*INTERSPEECH 2017*
August 20–24, 2017, Stockholm, Sweden

# Speaker Verification Under Adverse Conditions Using I-vector Adaptation and Neural Networks

*Jahangir Alam*[1], *Patrick Kenny*[1], *Gautam Bhattacharya*[1], *Marcel Kockmann*[2]

[1]Computer Research Institute of Montreal (CRIM)
[2]VoiceTrust, Germany
{jahangir.alam, patrick.kenny, gautam.bhattacharya}@crim.ca

## Abstract

The main challenges introduced in the 2016 NIST speaker recognition evaluation (SRE16) are domain mismatch between training and evaluation data, duration variability in test recordings and unlabeled in-domain training data. This paper outlines the systems developed at CRIM for SRE16. To tackle the domain mismatch problem, we apply minimum divergence training to adapt a conventional i-vector extractor to the task domain. Specifically, we take an out-of-domain trained i-vector extractor as an initialization and perform few iterations of minimum divergence training on the unlabeled data provided. Next, we non-linearly transform the adapted i-vectors by learning a speaker classifier neural network. Speaker features extracted from this network have been shown to be more robust than i-vectors under domain mismatch conditions with a reduction in equal error rates of 2-3% absolute. Finally, we propose a new Beta-Bernoulli backend that models the features supplied by the speaker classifier network. Our best single system is the speaker classifier network - Beta-Bernoulli backend combination. Overall system performance was very satisfactory for the fixed condition task. With our submitted fused system we achieve an equal error rate of 9.89%.

**Index Terms**: Speaker verification, i-vector, PLDA, SRE 2016, Beta-Bernoulli backend, speaker classifier network

## 1. Introduction

Given two recordings of speech, each assumed to have been uttered by a single speaker, the goal of speaker detection is to determine whether both speech utterances are uttered by the same speaker or by two different speakers. This is the core task in NIST's speaker recognition evaluations (SREs). Like previous SREs, the 2016 speaker recognition evaluation (SRE16) focuses in telephone speech recorded over different types of handset but several challenges have been introduced in this evaluation. One of the major challenges is the domain mismatch between the labeled training and the evaluation data. This is because most of the labeled training data were spoken in English and the evaluation data is in oriental languages such as Mandarin, Cantonese, Cebuano and Tagalog. By evaluation data we mean the unlabeled in-domain training and enrollment/test data of SRE16. Another challenge is the introduction of more duration variability in the test data than previous SREs. The other challenges are the unlabeled in-domain training data and imbalanced single- and multi-enrollment trials [1].

In this paper, we describe the techniques adopted by the CRIM team to tackle the above-mentioned challenges of SRE16 and build speaker verification systems that are robust to adverse conditions. Our submitted fused system [2] for SRE16 is the equal weighted summation of several sub-systems.

## 2. Preparation of Training Data

We build our speaker recognition systems for fixed training condition only. For that we use data from SREs 2004-2008 and Switchboard corpus as labeled training (or background) data. All training (labeled + unlabeled) data are partitioned into four subsets and are described briefly in Table 1.

Table 1: *Partition of in-domain and out-of-domain training data used for fixed condition of the 2016 NIST speaker recognition evaluation.*

| Name | Description |
|---|---|
| Oriental background data (OBD) | This set includes recordings from Mandarin, Chinese, and Tagalog from NIST 2004-2008 SREs. No. of utterances is 2941. |
| Primary background data (PBD) | This set includes all recordings from NIST 2004-2008 SREs excluding the Mandarin, Chinese, and Tagalog plus the Switchboard corpus. No. of utterances in this set is 53228. |
| SRE16UL | In-domain unlabeled training data (2472 utterances) from SRE16 corpus. |
| Oriental data (OD) | Combination of OBD and SRE16UL. |

## 3. Processing of Frontends

Processing of front-ends include removing non-speech frames using voice activity detection and acoustic features extraction.

### 3.1. Voice Activity Detection

In order to remove non-speech frames, we use a Gaussian mixture model-based unsupervised voice activity detector (VAD) described in [3]. This VAD is conceptually similar to the VQ-based self-adaptive VAD proposed in [4]. In VQ-based VAD speech and non-speech models are estimated using k-means (with k = 16) clustering whereas in [3] they are trained using 16- component GMMs with diagonal covariance matrices and *k*-means clustering is used just for initialization.

