Interspeech 2018
2-6 September 2018, Hyderabad

Compensation for domain mismatch in text-independent speaker recognition

Fahimeh Bahmaninezhad, John H.L. Hansen

Center for Robust Speech Systems (CRSS), University of Texas at Dallas, Richardson, TX 75080
{fahimeh.bahmaninezhad, john.hansen}@utdallas.edu

Abstract

Domain mismatch continues to be a major research challenge for speaker recognition in naturalistic audio streams. This study presents a new technique for domain mismatch compensation within a text-independent speaker recognition scenario. The proposed method is designed for the NIST speaker recognition evaluation 2016 (SRE16) task, where speakers from training, development and evaluation data belong to different sets of languages. An i-vector/PLDA speaker recognition system is adopted for this study. To address the mismatch problem, we propose to append auxiliary features to the i-vectors. These auxiliary features are adapted representations of the i-vectors to the specific in-domain data; therefore, the new feature vector has two parts: (1) i-vectors which represent speaker identity and (2) auxiliary features which are representations of i-vectors in the in-domain data feature space (and may not contain speaker identity information). This new concatenated feature vector (we call this *a-vector*) is then post-processed with support vector discriminant analysis (SVDA) for further domain compensation. Evaluations based on the SRE16 confirm the effectiveness of the proposed technique. In terms of minimum Cprimary cost, a-vector outperforms the i-vector consistently. Moreover, comparing to previous single systems introduced for SRE16, we achieved 8.5%-18% improvements in terms of equal error rate.
Index Terms: speaker recognition, domain mismatch, auxiliary features, domain-adapted triplet loss function, a-vector.

1. Introduction

Speaker recognition is the task of recognizing whether an unknown speech segment was produced by a target speaker or not [1]. NIST has been organizing a series of speaker recognition evaluations (SRE) for many years to evaluate new advances in this area and continue to explore new challenges to address the recent concerns of automatic speaker recognition systems as well as more realistic data [2]. The most recent SRE (i.e., SRE16) was focused primarily on domain mismatch problem (i.e., train, development and evaluation data belong to separate sets of languages). In addition, some other differences compared to previous SREs were introduced in SRE16; such as, greater duration variability, providing a pool of unlabeled in-domain data, etc [2]. Interested sites world wide submitted their systems, where results confirm that there is still a wide gap to achieve effective performance for current mismatch challenges. In this study, we present our continued advancements for the NIST SRE16 and introduce new insights towards compensating for specific domain mismatch cases seen in the SRE16.

In general, most submitted systems to the challenge (as well as ongoing research after the challenge) used i-vectors [3] to compress speaker identity of given speech segments to a fixed low-dimensional representation. However, variations are introduced in the traditional steps of extracting i-vectors or calculating scores to suppress the domain mismatch. The key point here is adopting unlabeled in-domain data.

In our solution [4], we extracted i-vectors using both UBM and DNN based frameworks; the UBM/i-vector had significantly better performance, but UBM-based and DNN-based i-vectors are complimentary and their score fusion helped with the overall performance. Support vector discriminant analysis (SVDA), unlabeled probabilistic linear discriminant analysis (PLDA), mean normalization using unlabeled data are among the strategies we adopted to compensate for domain mismatch.

One group [5] used different feature sets, two classifiers and three alternate models. Their submitted system consisted of a fusion of four GMM/i-vector systems with pairwise support vector machine (SVM), two DNN/i-vector with pairwise SVM, and one GMM-SVM with Nuisance Attribute Projection (NAP). The latter system was trained on unlabeled data which was clustered. They also studied other methods for unsupervised compensation, using in-domain data for MAP adaptation of GMM models which were shown to be effective.

For another team [6], their primary submission is fusion of four different i-vector based systems. These four systems differed with respect to the feature vector which was then used for training the UBM, total variability (TV)-matrix and extracting the i-vectors. For domain mismatch compensation, they applied multiple techniques: (1) whitening and mean centralization using in-domain data (2) multi-stage PLDA adaptation which also uses clustered unlabeled in-domain data.

