recordings. All wideband audio recordings are downsampled to 8 kHz. For training the x-vector model, Switchboard dataset (SWBD)[1], main NIST dataset (SRE)[2], and Voxceleb dataset (VCELEB)[3] are 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

[1]LDC2018E48
[2]Including LDC2009E10 and LDC2012E09
[3]http://www.robots.ox.ac.uk/ vgg/data/voxceleb

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 are augmented with additive noise and reverberation. For reverberation and noise, RIR, and MUSAN datasets are used, respectively[4]. The strategy for augmenting the data is similar to x-vector system [5].

As adapting the DSUs requires labeled data (i.e. supervised adaptation method), CMN2 part of the evaluation set of NIST SRE 2018 is used. This set contains 188 unique speakers with 13,451 segments. For increasing the variability of this dataset, data augmentation is applied (similar to Section 2, however, for increasing the size of the data, we did not apply any sub-sampling). The final size of the adaptation set is 67,255 segments.

4.2. Experimental Setup

After down-sampling the speech data to 8 kHz, 23-dimensional MFCCs are extracted with 25 ms window from speech, with 10 ms frame-shift. Band-pass filtering is applied between 20 to 3700 Hz. Log of energy is added to the feature vector and the extracted speech features are mean-normalized over a sliding window of up to 3 seconds. Energy-based voice activity detection (VAD) is used to remove non-speech frames. For training the x-vector, a chunk size of speech frames is chosen between 200 to 400 frames. For training the model from extracted features, the Tensorflow code is applied[5]. In our network architecture, instead of TDNN layers, CNN layers are employed. As the number of parameters in TDNN architecture is smaller than E-TDNN one, we did not apply dilation and the kernel size of the first three layers is set with values of 5, 5, and 7, respectively. However for E-TDNN architecture, similar to [19], dilation is set to 2, 3, and 4 in the third, fifth, and seventh layer, respectively. For tuning the margin of AMSoftmax and ArcSoftmax, some experiments were performed with 0.1, 0.15, and 0.2 margins. Based on the preliminary results, 0.15 margin indicates the best performance. In extraction time, a chunk size of 100 seconds (10,000 frames) with a minimum size of 250 ms is used, while for longer utterances, the average x-vector from input chunks is extracted. In the PLDA back-end, the dimension of LDA is set to 150.

As the VCELEB dataset contains more than 1.2 M utterances, we did not apply data augmentation. The x-vector system is trained on the combination of VCELEB and augmented versions of SWBD and SRE datasets. First, we train the PLDA classifier on augmented version of SRE. PLDA adaptation to target domain is then performed using Bayesian maximum a *posteriori* (MAP) estimation on test part of evaluation set of SRE 2018. Nevertheless, we realized that training the LDA and PLDA with in-domain data (i.e. using augmented version of the evaluation set of SRE 2018) will provide better performance on the development set of SRE 2018. The development set of SRE 2018 is used for initial evaluations, selecting the score normalization method, and calibration.

For layer-by-layer adaptation, the last layer of the pretrained model is changed to the fully connected layer with output size of the number of speakers in the adaptation set. For regularization, dropout layer with 40% dropout rate is applied before the final output layer. The development set of SRE 2018 is used for selecting the score normalization method and calibration.

4.3. System Performance and Results

As mentioned above, all SV systems are evaluated on the CMN2 part of the development set of NIST SRE 2018. The same set is used for calibration and score fusion.[6] To investigate the effect of adapting the DSUs, first, we adapt all the W, β, and γ parameters. Under this condition, just adapting the first layer slightly improved the performance, while adapting more layers caused over-fitting on the small adaptation set. In addition, process of adaptation from the first layers outperformed the adaptation from the last layers. This observation satisfied the hypothesis that low-level CNN layers can be considered as domain-specific representations. The result of layer-by-layer adaptation from the initial, or from the final layer is shown in Table 1. Based on the observed results, initial layers are more informative for domain adaptation. However, because of using the limited adaptation set, increasing the number of adaptation layers causes over-fitting on the adaptation data.

