

# THE INTELLIGENT VOICE 2016 SPEAKER RECOGNITION SYSTEM

Abbas Khosravani, Cornelius Glackin, Nazim Dugan, Gérard Chollet, Nigel Cannings

Intelligent Voice Limited, St Clare House, 30-33 Minories, EC3N 1BP, London, UK

## ABSTRACT

This paper presents the Intelligent Voice (IV) system submitted to the NIST 2016 Speaker Recognition Evaluation (SRE). The primary emphasis of SRE this year was on developing speaker recognition technology which is robust for novel languages that are much more heterogeneous than those used in the current state-of-the-art, using significantly less training data, that does not contain meta-data from those languages. The system is based on the state-of-the-art *i*-vector/PLDA which is developed on the fixed training condition, and the results are reported on the protocol defined on the development set of the challenge.

**Index Terms**— Speaker Recognition, Speech Processing

## 1. INTRODUCTION

Compared to previous years, the 2016 NIST speaker recognition evaluation (SRE) marked a major shift from English towards Austronesian and Chinese languages. The task like previous years is to perform speaker detection with the focus on telephone speech data recorded over a variety of handset types. The main challenges introduced in this evaluation are duration and language variability. The potential variation of languages addressed in this evaluation, recording environment, and variability of test segments duration influenced the design of our system. Our goal was to utilize recent advances in language normalization, domain adaptation, speech activity detection and session compensation techniques to mitigate the adverse bias introduced in this year's evaluation.

Over recent years, the *i*-vector representation of speech segments has been widely used by state-of-the-art speaker recognition systems [3]. The speaker recognition technology based on *i*-vectors currently dominates the research field due to its performance, low computational cost and the compatibility of *i*-vectors with machine learning techniques. This dominance is reflected by the recent NIST *i*-vector machine learning challenge [7] which was designed to find the most promising algorithmic approaches to speaker recognition specifically on the basis of *i*-vectors [11, 18, 23, 12]. The outstanding ability of DNN for frame alignment which has achieved remarkable performance in text-independent speaker recognition for English data [13, 9], failed to provide even comparable recognition performance to the traditional

GMM. Therefore, we concentrated on the cepstral based GMM/*i*-vector system.

We outline in this paper the Intelligent Voice system, techniques and results obtained on the SRE 2016 development set that will mirror the evaluation condition as well as the timing report. Section 2 describes the data used for the system training. The front-end and back-end processing of the system are presented in Sections 3 and 4 respectively. In Section 5, we describe experimental evaluation of the system on the SRE 2016 development set. Finally, we present a timing analysis of the system in Section 6.

## 2. TRAINING CONDITION

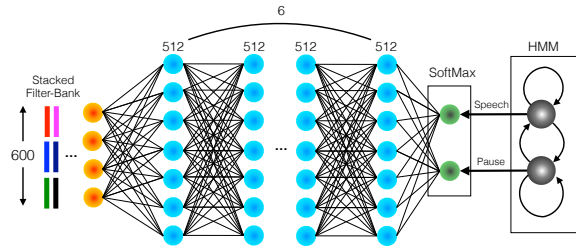
The fixed training condition is used to build our speaker recognition system. Only conversational telephone speech data from datasets released through the linguistic data consortium (LDC) have been used, including NIST SRE 2004-2010 and the Switchboard corpora (Switchboard Cellular Parts I and II, Switchboard2 Phase I,II and III) for different steps of system training. A more detailed description of the data used in the system training is presented in Table 1. We have also included the unlabelled set of 2472 telephone calls from both minor (Cebuano and Mandarin) and major (Tagalog and Cantonese) languages provided by NIST in the system training. We will indicate when and how we used this set in the training in the following sections.

**Table 1.** The description of the data used for training the speaker recognition system.

	#Langs	#Spks		#Segs	
		Male	Female	Male	Female
English	1	1925	2603	19556	25835
non-English	34	274	489	1428	2657

## 3. FRONT-END PROCESSING

In this section we will provide a description of the main steps in front-end processing of our speaker recognition system including speech activity detection, acoustic and *i*-vector feature extraction.