### 3.2. Features Extraction

We extracted three types of acoustic features namely Mel-frequency cepstral coefficients (MFCC), linear frequency cepstral coefficients (LFCC), and linear prediction cepstral coefficients (LPCC). The analysis frame length is 25ms with a frame shift of 10ms. All features are of 60-dimensional



http://dx.doi.org/10.21437/Interspeech.2017-1240

including the delta and double delta coefficients. After removing non-speech frames features are normalized using a short-time mean and variance normalization technique over a window of 3s.

# 4. Extraction of Domain-adapted i-vectors

Speaker feature representation is based on i-vector [5].

### 4.1. Training of Universal Background Model (UBM)

For all systems based on the MFCC, LFCC and LPCC front-ends a gender-independent diagonal covariance UBM with 2048 Gaussians is used. The UBM is first trained on the PBD (primary background data) features and then iteratively (with 5 iterations) adapted it (mean only) to the OD (oriental data) features using the relevance MAP with a relevance factor of 2. Similarly, we train a gender-independent full covariance UBM composed of 2048 Gaussians for two of our systems based on MFCC features.

### 4.2. Training of I-vector Extractor

For each system, we train a 600-dimensional gender-independent i-vector extractor using the sufficient statistics generated from all of the out-of-domain primary background data. We refer this extractor as the primary i-vector extractor. Using this primary i-vector extractor as an initialization and after performing several iterations of minimum divergence training [6] on the sufficient statistics generated from all of the in-domain oriental data we obtain a domain-adapted i-vector extractor, denoted here as the oriental i-vector extractor.

### 4.3. Extraction and Post-processing of I-vectors

We extract 600-dimensional adapted i-vectors from all the training, development and evaluation data using the oriental i-vector extractor. After that nuisance attribute projection (NAP) is applied on the top of all i-vectors. The classes for NAP are selected by splitting oriental data (OD) into 8 subsets based on gender - language (for oriental background data) and major-minor (for SRE16 unlabeled data) information. The NAP projected i-vector are then length normalized [14] to approximately Gaussianize their distributions.

### 4.4. Backends for Scoring the I-vectors

For scoring the i-vector - based systems we employ cosine distance and probabilistic linear discriminant analysis (PLDA) backends. With PLDA backend we develop four systems based on three different frontends (MFCC, LFCC, and LPCC), diagonal and full covariance UBMs. These systems are denoted as MFCC-PLDA, LFCC-PLDA, LPCC-PLDA and MFCC-FC-PLDA. Among these MFCC-FC-PLDA uses full covariance UBM.

For training gender-independent PLDA model NAP projected and length normalized i-vectors (593-dimensional) from the labeled background set (primary background data + oriental background data) are used. Speaker space dimension is fixed to 200. No LDA (linear discriminant analysis) and WCCN (within class covariance normalization) is applied.

Similarly, using cosine distance backend we implement four systems denoted as MFCC-CD, LFCC-CD, LPCC-CD and MFCC-FC-CD. The MFCC-FC-CD utilizes full covariance UBM.

Note that the full covariance UBM-based systems MFCC-FC-PLDA and MFCC-FC-CD are developed after the evaluation. So, our submitted fused system to the SRE16 does not include these two systems.

### 4.5. Extraction and Post-processing of Primary I-vectors

In order to demonstrate the effectiveness of minimum divergence training-based domain adaptation we also extract i-vectors using the primary i-vector extractor (PIVE) with MFCC frontend and adapted UBM. The primary i-vectors are length normalized after being projected by NAP. PLDA is used as backend. We denote this system as MFCC-PIVE-PLDA.

# 5. Speaker Representations using Neural Networks

Speaker representations are normally based on i-vectors [5, 6]. 1. In [7, 8, 9, 10, 11] neural networks-based automatic speech recognition acoustic models have been employed successfully for improving these representations. Recently, convolutional neural networks with network in network (NIN) nonlinearity have been proposed for extracting speaker discriminant features [12]. In [13], we proposed a speaker classifier network (SCN) where we used a feedforward neural network to learn mapping between i-vectors and speaker labels. Projecting the i-vectors into a higher dimension space significantly improves speaker discriminant properties of the resulting features [13].

In this work, we employ the SCN to non-linearly transform the adapted i-vectors extracted in section 4 and to provide a robust speaker discriminant feature representation.

