INVESTIGATING DOMAIN SENSITIVITY OF DNN EMBEDDINGS FOR SPEAKER RECOGNITION SYSTEMS

Md Hafizur Rahman, Ivan Himawan*, Sridha Sridharan, Clinton Fookes*

Speech and Audio Research Laboratory, SAIVT, Queensland University of Technology, Australia
{m20.rahman, i.himawan, s.sridharan, c.fookes}@qut.edu.au

ABSTRACT

A speaker embeddings framework achieves state-of-the-art speaker recognition performance by modeling speaker discriminant information directly using deep neural networks (DNNs). After the introduction of neural network based speaker embeddings, researchers have explored the requirements for training an effective embeddings network. However, the domain of the data used for system development should match the domain of operation for optimal performance. In this paper, we investigate the sensitivity of domain mismatch in the embeddings space. Specifically, degradation in performance is observed when back-end scoring with embeddings is performed with out-domain data. To compensate for the domain mismatch, we propose two novel deep domain adaptation techniques based on autoencoder architectures trained on embeddings in an unsupervised fashion. The results show that domain mismatch can be compensated effectively using autoencoders to adapt the out-domain data to in-domain.

Index Terms— deep domain adaptation, speaker embeddings, autoencoder, score normalization, speaker recognition

1. INTRODUCTION

Speaker recognition technology has been greatly influenced recently by the use of deep neural networks (DNNs). A notable performance gain is obtained when automatic speech recognition (ASR) DNNs are used to replace the universal background models (UBMs) for extracting sufficient statistics for i-vector computation [1, 2, 3]. Although mixtures of the GMMs are considered to be related to phonetic events, DNNs provide a more efficient way to model the acoustic contents of the speech signal by representing each phonetic event by a number of tied-triphone states referred to as *senones*. More recently, speaker embedding frameworks have emerged by combining all the necessary steps for speaker classification directly inside the DNN framework. These frameworks have shown impressive performance in speaker recognition tasks, providing sufficient training data for successful implementation. The objective of the embeddings training is to maximize the same speaker probability and minimize the between speaker probability. Thus, the network learns speaker discriminant information during training by classifying speakers.

Several embedding frameworks have been proposed for the text independent speaker recognition task, among which "d-vector" [4] and "x-vector" [5, 6, 7] systems, have proven to be very efficient. In [4], Variani *et al.* computed the average of the output activations from the last hidden layer using standard feed-forward propagation in the trained embeddings network to represent the speaker models, referred to as "d-vector". Snyder *et al.* [5, 6, 7] proposed the "x-vector" system using a feed-forward DNN, which maps the stacked input features fed to the network into speaker embeddings. This network consists of five fully connected hidden layers, a temporal pooling layer and a softmax layer. The pooling layer aggregates the average and standard deviation of the activations and pass it to the last hidden layer. The embeddings (i.e., speaker discriminative features) were later extracted from the activation of an affine layer on top of statistics pooling.

Recently, McLaren *et al.* [8] studied different aspects of modeling robust speaker embedding systems. They explored the effects of speech activity detection (SAD), data degradation of the training data by adding different levels of noise, reverberation, music and pitch. They showed that a successful and robust speaker embedding system depends on ensuring the use of a large cohort of speakers, as well as artificially degrading the training data as much as possible with different types of noise, and with a lower reverberation and SAD threshold. All of these investigations were performed using PRISM training data [9] for the DNN embeddings, and the NIST speaker recognition evaluation (SRE) data for the probabilistic linear discriminant analysis (PLDA) back-end scorer training.

The PLDA speaker recognition system is very sensitive to the training data and its performance degrades quite substantially when training the PLDA models on out-domain data. We always have to ensure sufficient training data as well as providing target domain data for a robust PLDA speaker modeling [10]. This domain sensitivity of the PLDA modeling was first introduced as a challenge in the Speaker and Language Recognition Workshop at Johns Hopkins University (JHU) in 2013 [11]. Two domains were investigated in this challenge. The source or out-domain data were collected from the LDC Switchboard corpus (SWB) telephone dataset and the target or in-domain data were collected from National Institute of Standards and Technology (NIST) telephone dataset. The findings showed that domain mismatch adds around 15-40% performance degradation in the i-vector based PLDA speaker recognition performance. Several unsupervised [12, 13, 14, 15, 16, 17, 18], and supervised [19, 20, 21, 22] i-vector based PLDA domain adaptation techniques have recently been proposed to address this issue of domain sensitivity inside the i-vector subspace.