Another submission [7] used different features (MFCC, PLP, BNF) and classifiers (PLDA, discriminative PLDA, SVM, cosine distance, Latent Dirichlet Allocation). One new aspect in their submission was training a speaker classifier neural network for extraction of d-vectors. Interestingly, they did not attempt to assign pseudo speaker labels to the unlabeled data.

The submissions to the challenge confirm that SRE16 is a difficult task and needs further investigation. After the initial SRE16 competition, different techniques are applied to overcome the challenges introduced in SRE16. As an example, [8] applied an unsupervised Bayesian adaptation method and achieved promising results. On the other hand, [9] replaced i-vectors with two new proposed embeddings which are derived based on DNN architecture. They evaluated the performance of the embeddings on both SRE10 and SRE16 tasks. In addition, domain mismatch has been previously studied for other databases or tasks as well, including [10, 11, 12, 13, 14, 15].

Here, our goal is to employ in-domain unlabeled data to achieve further compensation of domain mismatch. After the challenge, NIST provided ground truth labels, but here we are not using them or applying any clustering to generate pseudo labels. The goal of our study here is leveraging unlabeled data to improve our system in an unsupervised manner. In addition, of

This project was funded by AFRL under contract FA8750-15-1-0205 and partially by the University of Texas at Dallas from the Distinguished University Chair in Telecommunications Engineering held by J. H. L. Hansen.

10.21437/Interspeech.2018-1446

course score fusion of multiple complimentary systems always helps. However, here we do not want to focus on the score normalization or calibration, our goal is to just focus on developing an effective single system. We propose using auxiliary and complimentary features in addition to i-vectors. These features are specifically designed to only carry directions related to the in-domain languages. For this purpose, we train a simplified version of inception-v4 network [16] and propose a new loss function (we call it *domain-adapted triplet loss*).

2. Problem Setup

2.1. Database

2.1.1. Training data

There are two training conditions defined for SRE16 task, (1) fixed: using a fixed dataset for training; (2) open: additional publicly available data are permitted to be used. Our focus is on the fixed training condition, which includes data provided from Call My Net corpus, previous Mixer/SRE data, both landline and cellular Switchboard and Fisher [2]. Here, we did not use Fisher data and Call My Net corpus for the training. Therefore, in our system, the total number of speakers and segments used for training UBM and TV-matrix are 5756 and 57273 respectively. At the back-end, we also did not use any of the Switchboard data, which leads to a total of 3794 speakers and 36410 segments for training LDA/SVDA/PLDA.

2.1.2. Development and evaluation data

Data assigned to the development and evaluation sets were collected from the Call My Net corpus. Data was collected outside of North America and consists of two subsets: (1) *Major*: contains Tagalog and Cantonese languages, (2) *Minor*: contains Cebuano and Mandarin languages. Development data includes data from both minor and major language sets; evaluation data only contains data from the major set [2].

Development data includes labeled and unlabeled sets. The labeled set is only from minor languages; 10 speakers talking Cebuano and 10 speakers talking Mandarin, with each possessing 10 segments. The unlabeled one has 2272 and 200 calls from major and minor languages, respectively (They do not have speaker id, language, gender, etc information) [2].

Overall, the total number of speakers/segments in enrollment set for development and evaluation are 80/120, and 802/1202, respectively. In addition, number of target/non-target trials for development and evaluation are 4828/19312 and 1986729/1949666, respectively. Throughout the paper we refer to the development as DEV and evaluation as EVAL.

2.2. Evaluation Metric

For SRE16, NIST provided a scoring software to the participating sites; it calculates the equal error rate (EER), minimum primary cost (min-Cprimary), and actual primary cost. In addition, the software reports both equalized (false alarm and false reject counts were equalized over various partitions) and unequalized scores. Details on these costs and their calculation are provided in [2, 17].