Table 1: *Investigation of layer-by-layer adaptation from the initial and final layers of TDNN architecture on the CMN2 part of development set of NIST SRE 2018 without score normalization and in-domain PLDA. Numbers with beg or end (e.g., 2_beg and 2_end) are combination of training parameters from the first and the last 2 layers, respectively. Transfer learning is done with updating the parameters of the mentioned layers. EER: Equal Error Rate, min_C: minimum Decision Cost Function.*

Adapt Set	TDNN	
	EER (%)	min_C
baseline	6.9	**0.418**
1_beg	**6.4**	0.428
2_beg	12.9	0.750
3_beg	12.0	0.784
1_end	6.9	0.511
2_end	7.5	0.513
3_end	8.5	0.550

To alleviate the problem of over-fitting on limited adaptation data, we performed experiments for adapting the β and γ parameters individually and at the same time. Under this condition, with mean shift and scaling the covariance of the batch norm layers, the target domain mapped to the source domain. Based on these results, we hypothesize that language mismatch between source and target domains is more complex to be modeled in one single input layer, however for adapting with language mismatch, deeper input layers are more informative than the final layers. In addition, mean shift and covariance estimation of batch norm layers will help to adapt the target domain with limited amount of data. Individual adaptation of β and γ parameters are shown in Table 2. The result of combined adaptation of β and γ parameters on TDNN and E-TDNN architectures are shown in Table 3.

For TDNN SV systems, individual adaptation of β and γ parameters relatively outperformed the baselines with 7.6 and 7.0 %, respectively in equal error-rate. Adapting the β and γ parameters from the first three initial layers showed the best performance. Similar pattern was observed for E-TDNN SV systems.

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

[5]Partially the code from https://github.com/mycrazycracy/tf-kaldi-speaker was used in this implementation which internally uses Kaldi speech recognition toolkit [25].

[6]Fusion and calibration were performed using the Bosaris toolkit.

Table 2: *Individual layer-by-layer adaptation of β and γ parameters of batch norms. Results of TDNN and E-TDNN architectures on the CMN2 part of development set of NIST SRE 2018 without score normalization and in domain PLDA. Numbers with beg (e.g., 4_beg) are combination of batch norm parameters (β or γ for individual parameter adaptation) from the first 4 layers. Transfer learning was done with updating the parameters of the mentioned layers.*

Adapt Set	TDNN/E-TDNN			
	adapting β		adapting γ	
	EER (%)	min_C	EER (%)	min_C
baseline	6.97/6.91	0.418/0.47	6.97/6.91	0.421/0.47
1_beg	6.63/6.58	0.417/0.463	6.8/6.78	0.464/0.453
2_beg	6.61/6.46	0.417/0.444	6.71/6.62	0.457/0.444
3_beg	**6.44/6.42**	**0.412/0.439**	**6.48/6.45**	**0.418/0.441**
4_beg	6.45/6.44	0.414/0.441	6.73/6.60	0.450/**0.441**
5_beg	6.53/6.51	0.416/0.453	6.81/6.79	0.460/0.451
6_beg	6.63/6.57	0.426/0.473	6.86/6.83	0.482/0.471
7_beg	-/6.58	-/0.469	-/6.59	-/0.474
8_beg	-/6.60	-/0.473	-/6.56	-/0.483
9_beg	-/6.59	-/0.476	-/6.55	-/0.481
10_beg	-/6.61	-/0.484	-/6.73	-/0.491
11_beg	-/6.67	-/0.493	-/5.97	-/0.472

Adapting the β parameters of the batch norm layers, shifts the mean of the represented components of the target domain to the source domain. γ parameters adaptation, scales the covariance of the represented components of the target domain to the source domain. Adaptation of combination of these two parameters shows better performance. Here, adapting the β and γ parameters from the first four initial layers showed the best performance. The number of layers for transfer learning is relevant to the size of adaptation dataset. For TDNN and E-TDNN x-vector SV systems, the adapted models on the CMN2 part of the development set of NIST SRE 2018 outperformed the baselines with relative improvements of 11.0 and 13.8 %, respectively in equal error-rate.

In Table 4, score normalization and in-domain PLDA adaptation results are reported. Using smaller in-domain dataset is one of the main reasons for observing the current performance with respect to the top reported systems in NIST SRE 2019 challenge. 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 of the proposed batch norm adaptation on top of TDNN and E-TDNN systems, respectively. Based on the observed results except min_C for E-TDNN-AM-BNAD, the proposed batch norm adaptation technique significantly improved the performance of the SV systems. With normalized scores and in-domain PLDA, for the CMN2 part of development set of NIST SRE 2018 dataset, in terms of equal error-rate (EER), for the TDNN and E-TDNN SV systems the adaptation models improved relatively by 9.8 and 7.0 %, respectively. Similar pattern was observed for the evaluation set of NIST SRE 2019. In this set, in terms of equal error-rate, for the TDNN and E-TDNN SV systems, the adaptation models improved relatively by 9.4 and 8.9 %, respectively. Observing the similar pattern for both sets shows the generalizability of the proposed adaptation method.. E-TDNN-AM-BNAD gives the best performance across the individual SV systems. Each SV system is calibrated before the final score fusion. For score fusion, logistic regression was used. For the evaluation set of SRE 2019, the fused score is reported.