To date, very few studies have been reported addressing the domain mismatch issues for the end-to-end speaker recognition systems. Despite the initial success of previously developed domain compensation techniques, domain mismatch issues have not been completely solved. For example, it is not clear if inter-dataset variability compensation (IDVC) [15], and domain-invariant covariance normalization (DICN) [16] can be successfully applied on the embeddings subspace. Using the dataset means to model the mismatch may not be sufficient since mismatch may also manifest in the higher-order statistics of the dataset [23]. Few recent works have ex-

*Both authors contribute equally to this paper.

plored the use of a neural network (i.e., autoencoder) to reduce inter-dataset mismatch in unsupervised settings, and reported promising results when i-vectors are used for training autoencoder [23, 24]. The maximum mean discrepancy (MMD) metric is the most commonly used method for comparing between the two domains as well as minimizing distribution shift between domains [25]. In contrast to MMD, correlation alignment (CORAL) aligns the second-order statistics between source and target distributions using a linear transformation [26]. CORAL is later extended to deep neural networks (deep CORAL) to learn a nonlinear transformation [27].

This paper studies the domain sensitivity of the DNN embeddings network and reports the performance degradation of the system resulting from the influence of the domain mismatch in the training data. This is accomplished by training the DNN embeddings with both in-domain and out-domain data, and later investigating the effects of domain variability in the embeddings space prior to the PLDA training. The paper also proposes a deep learning based solution to the domain mismatch problem.

To address the domain mismatch, we have employed autoencoders for deep domain adaptation. In the field of computer vision, deep domain adaptation is referred as a technique that employs deep networks to solve the problem of domain shift between the source and target domains. It is generally assumed that the source and target domains are similar and the knowledge transfer between the two domains can be performed in one step [28]. We make similar assumptions in our approach, and in the supervised setting, the target label is given. However, for unsupervised case, the focus is on learning domain invariant features by minimizing the domain distribution discrepancy. In our work, CORAL is used as the loss function of autoencoder trained on speaker embeddings, in contrast, [29] used CORAL directly on out-domain and in-domain speaker embeddings for unsupervised domain adaptation. We show that the autoencoder trained to learn shared representations of the source and target domain data is capable of producing domain compensated features while preserving the speaker information. In addition, we have proposed a two-stream autoencoder using separate weights to explicitly model the domain shift [30], and at the same time employing CORAL loss to minimize the discrepancy between source and target domains.

This paper is organized as follows. Section 2 describes the DNN embeddings system, autoencoder setups, and PLDA back-end setups used in this paper. Section 3 outlines the system protocol and experimental methodology. Experimental results are discussed in Section 4. Finally, Section 5 concludes the paper.

2. SYSTEM DESCRIPTION

2.1. Speaker embeddings network

We used a feedforward end-to-end DNN to embed the speaker discriminant information directly into the DNN architecture proposed by Snyder *et al.* [5, 6, 7]. This network consists of five fully connected hidden layer working on a frame level features, followed by a temporal pooling layer, two hidden layers and a softmax layer working on utterance level features. The temporal pooling layer collects the mean and standard deviations of the activations from the previous layer for all of the frames corresponding to the same session. The embeddings were extracted from the 6th hidden layer after removing the last two layers from the network. Our x-vector systems were developed based on the *nnet3* Kaldi recipe [31].

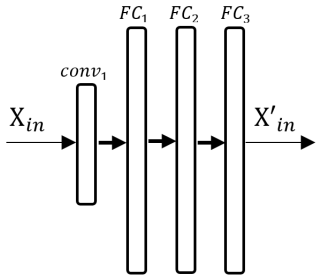

Fig. 1. Autoencoder architecture used in this paper.

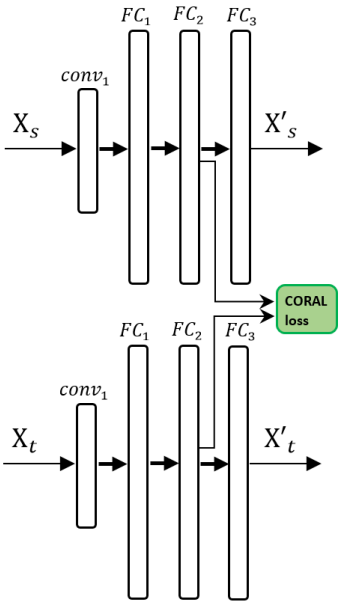

Fig. 2. Two-stream autoencoder with separate weights are trained jointly. The input data to the first layer on the top and to the first layer on the bottom are from source and target domains, respectively. CORAL loss is employed to reduce discrepancy between source and target domains.