### 5.1. Training and Extraction of SCN Features

The SCN is two layers deep and uses sigmoid non-linearity in the hidden layers. Each hidden layer consists of 2000 hidden units. The softmax output distribution is over 4323 speakers in the background set (primary background data + oriental background data). The speakers are filtered based on the number of their recordings. Speakers having more than 4 recordings/i-vectors are selected. We make use of 600-dimensional i-vectors that have been adapted to the oriental data for SCN training. The i-vectors are length normalized before being processed by the SCN. After SCN model is trained, it is used to extract features from all training, development (enrollment/test) and evaluation (enrollment/test) data. More specifically, we extract the activations of the last hidden layer and treat them as feature vectors for speaker verification. The SCN features is of 2000-dimensional and we only make these features to be of unit norm and do not perform any mean-centering.

### 5.2. Scoring of SCN-projected Features

With the SCN-projected features speaker verification scoring is done using a cosine distance backend. For speaker models with three enrollment trials we conduct score-level and i-vector/SCN feature-level averaging to provide a single score. To this end with MFCC frontend and cosine distance backend we develop following three system variants:

MFCC-DNN1: This system uses 2000-dimensional length normalized SCN-projected features. Score-level averaging is used for speaker models with 3 enrolment trials during speaker verification by cosine distance backend. Besides this system

all other systems produce a single score by using i-vector/SCN feature-level averaging technique.

MFCC-DNN2: The NAP projection is applied on the top of all speaker features produced by a SCN. After that length normalization is applied to normalize these features to unit norm.

MFCC-DNN3: In this case we reduce the dimension of the NAP projected SCN feature vectors using a principal component analysis (PCA) technique and then normalize to unit norm.

## 6. The Beta-Bernoulli Backend

In this section, we propose a new probabilistic backend to model the hidden activations of the speaker classifier network (SCN) described in the previous section. We refer to this backend as Beta-Bernoulli (BB) backend. In this backend, for each node in the last hidden layer, the activations on the enrollment and test are compared by assuming them to be generated by a biased coin tosses $T^s$. Here $s = 1,2,...,S$ is the speaker index and $T^s$ is modeled by drawing a Bernoulli probability $\pi$ from the Beta distribution $B(a,a')$. Parameters $a$ and $a'$ control the shape of the distribution. Same speaker hypothesis is computed with a single draw from the Beta prior and for different speaker hypothesis there are two draws – one for enrollment and one for test data. Therefore, likelihood ratio $lr_{BB}$ in Beta-Bernoulli (BB) backend is given by [15]:

$$lr_{BB} = \frac{B(a+N^e+N^t, a'+N'^e+N'^t)B(a,a')}{B(a+N^e, a'+N'^e)B(a+N^t, a'+N'^t)}, \quad (1)$$

where $N^e, N'^e$ denote the counts for enrollment data and $N^t, N'^t$ represent the counts for test data.

In this work, we denote the SCN and BB backend combination system as MFCC-DNN-BB. This system differs from MFCC-DNN1 (described in section 5) in that we replaced the cosine distance backend with a probabilistic backend which was trained blindly on the unlabeled training data. We did not attempt to assign speaker, language or gender labels to the training data. As in MFCC-DNN1, the feature vector used to represent an utterance consisted of the sigmoid activations of the last hidden layer of the DNN. We viewed these features as noisy binary vectors and modeled them by a hidden vector of Bernoulli probabilities. If speaker labels were available, we would associate one Bernoulli probability vector with each speaker. Since we did not have speaker labels for the in-domain training set (SRE16UL), we treated the recordings as if they all came from different speakers. We treated the components of the feature vector as being statistically independent and we placed a Beta prior on each of the Bernoulli probabilities. We "estimated" the priors by appealing to the maximum likelihood II principle, using the methods in [15].

## 7. Score Calibration and Fusion

Our submitted fused system's score for the SRE 2016 is obtained by fusing the scores of several sub-systems. The labeled minor SRE16 development data is used for training the fusion parameters. The sub-systems included in the fusion were selected according to individual performance on the labeled minor SRE16 development data. Inclusion or exclusion decisions of sub-systems to the final fused system we made by looking at the regularity of score histograms, DET (detection error trade-off) curves and normalized DCF (detection cost function) curves. As the EER (equal error rate) operating point is too far from the DCF16 (DCF of SRE 2016) operating points we decided not to judge system goodness by EER [2]. CRIM's final submitted system is the fusion of following eight sub-systems: LFCC-CD, LFCC-PLDA, MFCC-CD, LPCC-CD, MFCC-DNN1, MFCC-DNN2, MFCC-DNN3 and MFCC-DNN-BB [2]. Note that MFCC-FC-PLDA and MFCC-FC-CD sub-systems could not make them to the final fused system as these systems scores were not ready before the submission deadline.