3. i-vector/PLDA speaker recognition

Speaker recognition is the task of recognizing whether a target speaker is talking in a given speech segment or not. UBM/i-vector with PLDA scoring is the state-of-the-art speaker recognition. However, variations of this system had also successfully applied to different problems and tasks in speaker recognition. These variations include, DNN/i-vector [18], cosine distance [19] or pairwise discriminant analysis [5] scoring with i-vector, and etc. Here, as we use UBM/i-vector with PLDA scoring for development of our system, a brief description on the system is presented in this section.

Generally, the front-end of the UBM/i-vector system starts with extracting Mel-frequency cepstral coefficients (MFCCs) as the input feature vector. Next, non-speech segments of the speech are removed with voice activity detection (VAD), which we use energy-based VAD. Next, UBM and total variability (TV)-matrix are trained and used for extraction of i-vectors. At the back-end level, i-vectors are typically post-processed with LDA and length normalization [20], PLDA finally calculates the likelihood scores.

The baseline we used here is based on CRSS best single system submitted to the NIST SRE16 [4]. We did not incorporate any of the development data or any part of the Call My Net corpus at the front-end level; we mainly focused on the mismatch compensation at the back-end level. One of the main modules in our system that had a significant role in the success of our submissions was SVDA, which we also used here. We provided a brief description of that in the next subsection; more details can be found in [21].

3.1. Support vector discriminant analysis (SVDA)

Discriminant analysis via support vectors (SVDA) is a variation of LDA; however within class and between class covariance matrices are calculated in different ways. For both LDA and SVDA, the following $\hat{A}$ matrix is used for optimizing the class separation criterion [22],

$$\hat{A} = \underset{A^T S_w A = I}{\operatorname{argmax}} \, [tr(A^T S_b A)], \tag{1}$$

where S_w and S_b are within and between class covariance matrices. For SVDA, between class covariance matrix is defined,

$$S_b = \sum_{1 \le c_1 \le c_2 \le C} w_{c_1 c_2} w_{c_1 c_2}^T, \tag{2}$$

and the within class covariance matrix is

$$S_w = \sum_{c=1}^{C} \sum_{i \in \hat{I}_c} (\hat{x}_i - \hat{\mu}_c)(\hat{x}_i - \hat{\mu}_c)^T. \tag{3}$$

where the optimal direction to classify classes c_1 and c_2 with a linear SVM is $w_{c_1 c_2}$; only support vectors of the two classes are used for calculation of $w_{c_1 c_2}$, instead of all samples in both classes. $\hat{X} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_{\hat{N}}]$ contains all support vectors and $\hat{N}$ is their total number. Indexes of all support vectors in class c and their mean are shown with $\hat{I}_c$ and $\hat{\mu}_c$, respectively. Finally, optimized $\hat{A}$ includes the k eigenvectors related to the k largest eigenvalues of $S_w^{-1} S_b$

Similar to our submission to the challenge, here we also use one-versus-rest strategy for training the linear SVM used in the SVDA optimization. We always incorporate the unlabeled data into the rest class, which means we do not need to cluster unlabeled data and generate pseudo labels.

4. Domain mismatch compensation

For NIST SRE16 challenge, CRSS had 4 baseline systems (2 UBM/i-vector and 2 DNN/i-vector) and then developed 11 single systems based on that with different strategies to address

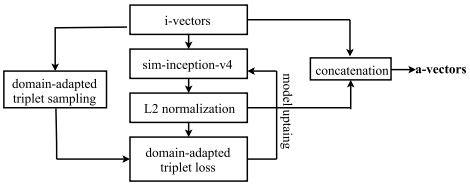

Figure 1: *Overview of the system designed for generating auxiliary domain-adapted features and a-vectors.*

the domain mismatch. Details of these systems are provided in [4]. Our best single system used a UBM/i-vector speaker representation that was post-processed with SVDA, LDA and final scores were calculated with PLDA. SVDA was able to use unlabeled in-domain data without any pseudo labels. Based on our experiments and other participating sites, leveraging unlabeled data for the purpose of adaptation or normalization was the key point to achieve a good performance.