2.2. Domain adaptation using Autoencoder

Our first proposed approach employs a basic autoencoder to learn high-level representation of embeddings in an unsupervised manner. A typical autoencoder consists of an encoder network which encodes the input, extract significant characteristics of the input, and followed by a decoder network. Let us denote the sets of embeddings $\mathbf{X}_s$ and $\mathbf{X}_t$ from source and target domains, respectively. Since an autoencoder is used to reconstruct the union of data from two different domains with the least possible amount of distortion, the learned features embedded in the latent space can represent both the source and target domain data. Thus, we concatenated the data from both source and target domain as $\mathbf{X}_{in}$ and the network was trained by minimizing the reconstruction errors, i.e., mean squared error loss: $\mathcal{L}(\mathbf{X}_{in}, \mathbf{X'}_{in}) = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{X}_{in}^{(i)} - \mathbf{X'}_{in}^{(i)}||^2$ with N the amount of training samples. The $\mathbf{X'}_{in}$ is the reconstructed version of the original input $\mathbf{X}_{in}$.

The autoencoder network contains four layers, the 1-D convolu-

tion layer $conv_1$, followed by a fully connected layer FC_1 (512 neurons), a code layer FC_2 (512 neurons), and a fully connected layer FC_3 (512 neurons). The Sigmoid activation function was employed after each layer in the network except the code layer. The domain compensated embeddings were extracted from the code layer. We did not reduce the dimension of embeddings to retain the speaker information as much as possible. Figure 1 depicts the autoencoder network used for our experiments. Tensorflow [32] is used to implement the model.

2.3. Domain Adaptation using Two-Stream Autoencoder

We addressed the impact of domain shift by making the source data distributions to be as similar as the target data distributions. In our case, we used a CORAL loss computed from the learned feature distributions between the source and target domains. Hence, we constructed a two-stream autoencoder where the weights between the two streams were not shared. The input to the first stream was input data from source domain, and the input to the second stream was data from target domain. The two-stream autoencoder was trained by minimizing the reconstructions errors from both source and target domains, and at the same time the CORAL loss,

$$\mathcal{L}(\mathbf{X}_s, \mathbf{X'}_s, \mathbf{X}_t, \mathbf{X'}_t) = \mathcal{L}(\mathbf{X}_s, \mathbf{X'}_s) + \mathcal{L}(\mathbf{X}_t, \mathbf{X'}_t) + \mathcal{L}_{CORAL} \quad (1)$$

where $\mathcal{L}(\mathbf{X}_s, \mathbf{X'}_s)$ and $\mathcal{L}(\mathbf{X}_t, \mathbf{X'}_t)$ are the reconstruction loss for source and target domains, respectively. The CORAL loss is defined as,

$$\mathcal{L}_{CORAL} = \frac{1}{4d^2}||C_s - C_t||_F^2 \quad (2)$$

where C_s and C_t denote the covariance matrices (second-order statistics) of the source and target d-dimensional features, respectively. The $||.||_F^2$ denotes the squared matrix Frobenius norm.

Figure 2 depicts the two-stream autoencoder network used for our experiments. The same network architecture (using the configuration in Figure 1) is used for both source and target streams.

2.4. PLDA Back-end Scoring

We used PLDA back-end scorer to calculate the scores between enroll and test x-vectors. The dimension and channel effects of the x-vectors were reduced using linear discriminant analysis (LDA) subspace transformation by selecting 150 eigenvectos from 512 based on highest eigen values. Later, length normalization was applied prior to the PLDA modeling. The PLDA scorings between the target and test embeddings were computed using the batch likelihood ratio [33]. For a given target sample $\mathbf{x}_{target}$ and test sample $\mathbf{x}_{test}$, the batch likelihood ratio can be calculated as follows,

$$\ln \frac{P(\mathbf{x}_{target}, \mathbf{x}_{test} \mid H_1)}{P(\mathbf{x}_{target} \mid H_0)P(\mathbf{x}_{test} \mid H_0)} \quad (3)$$

where H_1: The speakers are same, H_0: The speakers are different. We used the PLDA implementation based on [34] in Kaldi toolkit.