Due to data scarcity and to combat over-training, generative fusion and calibration strategies, with as few as possible parameters, were used. The fusion strategy was quadrature-Gaussian pre-calibration (quadrature calibration, qcal) [2] of each sub-system, followed by equal-weighted summation. Single enrollment and three enrollment trials were calibrated separately independent of gender information and the calibrated scores were merged together. The qcal calibration is done by computing the log-LR obtained from a generative model with two univariate Gaussians for targets and non-targets, with different means and covariances. The parameters were estimated with maximum likelihood. The main purpose of pre-calibration before summation is to replace the missing scores (due to VAD failure) by log-LR = 0 and to give roughly same scale to sub-systems' scores. This ensures better system to contribute a bit more than weaker systems [2].

Table 2: *Brief description of systems and their fusions developed at CRIM during and after the 2016 NIST speaker recognition evaluation. Except the MFCC-FC-CD and MFCC-FC-PLDA sub-systems all sub-systems mentioned here are based on diagonal covariance UBM.*

| System Name | Description |
|---|---|
| **MFCC-CD** | MFCC frontend, adapted and NAP projected i-vectors, cosine distance backend |
| **LFCC-CD** | Same as MFCC-CD but with LFCC frontend |
| **LPCC-CD** | Same as MFCC-CD but with LPCC frontend |
| **MFCC-PLDA** | Same as MFCC-CD but with PLDA backend |
| **MFCC-PIVE-PLDA** | Same as MFCC-PLDA but with primary i-vectors (i.e., un-adapted i-vectors) |
| **LFCC-PLDA** | Same as LFCC-CD but with PLDA backend |
| **LPCC-PLDA** | Same as LPCC-CD but with PLDA backend |
| **MFCC-DNN1** | MFCC frontend, SCN features, cosine distance backend |
| **MFCC-DNN2** | MFCC frontend, NAP-projected SCN features, cosine distance backend |
| **MFCC-DNN3** | MFCC frontend, NAP-projected SCN features, PCA to reduce the dimension further, cosine distance backend |
| **MFCC-DNN-BB** | Same as MFCC-DNN1 but with the proposed Beta-Bernoulli backend instead of cosine distance backend |
| **MFCC-FC-CD** | Same as MFCC-CD but with full covariance UBM |
| **MFCC-FC-PLDA** | Same as MFCC-PLDA but with full covariance UBM |
| **FUSION7** | Fusion of MFCC-CD, MFCC-FC-PLDA, MFCC-FC-CD, LFCC-CD, LPCC-CD, MFCC-DNN2, MFCC-DNN-BB |
| **FUSION8** | Fusion of FUSION7 and MFCC-PIVE-PLDA |
| **FUSION4** | Fusion of MFCC-CD, MFCC-FC-CD, MFCC-DNN3 and MFCC-DNN-BB |
| **FUSION5** | Fusion of FUSION4 and MFCC-PIVE-PLDA |
| **SRE16_FUSION** | CRIM's fused system submitted to SRE16 |

## 8. Performance Evaluation

CRIM's systems were intended for the fixed training condition of the 2016 NIST speaker recognition evaluation (SRE16). In table 2 we briefly describe the individual systems and their fusions, including our fused system to SRE16, developed during and after the evaluation. For the evaluation of performances we reported results on the development and evaluation test sets of the 2016 SRE. The evaluation metrics used are equal error rate (EER), minimum Cprimary ($minC_{prm}$) and actual Cprimary ($actC_{prm}$) costs [1]. Among the 13 sub-systems of table 2 only 8 were made to the final fused system (SRE16_FUSION) of CRIM.

Table 3: *Speaker verification performance on the dev and eval test sets of NIST SRE 2016 in terms of EER (in %) measure (unequalized). MFCC-PLDA system is based on adapted i-vectors whereas MFCC-PIVE-PLDA is same as MFCC-PLDA but without domain adaptation using minimum divergence training.*

| | EER | |
|---|---|---|
| | **dev** | **eval** |
| **MFCC-PIVE-PLDA** | 18.68 | 13.10 |
| **MFCC-PLDA** | **17.09** | **11.75** |

Table 3 shows the EERs obtained by the MFCC-PLDA and MFCC-PIVE-PLDA sub-systems on the development and evaluation test sets. It is observed from this table that by adapting the i-vector extractor to the task domain by performing minimum divergence training on the oriental data helped reduce the EER by more than 1.3% absolute. The EERs, $minC_{prm}$ and $_{act}C_{prm}$ costs obtained by evaluating scores of individual and fused systems are shown in tables 4 and 5 for the development (dev) and evaluation (eval) test sets, respectively. Comparing the results of MFCC-DNN2 to that of MFCC-CD/MFCC-PLDA it can be concluded that speaker classifier network (SCN) resulted in further 1% absolute reduction in EER. Compared to other sub-systems SCN-based (either with cosine distance backend or with Beta-Bernoulli backend) sub-systems yielded better performances in terms of $minC_{prm}$ and $_{act}C_{prm}$ costs, specifically on the eval data.