In this section, we propose a new method for domain mismatch compensation and our work has been inspired by [23]. The focus of [23] is on speech recognition and authors propose to incorporate i-vectors as well for the input of DNN to provide speaker, channel and background normalization, and achieved a significant reduction in word error rate. Here, we propose new auxiliary features to be concatenated with the i-vectors. These features are domain-adapted representations of i-vectors and we derived them based on a convolutional neural network (CNN) and a new proposed loss function (which is a variation of triplet loss function and we call it *domain-adapted triplet loss*). In the rest of the paper, the concatenation of i-vectors and the auxiliary features are referred to as *a-vectors*. i-vectors represent speaker-dependent information while auxiliary features are domain adapted representations which are used for the purpose of domain normalization. a-vectors are post-processed with SVDA/LDA and likelihoods are calculated by PLDA similar to our best single system which is used here as the baseline. Details on the network architecture and the proposed loss function are provided in the following subsections.

4.1. Convolutional neural network a-vector representation

4.1.1. Overview

The proposed system for extracting auxiliary features is a simplified version of inception-v4 [16] and we call that sim-inception-v4. Our network takes i-vectors as the input and generates the auxiliary features that minimize the loss function introduced in 4.1.3. These auxiliary features are next concatenated with the i-vectors and created the a-vectors.

The overall system representation is shown in Fig. 1. The network architecture is described in Sec. 4.1.2 and the loss function is defined in Sec. 4.1.3.

4.1.2. Network architecture

The network is illustrated in Fig. 2. It has the stem part, inception-A and reduction-A part of the original inception-v4 network [16] (because of the limitations of our GPU we restricted the network layers). Details of the network is exactly the same as the inception-v4; however, tensors here are 1-D therefore the weight shapes of the convolution neural network are also changed to 1-D. The filter size on the remaining dimension is set to the exact values of the inception-v4. Please refer to [16] for more details of the system.

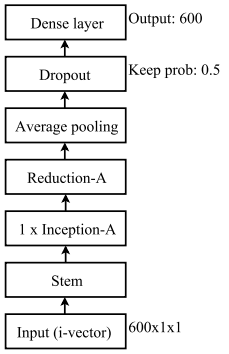

Figure 2: *simplified inception-v4 (sim-inception-v4) used for generating domain-adapted i-vector. Please refer to [16] for details on the Stem, Inception-A and Reduction-A.*

4.1.3. Proposed domain-adapted triplet loss function

The loss function proposed here is inspired by the triplet loss function. Triplet loss function was originally developed for FaceNet [24] and is also successfully applied to speaker recognition [25]. In [25] an end-to-end speaker recognition system is developed to estimate a new embedding as a replacement for the i-vector, and triplet loss is applied to make sure that the embedding carries the speaker-related information. Here, we present a domain-adapted triplet loss which maps the inputs of the network to the in-domain feature space.

As Fig.1 shows, first i-vectors are sampled into triplet sets. In the original triplet sampling, for an anchor feature vector one positive and one negative feature vectors are sampled; the positive one has the same speaker identity as the anchor one and the negative one has a different identity. Different strategies can be adopted for the selection of triplets [24, 25].

Domain-adapted triplet loss in contrast has a different meaning for the positive and negative samples. Here, for each anchor feature vector, the positive samples are in-domain unlabeled vectors (both from minor and major languages) and the negative samples are out-domain vectors which are chosen from previous years SRE data subset.

The loss function used for the training of the network minimizes the distance between the anchor and positive samples and maximizes the distance between the anchor and negative samples. It is clear that loss function applies to the output of the sim-inception-v4 which is our output auxiliary feature vector.