3. EXPERIMENTAL METHODOLOGY

3.1. Datasets

The training datasets were collected from NIST, collectively referred as SRE (in-domain), and SWB (out-domain) datasets as reported in DAC [11]. The in-domain dataset consists of 36,470 sessions gathered from NIST-2004, 2005, 2006 and 2008 SRE datasets.The out-domain dataset contains 33,039 sessions telephone data collected from Switchboard I, II phase I, II, III corpora. These training data are used for both DNN embeddings and PLDA back-end scorer training. We performed data augmentation strategy where noise and reverberation are added into the original data to increase the amount and the diversity of the existing data. The MUSAN noise corpus and simulated room impulse responses (RIR) samples are available in `http://www.openslr.org/`. A subset of 5,000 in-domain data pooled from NIST 2004, 2005, 2006 and 2008 datasets are used for score normalization (using S-Norm) [35, 36]. For PLDA adaptation, a randomly selected unsupervised 3,000 utterance are collected from the in-domain dataset. The performances were evaluated on extended *core-core* telephone-telephone condition of NIST 2010 SRE plan and the performances were measured using the equal error rate (EER).

Table 1. *Performance comparison of baseline speaker recognition systems, evaluated on NIST-2010 extended core-core condition.*

DNN Embeddings	PLDA	S-Norm	EER(%)
Out-domain	Out-domain	–	9.11
		Out-domain	12.36
		In-domain	4.88
	In-domain	–	**1.80**
		Out-domain	2.89
		In-domain	2.19
	Pooled	–	1.99
		Out-domain	3.35
		In-domain	2.20
In-domain	Out-domain	–	15.54
		Out-domain	20.59
		In-domain	6.97
	In-domain	–	1.89
		Out-domain	2.83
		In-domain	2.28
	Pooled	–	1.95
		Out-domain	3.28
		In-domain	2.17

3.2. Feature Extraction

The 23-dimensional feature-warped MFCCs with Δ and $\Delta\ \Delta$ coefficients were extracted from the 8 kHz speech signal using 25 ms frames with 10 ms frame shift. The silence frames were removed from the features using energy-based voice activity detection (VAD). For the DNN embeddings training, speakers less than 8 sessions and sessions having less than 5 seconds of conversation were removed from the training data. The embeddings features were extracted from the 6th hidden layer after removing the last two layers from the network. The domain compensated embeddings were extracted from the code layer FC_2 of the autoencoder, and for the two-stream autoencoder features were extracted from the FC_2 layer of the top network (source stream).

4. EXPERIMENTAL RESULTS

4.1. Baseline performance

This section presents the performance of DNN embeddings systems to understand the sensitivity of embeddings to a specific domain data. No adaptation techniques had been applied for this sets of experiments. Experimental results presented in Table 1 shows that

Table 2. *Performance comparison of autoencoder (AE) based domain adaptation, evaluated on NIST-2010 extended core-core condition. Out-domain data are used for training DNN embeddings and PLDA model. Unsupervised PLDA adaptation is referred as adPLDA.*

System	Backend	Score normalization	EER(%)
AE	PLDA	–	6.25
		s-norm	4.58
	adPLDA	–	3.50
		s-norm	3.36
AE + Coral	PLDA	–	6.17
		s-norm	3.40
	adPLDA	–	4.71
		s-norm	**3.20**

the out-domain PLDA performs catastrophically worse regardless of the DNN embeddings training data. However, out-domain DNN embeddings performs relatively well compared to the in-domain system. The reason behind this performance difference is that while training the DNN embeddings with out-domain data, it learns the domain specific information which produces a more efficient speaker embeddings, leading to a domain mismatch compensated PLDA speaker models. Now despite of the DNN embeddings training, the in-domain PLDA perform relatively well (1.80∼1.89%) compared to the out-domain systems, which suggests that PLDA training data is very crucial for the DNN embeddings speaker recognition system. Specifically, the target domain data should always be provided for the PLDA training for a reliable system performance.