Table 4: *Speaker verification performance on the development (dev) test set of NIST SRE 2016 in terms of EER (in %), $minC_{prm}$ and $actC_{prm}$ costs (unequalized).*

| | **EER** | **$minC_{prm}$** | **$actC_{prm}$** |
|---|---|---|---|
| **MFCC-CD** | 17.58 | 0.7067 | 0.7517 |
| **LFCC-CD** | 19.46 | 0.7741 | 0.7995 |
| **LPCC-CD** | 19.75 | 0.7620 | 0.8033 |
| **MFCC-PLDA** | 17.09 | 0.8830 | 1.0920 |
| **LFCC-PLDA** | 20.75 | 0.8154 | 0.9763 |
| **LPCC-PLDA** | 20.01 | 0.8891 | 1.00 |
| **MFCC-DNN1** | 16.45 | 0.7154 | 0.7849 |
| **MFCC-DNN2** | 15.47 | 0.7198 | 0.7809 |
| **MFCC-DNN3** | 15.51 | 0.7189 | 0.7710 |
| **MFCC-DNN-BB** | 15.01 | 0.8064 | 0.8472 |
| **MFCC-FC-CD** | 17.19 | 0.7136 | 0.7730 |
| **MFCC-FC-PLDA** | 16.38 | 0.6958 | 0.9810 |
| **FUSION7** | **13.88** | **0.6003** | **0.7245** |
| **FUSION8** | **13.88** | **0.5930** | **0.7341** |
| **FUSION4** | **14.01** | **0.6224** | **0.6943** |
| **FUSION5** | **14.31** | **0.6180** | **0.7148** |
| **SRE16_FUSION** | **14.45** | **0.6110** | **0.7536** |

By looking at the performances of MFCC-DNN1 and MFCC-DNN-BB it is evident that proposed Beta-Bernoulli backend aided to reduce the EER both on dev and eval sets. In terms of EER measure MFCC-DNN-BB sub-system performed the best whereas MFCC-DNN3 sub-systems showed best performance in terms of $minC_{prm}$ and $_{act}C_{prm}$ costs. Our best single sub-system in EER measure is MFCC-DNN-BB and in terms of all three evaluation metrics MFCC-DNN3 is the best. In general, the best speaker verification results were obtained when pre-calibrated sub-systems' scores were linearly fused with equal weights. Our submitted fused system SRE16_FUSION gave competitive performance (in EER) to the ABC [2] and the best system [16] of SRE16. Full covariance UBM-based systems contributed in the fused systems FUSION4, FUSION7. Inclusion of MFCC-PIVE-PLDA system, i.e., system based on primary i-vectors, in the fusion (FUSION5 and FUSION8) helped to gain the performance, specifically in EER and $minC_{prm}$ measures.

Table 5: *Speaker verification performance on the evaluation (eval) test set of NIST SRE 2016 in terms of EER (in %), $minC_{prm}$ and $actC_{prm}$ costs (unequalized).*

| | **EER** | **$minC_{prm}$** | **$actC_{prm}$** |
|---|---|---|---|
| **MFCC-CD** | 12.42 | 0.7705 | 0.8800 |
| **LFCC-CD** | 13.95 | 0.8435 | 0.9109 |
| **LPCC-CD** | 14.12 | 0.8481 | 0.9297 |
| **MFCC-PLDA** | 11.75 | 0.8547 | 0.9993 |
| **MFCC-PIVE-PLDA** | 13.10 | 0.8901 | 0.9524 |
| **LFCC-PLDA** | 14.93 | 0.8891 | 0.9603 |
| **LPCC-PLDA** | 14.25 | 0.9357 | 1.0000 |
| **MFCC-DNN1** | 12.49 | 0.7350 | 0.7377 |
| **MFCC-DNN2** | 11.53 | 0.7290 | 0.7366 |
| **MFCC-DNN3** | 11.43 | 0.7255 | 0.7305 |
| **MFCC-DNN-BB** | 11.43 | 0.7435 | 0.7889 |
| **MFCC-FC-CD** | 12.35 | 0.7792 | 0.9091 |
| **MFCC-FC-PLDA** | 11.38 | 0.7646 | 0.9748 |
| **FUSION7** | **9.53** | **0.6874** | **0.7511** |
| **FUSION8** | **9.34** | **0.7092** | **0.7212** |
| **FUSION4** | **9.78** | **0.6915** | **0.6945** |
| **FUSION5** | **9.70** | **0.6864** | **0.6914** |
| **SRE16_FUSION** | **9.90** | **0.6928** | **0.7180** |