If we represent anchor, positive and negative i-vectors with x^a, x^p and x^n respectively and define $f(x)$ as the output auxiliary feature vector, then the network training process makes the $f(x)$ to satisfy the following relation:

$$||f(x_i^a) - f(x_i^p)||_2^2 + \alpha < ||f(x_i^a) - f(x_i^n)||_2^2, \quad \forall x_i^a, x_i^p, x_i^n \in T \tag{4}$$

where T contains all possible triplets (x^a, x^p, x^n), and α is a margin enforced between negative and positive pairs (we set $\alpha = 0.2$ in our experiments). Therefore, the loss function is defined as:

$$loss = \Sigma_{i \in T} max(0, \Delta_i). \tag{5}$$

where Δ is defined as:

$$\Delta_i = ||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha. \tag{6}$$

Generally, T contains all possible triplets, but this set will be huge if we consider all combinations and makes the convergence slower [24]; therefore, we selected a smaller subset of that in the experiments. We chose 5 random samples from the in-domain data as the positive i-vectors and 5 random i-vectors from the previous SREs data as the negative ones.

The sampling for the domain-adapted triplet loss moves all auxiliary features toward the in-domain features and far from out-domain auxiliary features, in contrast to the original triplet loss which makes the same speaker embeddings closer and different speaker embeddings farther.

5. Experiments

5.1. Experimental conditions

5.1.1. UBM/i-vector with PLDA scoring

60-D MFCC features within a 25-ms window with 10-ms skip rate are extracted first. Next, non-speech frames are removed with energy based VAD. A 2048-mixture full covariance UBM and TV-matrix are trained using parts of fixed training data of SRE16 (i.e., SRE2004, 2005, 2006, 2008 and Switchboard II phase 2,3 and Switchboard Cellular Part1 and Part2). Extracted i-vectors are centralized with global mean calculated from major and minor in-domain unlabeled data, and then they are length normalized. Now, i-vectors are concatenated with auxiliary features (output of sim-inception-v4). Output of the system is 600-D which is reduced to 150-D by PCA. The 750-D a-vectors are then fed into SVDA and their dimension reduced to 500, next LDA reduces the dimension to 400. For training LDA and PLDA only previous years SRE data is used (SWD data is not used at the back-end at all). For SVDA, in addition to the SREs data, unlabeled in-domain data is also used; which is added to the rest class while training the SVM.

5.1.2. sim-inception-v4

In our experiments, for each epoch we randomly choose 500 speakers. All utterances of these 500 speakers are selected as anchors, among previous years SRE data 5 utterances are chosen randomly as negative samples and 5 utterances from in-domain data are chosen as positive samples. Learning rate starts with 1e-2 and after 50 epochs is set to 1e-3 and after 200 iterations is 1e-4. The maximum number of epochs is 1000. RMSprop optimizer is also used for the learning process.

5.2. Experimental results

This section presents experimental results comparing 4 different systems: (1) i-vector + LDA, (2) a-vector + LDA, (3) i-vector + SVDA, (4) a-vector + SVDA. In the tables i-vector and a-vector are referred to as ivec and avec for simplicity.

Table 1 and 3 summarize EERs for the DEV and EVAL respectively, results are reported for each language as well as on the pooled data. Table 2 and 4 also represent min-Cprimary for DEV and EVAL sets, respectively. For all cases, the SVDA-based systems perform better than the LDA-based ones. In table 2, ivec+SVDA has 3%/6% relative improvement over ivec+LDA; for avec-based one also SVDA has 4%/6% improvement over LDA. In table 4, ivec+SVDA achieved better performance over ivec+LDA with 12%/14% rate; and for avec one also SVDA achieved 13%/14% improvement over avec+LDA. Improvements for EVAL data is more significant comparing to the DEV data. Comparing a-vector against i-vector in table 2, the a-vector one achieved 0.7%/0.6% and 2%/0.3% relative improvements for LDA and SVDA based systems. And for the EVAL data also has a similar range of improvements.