We also investigated the performance of the PLDA training with pooled in-domain and out-domain data. One can argue that pooling PLDA training data is a relatively straight-forward domain adaptation, but this is really important for the sake of the investigation in order to understand that how much system performance varies if we add target domain data for the PLDA training. From the experimental results, it is clear that the system performance is within an acceptable range of 1.95∼1.99% compared to the in-domain PLDA performance for both in- and out-domain DNN embeddings. Therefore, it proves that a seen target domain PLDA model is a key to a successful speaker recognition system implementation. Score normalization also plays a vital role and the S-Norm training data should always match to the testing condition for improving overall system performance. For out-domain PLDA system, in-domain S-Norm yielded improvement of at least 46% relative (i.e., from EER of 9.11% to 4.88%). These results are also consistent for both in-domain and pooled PLDA systems as well.

4.2. Domain adaptation performance

Table 2 presents the performance of unsupervised domain adaptation using techniques presented in Section 2.2 and 2.3. For these experiments, we considered that we have only access to unlabeled in-domain data for adaptation. The autoencoder was trained by inputting both SWB (out-domain) and SRE (in-domain) embeddings. Both DNN embeddings and PLDA were trained on out-domain data. We also took advantage of the unsupervised PLDA adaptation [7] and in-domain mean shift using unlabeled in-domain data. Experimental results show that using a simple autoencoder (AE) for domain adaptation gains 31.4% and 58% (relative) performance improvements for PLDA and adapted PLDA (adPLDA) compared to the baseline performance, respectively. Also, an additional 6.2% (from 4.88% to 4.58%) and 31.1% (from 4.88 to 3.36%) performance improvements can be gained using a score normalization with the in-domain data for PLDA and adPLDA, respectively.

For domain mismatch compensation, we also employed two-stream autoencoder, out-domain and in-domain embeddings are inputted in parallel with CORAL loss estimated from the feature distributions from both pipelines. The motivation behind is to minimize the domain variability, while training two autoencoders jointly by minimizing the reconstruction errors and CORAL loss. Now, compared to the out-domain baseline performance, the two-stream autoencoder system (AE + Coral) yielded performance improvements of 32.3% (from 9.11% to 6.17%) with the normal PLDA system. By employing adapted PLDA this performance can be further improved to 48.2% (from 9.11% to 4.71%), compared to the out-domain baseline. Similar to the single autoencoder system, the best performance was achieved by using score normalization with in-domain data. As a result, the PLDA system obtained 30.3% (from 4.88% to 3.40%) and adPLDA system achieved 34.4% (from 4.88% to 3.20%) performance improvements (relative) compared to the out-domain baseline.

5. CONCLUSIONS

In this paper we have investigated the effects of training DNN embeddings speaker recognition system on the non-target domain data. We found that out-domain DNN embeddings training has very limited effects on the overall speaker recognition performance, as long as we provide proper training setup and sufficient training data. However, extracted embeddings for PLDA training has a crucial effect on the overall system performance. The out-domain PLDA modeling degrades the system performance substantially, although this large performance degradation can be remedied by employing in-domain score-normalization technique. We also presented two domain adaptation setups using autoencoders to compensate this mismatch prior to the PLDA training. Experimental results showed that employing simple autoencoder can suppress this domain variability from the PLDA training data. The two-stream autoencoder trained on out-domain and in-domain embeddings can successfully compensate this domain mismatch further. Also, using this two-stream autoencoder in together with unsupervised PLDA adaptation and in-domain score normalization achieved the best performance so far.

6. REFERENCES

[1] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 92–97.

[2] L. Ferrer, et al., "Study of senone-based deep neural network approaches for spoken language recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 1, pp. 105–116, 2016.

[3] M McLaren, Y. Lei, and L. Ferrer, "Advances in deep neural network approaches to speaker recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2015, pp. 4814–4818.

[4] E. Variani, et al., "Deep neural networks for small footprint text-dependent speaker verification," in *Proceedings of IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, pp. 4052–4056.

[5] D. Snyder, et al., "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proceedings of IEEE Workshop on Spoken Language Technology (SLT)*. IEEE, 2016, pp. 165–170.

[6] D. Snyder, et al., "Deep neural network embeddings for text-independent speaker verification," in *Proceedings of Interspeech*, 2017, pp. 999–1003.

[7] D. Snyder et al., "X-vectors: Robust DNN embeddings for speaker recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[8] M. McLaren et al., "How to train your speaker embeddings extractor," in *Proceedings of Odyssey The Speaker and Language Recognition Workshop*, 2018, pp. 327–334.