Table 3: *Combined layer-by-layer adaptation of β and γ parameters of TDNN and E-TDNN architectures on the CMN2 part of development set of NIST SRE 2018 without score normalization and in domain PLDA. Numbers with beg (e.g., 4_beg) are combination of batch norm parameters (β and γ) from the first 4 layers. Transfer learning was done with updating the parameters of the mentioned layers.*

Adapt Set	TDNN/E-TDNN	
	EER (%)	min_C
baseline	6.97/6.91	0.418/0.47
1_beg	6.63/6.43	0.408/0.473
2_beg	6.56/6.46	0.417/0.444
3_beg	6.25/6.29	0.396/0.439
4_beg	**6.18/5.95**	**0.381/0.425**
5_beg	6.54/6.16	0.404/0.437
6_beg	6.54/6.25	0.408/0.453
7_beg	-/6.22	-/0.467
8_beg	-/6.08	-/0.483
9_beg	-/6.15	-/0.487
10_beg	-/6.20	-/0.500
11_beg	-/5.97	-/0.455

Table 4: *Results on the CMN2 part of development set of NIST SRE 2018 and evaluation set of NIST SRE 2019 datasets for all systems presented with in-domain PLDA as provided by the NIST toolkit.*

System	SRE18 Dev/SRE19 Evaluation Set	
	EER (%)	min_C
TDNN-AM	5.88/5.17	0.355/0.444
TDNN-AM-BNAD	5.30/4.68	0.333/0.432
E-TDNN-AM	5.25/4.81	0.319/0.428
E-TDNN-AM-BNAD	**4.88/4.38**	**0.317/0.420**
Fusion	4.42/3.96	0.251/0.367

5. Conclusions

As a supervised model for domain adaptation with limited data, in this paper, we investigated the layer-by-layer adaptation from the initial and final layers of the pre-trained model. We observed that low-level CNN layers are more domain-specific features. In addition, for reducing the over-fitting problem, we investigated the adaptation using transfer learning of batch norm parameters. Based on the observed results, we hypothesize that language mismatch 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.

6. Acknowledgements

This work was partially supported by (1) the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833635 (ROXANNE: Real time network, text, and speaker analytics for combating organised crime), and by (2) the Swiss National Science Foundation project ODESSA (200021E-164336).

7. References

[1] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for natural language processing," *arXiv preprint*, 2016.

[4] A. Lozano-Diez, A. Silnova, P. Matejka, O. Glembek, O. Plchot, J. Pešán, L. Burget, and J. Gonzalez-Rodriguez, "Analysis and optimization of bottleneck features for speaker recognition," in *Proceedings of Odyssey*, vol. 2016, 2016, pp. 352–357.

[5] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *Submitted to ICASSP*, 2018.

[6] S. Madikeri, S. Dey, and P. Motlicek, "Analysis of language dependent front-end for speaker recognition," in *Proceedings of Interspeech 2018*, vol. 1-6, 2018, pp. 1101–1105.

[7] S. Dey, S. Madikeri, M. Ferras, and P. Motlicek, "Deep neural network based posteriors for text-dependent speaker verification," in *Proceedings of 2016 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2016)*. IEEE, Mar. 2016, pp. 5050–5054.

[8] S. Dey, P. Motlicek, S. Madikeri, and M. Ferras, "Template-matching for text-dependent speaker verification," *Speech Communication*, 2017.

[9] S. Dey, P. Motlicek, S. Madikeri, and F. Marc, "Exploiting sequence information for text-dependent speaker verification," in *Proceedings of 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017, pp. 5370–5374.

[10] W. Xia, J. Huang, and J. H. 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.

[11] W. Lin, M.-W. Mak, Y. Tu, and J.-T. 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.

[12] G. Bhattacharya, J. Monteiro, J. Alam, and P. 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.

[13] C. Zhang, S. Ranjan, and J. 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.

[14] S. Dey, S. Madikeri, and P. Motlicek, "Information theoretic clustering for unsupervised domain-adaptation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5580–5584.

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

[16] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.

[17] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7354–7362.

[18] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," in *Proc. INTERSPEECH*, 2019.

[19] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. 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.

[20] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," vol. 19(4). IEEE Tran. on Audio, Speech and Language Processing, 2011, pp. 788–798.

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

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

[23] D. E. Sturim and D. 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.*, vol. 1. IEEE, 2005, pp. I–741.

[24] P. Matejka, O. Novotnỳ, O. Plchot, L. Burget, and J. Cernockỳ, "Analysis of score normalization in multilingual speaker recognition," in *Proceedings of Interspeech*, 2017.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.