Table 1: *EER(%) equalized/unequalized scores on DEV*

System	Cebuano	Mandarin	Pool
ivec + LDA	21.42 / 21.78	9.14 / 9.75	15.59 / 16.08
avec + LDA	21.09 / 21.66	9.02 / 9.66	15.93 / 16.28
ivec + SVDA	20.47 / 21.66	8.14 / 8.76	15.58 / 15.95
avec + SVDA	20.69 / 21.70	8.31 / 8.88	15.35 / 15.91

Table 2: *min-Cprimary equalized/unequalized scored on DEV*

System	Cebuano	Mandarin	Pool
ivec + LDA	0.9 / 0.841	0.488 / 0.481	0.701 / 0.671
avec + LDA	0.894 / 0.839	0.471 / 0.475	0.696 / 0.667
ivec + SVDA	0.877 / 0.799	0.464 / 0.453	0.679 / 0.629
avec + SVDA	0.868 / 0.797	0.462 / 0.452	0.668 / 0.627

Table 3: *EER(%) equalized/unequalized scores on EVAL*

System	Tagalog	Cantonese	Pool
ivec + LDA	17.08 / 17.02	7.65 / 8.46	12.42 / 12.68
avec + LDA	17.21 / 17.05	7.48 / 8.25	12.41 / 12.6
ivec + SVDA	15.20 / 15.23	6.05 / 6.88	10.66 / 10.95
avec + SVDA	15.27 / 15.27	6.01 / 6.88	10.7 / 11.04

Table 4: *min-Cprimary equalized/unequalized scores on EVAL*

System	Tagalog	Cantonese	Pool
ivec + LDA	0.902 / 0.906	0.606 / 0.617	0.797 / 0.806
avec + LDA	0.902 / 0.905	0.59 / 0.607	0.791 / 0.8
ivec + SVDA	0.829 / 0.818	0.53 / 0.55	0.698 / 0.697
avec + SVDA	0.828 / 0.815	0.527 / 0.55	0.689 / 0.691

The results show that, SVDA consistently outperforms LDA, and improvements are more significant for min-Cprimary. Comparing i-vector and a-vector, in terms of min-Cprimary there is always a marginal improvement with a-vectors. However, in terms of EER improvements are not consistent.

The results show that the proposed a-vector is a promising representation; however, we think that if in each iteration we present a better selection of triplet sets, clear and consistent improvement might achieve.

Comparing our proposed system against those **single systems** (systems with no score fusion) introduced in [8, 9], we achieved 8.5% and 18% improvements respectively in terms of EER (their best performing single systems have been compared against here); and in terms of min-Cprimary a-vector is competitive with those single systems (a-vector achieved 0.689 and for those systems min-Cprimary are 0.686 and 0.689 respectively).

6. Conclusions

This study has presented a new method for compensation of the domain mismatch problem in SRE16. The proposed solution was based on concatenation of domain-adapted auxiliary features and the original i-vectors to normalize for specific language-dependent directions. For this purpose, we modeled a simplified version of the inception-v4 network to map i-vectors to these new auxiliary features. During the training process, we also proposed a new loss function called domain-adapted triplet loss function. Evaluations were based on SRE16 data, with reported EERs and min-Cprimary costs on DEV and EVAL sets confirming that the proposed method is promising in effectively addressing mismatch.

7. References

[1] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[2] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. Greenberg, E. S. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2016 NIST speaker recognition evaluation," *ISCA INTERSPEECH*, pp. 1353–1357, 2017.

[3] K. A. Lee, V. Hautamäki, T. Kinnunen, A. Larcher, C. Zhang, A. Nautsch, T. Stafylakis, G. Liu, M. Rouvier, W. Rao *et al.*, "The I4U mega fusion and collaboration for NIST speaker recognition evaluation 2016," *ISCA INTERSPEECH*, pp. 1328–1332, 2017.