### 8.1. CPU Execution Time

In order to report real time factor we carried out experiments on an Intel(R) Xeon(R) CPU X5650@ 2.67GHz with a total memory of 94.5GB. The execution time for the extraction of i-vectors/SCN-projected feature vectors (VAD segmentation to generation of i-vectors/SCN-vectors + enrollment of speaker model + scoring) in a single thread is of 8 times faster than the real time using 3.5GB of memory. The execution time reported above is for systems with diagonal covariance UBM.

## 9. Conclusion

In this paper, we presented CRIM's speaker recognition systems developed for fixed condition task of the 2016 NIST speaker recognition evaluation. Our adopted methods to tackle the major challenges of the evaluation were proved to be helpful. We proposed to take care of the domain mismatch problem by adapting a conventional i-vector extractor to the task domain by performing minimum divergence training on the unlabeled data. Speaker classifier network projected feature representations were found to be more robust than i-vectors under adverse conditions. Speaker classifier network and the proposed Beta-Bernoulli backend combination yielded best single system performance in terms of EER measure. Our submitted system is the equal weighted summation of eight sub-systems and provided competitive performance to the best system of the evaluation.

# 10. References


[1] NIST 2016 Speaker Recognition Evaluation Plan. https://www.nist.gov/sites/default/files/documents/itl/iad/mig/SRE16_Eval_Plan_V1-0.pdf

[2] Niko Brummer, et. al., "ABC NIST SRE 2016 System Description," NIST SRE 2016 Workshop, San Diego, CA, December 2016. http://www.crim.ca/perso/patrick.kenny/ABC_NIST2016_SAN_DIEGO.pdf http://www.crim.ca/perso/patrick.kenny/SRE_2016_ABC_1478007109_abc-nist-sre-systemdescription_v2.pdf

[3] Jahangir Alam, Patrick Kenny, Pierre Ouellet, Themos Stafylakis, and Pierre Dumouchel, "Supervised/unsupervised voice activity detectors for text-dependent speaker recognition on the rsr2015 corpus," In Proceedings of Odyssey Speaker and Language Recognition Workshop, June 2014.

[4] Tomi Kinnunen and Padmanabhan Rajan, "A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data," In Proceedings of International Conference on Acoustics, Speech and Signal Processing, p. 7229–7233, May 2013.

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

[6] Patrick Kenny, "A Small Footprint i-Vector Extractor," in proc. Odyssey Speaker and Language Recognition Workshop, Singapore, June 2012.

[7] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting Baum Welch statistics for speaker recognition," in Proc. Odyssey, 2014.

[8] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically aware deep neural network," in Proc. ICASSP, pp. 1695–1699, 2014.

[9] D. Garcia-Romero, X. Zhang, A. McCree, and D. Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in Proc. SLT Workshop, pp. 378–383, 2014.

[10] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in Proc. ASRU Workshop, pp. 92–97, 2015.

[11] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," Signal Processing Letters, IEEE, vol. 22 (10) pp. 1671–1675, 2015.

[12] David Snyder Pegah Ghahremani, Daniel Povey, Daniel Garcia-Romero, Yishay Carmiel and Sanjeev Khudanpur, "Deep Neural Network-based Speaker Embeddings for End-to-end Speaker Verification," in Proc. SLT Workshop, San Diego, California, USA, 2016.

[13] Gautam Bhattacharya, Jahangir Alam, Patrick Kenny, and Vishwa Gupta, "Modelling speaker and channel variability using deep neural networks for robust speaker verification," in Proc. SLT Workshop, San Diego, California, USA, 2016.

[14] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in Proc. Interspeech, Florence, Italy, Aug. 2011.

[15] T. Minka, "Estimating a Dirichlet distribution," 2012.

[16] Claudio Vair, et. el., "NPT System Description for NIST 2016 Speaker Recognition Evaluation," NIST SRE 2016 Workshop, San Diego, CA, December 2016.