[9] L. Ferrer et al., "Promoting robustness for speaker modeling in the community: the PRISM evaluation set," in *Proceedings of NIST 2011 workshop*. Citeseer, 2011.

[10] Md. H. Rahman, et al., "Investigating in-domain data requirements for PLDA training," in *Proceedings of Interspeech*, 2015, pp. 2322–2326.

[11] Domain Adaptation Challenge, "Speaker and language recognition summer workshop, Tech. Rep., 2013," in *2013 speaker recognition workshop. Available online: http://www. clsp. jhu. edu/workshops/archive/ws13-summerworkshop/groups/spk-13*, 2013.

[12] J. Villalba and E. Lleida, "Unsupervised adaptation of PLDA by using variational Bayes methods," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 744–748.

[13] D. Garcia-Romero, et al., "Unsupervised domain adaptation for i-vector speaker recognition," in *Proceedings of Odyssey The Speaker and Language Recognition Workshop*, 2014, pp. 260–264.

[14] O. Glembek et al., "Domain adaptation via within-class covariance correction in i-vector based speaker recognition systems," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 4032–4036.

[15] H. Aronowitz, "Inter dataset variability compensation for speaker recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 4002–4006.

[16] Md H. Rahman et al., "Dataset-invariant covariance normalization for out-domain PLDA speaker verification," in *Proceedings of Interspeech*, September 2015, pp. 1017–1021.

[17] Md H. Rahman et al., "Domain-invariant i-vector feature extraction for PLDA speaker verification," in *Proceedings of Odyssey The Speaker and Language Recognition Workshop*, 2018, pp. 155–161.

[18] Md H. Rahman et al., "Domain mismatch modeling of out-domain i-vectors for PLDA speaker verification," in *Proceedings of Interspeech*, 2017, pp. 1581–1585.

[19] J. Villalba and E. Lleida, "Bayesian adaptation of PLDA based speaker recognition to domains with scarce development data," in *Proceedings of Odyssey The Speaker and Language Recognition Workshop*, 2012.

[20] D. Garcia-Romero and A. McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014, pp. 4047–4051.

[21] Q. Wang, et al., "Domain adaptation using maximum likelihood linear transformation for PLDA-based speaker verification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2016, pp. 5110–5114.

[22] Md H. Rahman et al., "Improving PLDA speaker verification performance using domain mismatch compensation techniques," *Computer Speech & Language*, vol. 47, no. 1, pp. 240–258, 2018.

[23] W. Lin et al., "Reducing domain mismatch by maximum mean discrepancy based autoencoders," in *in Proceedings of Odyssey The Speaker and Language Recognition Workshop*, 2018, pp. 162–167.

[24] S. Shon et al., "Autoencoder based domain adaptation for speaker recognition under insufficient channel information," in *in Proceedings of Interspeech*, 2017, pp. 1014–1018.

[25] K. M. Borgwardt et al., "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. 49–57, 2006.

[26] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," *AAAI Conference on Artificial Intelligence*, pp. 2058–2065, 2016.

[27] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *European Conference on Computer Vision*, 2016, pp. 443–450.

[28] G. Csurka, *A Comprehensive Survey on Domain Adaptation for Visual Applications*, pp. 1–35, Springer, Cham, 2017.

[29] J. Alam, G. Bhattacharya, and P. Kenny, "Speaker verification in mismatch conditions with frustratingly easy domain adaptation," in *Proceedings of Odyssey The Speaker and Language Recognition Workshop*, 2018, pp. 176–180.

[30] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[31] D. Povey et al., "The Kaldi speech recognition toolkit," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, 2011.

[32] M. Abadi et al., "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[33] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in *Proceedings of Odyssey The Speaker and Language Recognition Workshop*, 2010.

[34] S. Ioffe, "Probabilistic linear discriminant analysis," in *in Proceedings of European Conference on Computer Vision*, 2006, pp. 531–542.

[35] N. Dehak et al., "Cosine similarity scoring without score normalization techniques," in *in Proceedings of Odyssey The Speaker and Language Recognition Workshop*, 2010, pp. 71–75.

[36] D. Sturim and D. Reynolds, "Speaker adaptive cohort selection for t-norm in text-independent speaker verification," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005, pp. 741–744.