[4] C. Zhang, F. Bahmaninezhad, S. Ranjan, C. Yu, N. Shokouhi, and J. H. L. Hansen, "UTD-CRSS systems for 2016 NIST speaker recognition evaluation," *ISCA INTERSPEECH*, pp. 1343–1347, 2017.

[5] V. C. Colibro, Daniele, E. Dalmasso, K. Farrell, C. S. Karvitsky, Gennady, and P. Laface, "Nuance-politecnico di torinos 2016 NIST speaker recognition evaluation system," *ISCA INTERSPEECH*, pp. 1338–1342, 2017.

[6] P. A. Torres-Carrasquillo, F. Richardson, S. Nercessian, D. Sturim, W. Campbell, Y. Gwon, S. Vattam, N. Dehak, H. Mallidi, P. S. Nidadavolu *et al.*, "The MIT-LL, JHU and LRDE NIST 2016 speaker recognition evaluation system," *ISCA INTERSPEECH*, pp. 1333–1337, 2017.

[7] O. Plchot, P. Matejka, A. Silnova, O. Novotnỳ, M. Diez, J. Rohdin, O. Glembek, N. Brümmer, A. Swart, J. Jorrın-Prieto *et al.*, "Analysis and description of ABC submission to NIST SRE 2016," *ISCA INTERSPEECH*, pp. 1348–1352, 2017.

[8] B. J. Borgstrom, D. A. Reynolds, E. Singer, and O. Sadjadi, "Improving the effectiveness of speaker verification domain adaptation with inadequate in-domain data," MIT Lincoln Laboratory Lexington United States, Tech. Rep., 2017.

[9] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *ISCA INTERSPEECH*, pp. 999–1003, 2017.

[10] A. Misra and J. H. L. Hansen, "Modelling and compensation for language mismatch in speaker verification," *Speech Communication*, vol. 96, pp. 58–66, 2018.

[11] D. Garcia-Romero, A. McCree, S. Shum, N. Brummer, and C. Vaquero, "Unsupervised domain adaptation for i-vector speaker recognition," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2014.

[12] A. Misra and J. H. L. Hansen, "Spoken language mismatch in speaker verification: An investigation with NIST-SRE and CRSS Bi-Ling corpora," in *IEEE SLT*, 2014, pp. 372–377.

[13] S. H. Shum, D. A. Reynolds, D. Garcia-Romero, and A. McCree, "Unsupervised clustering approaches for domain adaptation in speaker recognition systems," 2014.

[14] S. Shon, S. Mun, W. Kim, and H. Ko, "Autoencoder based domain adaptation for speaker recognition under insufficient channel information," *arXiv preprint arXiv:1708.01227*, 2017.

[15] F. Bahmaninezhad and J. H. L. Hansen, "Generalized discriminant analysis (GDA) for improved i-vector based speaker recognition." in *ISCA INTERSPEECH*, 2016, pp. 3643–3647.

[16] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[17] "NIST 2016 speaker recognition evaluation plan," https://www.nist.gov/sites/default/files/documents/itl/iad/mig/SRE16_Eval_Plan_V1-0.pdf, 2016.

[18] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[19] N. Dehak, P. J. 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, 2011.

[20] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in *ISCA INTERSPEECH*, 2011, pp. 249–252.

[21] F. Bahmaninezhad and J. H. L. Hansen, "i-vector/PLDA speaker recognition using support vectors with discriminant analysis," in *IEEE ICASSP*, 2017, pp. 5410–5414.

[22] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[23] A. Senior and I. Lopez-Moreno, "Improving DNN speaker independence with i-vector inputs," in *IEEE ICASSP*, 2014, pp. 225–229.

[24] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[25] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *ISCA INTERSPEECH*, 2017.