**Fig. 1.** The architecture of our DNN-HMM speech activity detection.

### 3.1. Speech Activity Detection

The first stage of any speaker recognition system is to detect the speech content in an audio signal. An accurate speech activity detector (SAD) can improve the speaker recognition performance. Several techniques have been proposed for SAD, including unsupervised methods based on a thresholding signal energy, and supervised methods that train a speech/non-speech classifier such as support vector machines (SVM) [16] and Gaussian mixture models (GMMs) [17]. Hidden markov models (HMMs) [19] have also been successful. Recently, it has been shown that DNN systems achieve impressive improvement in performance especially in low signal to noise ratios (SNRs) [21]. In our work we have utilized a two-class DNN-HMM classifier to perform this task. The DNN-HMM hybrid configuration with cross-entropy as the objective function has been trained with the back-propagation algorithm. The softmax layer produces posterior probabilities for speech and non-speech which were then converted into log-likelihoods. Using 2-state HMMs corresponding to speech and non-speech, frame-wise decisions are made by Viterbi decoding. As input to the network, we fed 40-dimensional filter-bank features along with 7 frames from each side. The network has 6 hidden layers with 512 units each. The architecture of our DNN-HMM SAD is shown in Figure 1. Approximately 100 hours of speech data from the Switchboard telephony data with word alignments as ground-truth were used to train our SAD. The DNN training is performed on an NVIDIA TITAN X GPU, using Kaldi software [20]. Evaluated on 50 hours of telephone speech data from the same database, our DNN-HMM SAD indicated a frame-level miss-classification (speech/non-speech) rate of 5.9% whereas an energy-based SAD did not perform better than 20%.

### 3.2. Acoustic Features

For acoustic features we have experimented with different configurations of cepstral features. We have used 39-dimensional PLP features and 60-dimensional MFCC features (including their first and second order derivatives) as acoustic features. Moreover, our experiments indicated that

the combination of these two feature sets performs particularly well in score fusion. Both PLP and MFCC are extracted at 8kHz sample frequency using Kaldi [20] with 25 and 20 ms frame lengths, respectively, and a 10 ms overlap (other configurations are the same as Kaldi defaults). For each utterance, the features are centered using a short-term (3s window) cepstral mean and variance normalization (ST-CMVN). Finally, we employed our DNN-HMM speech activity detector (SAD) to drop non-speech frames.

### 3.3. *i*-Vector Features

Since the introduction of *i*-vectors in [3], the speaker recognition community has seen a significant increase in recognition performance. *i*-Vectors are low-dimensional representations of Baum-Welch statistics obtained with respect to a GMM, referred to as *universal background model* (UBM), in a single subspace which includes all characteristics of speaker and inter-session variability, named *total variability matrix* [3]. We trained on each acoustic feature a full covariance, gender-independent UBM model with 2048 Gaussians followed by a 600-dimensional *i*-vector extractor to establish our MFCC- and PLP-based *i*-vector systems. The unlabeled set of development data was used in the training of both the UBM and the *i*-vector extractor. The open-source Kaldi software has been used for all these processing steps [20].

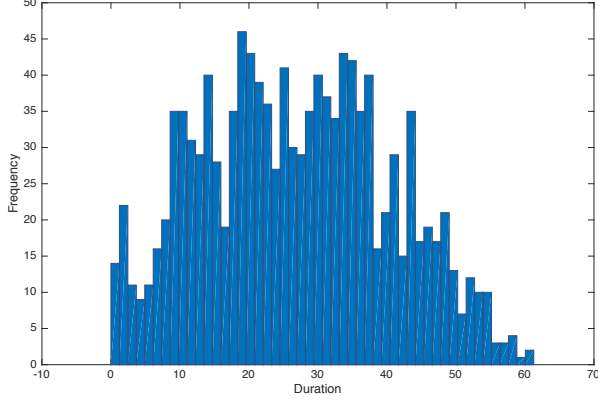
It has been shown that successive acoustic observation vectors tend to be highly correlated. This may be problematic for maximum a posteriori (MAP) estimation of *i*-vectors. To investigating this issue, scaling the zero and first order Baum-Welch statistics before presenting them to the *i*-vector extractor has been proposed. It turns out that a scale factor of 0.33 gives a slight edge, resulting in a better decision cost function [10]. This scaling factor has been performed in training the *i*-vector extractor as well as in the testing.

## 4. BACK-END PROCESSING

This section provides the steps performed in back-end processing of our speaker recognition system.

### 4.1. Nearest-neighbor Discriminant Analysis (NDA)

The nearest-neighbor discriminant analysis is a nonparametric discriminant analysis technique which was proposed in [4], and recently used in speaker recognition [22]. The non-parametric within- and between-class scatter matrices  $\hat{S}_w$  and  $\hat{S}_b$ , respectively, are computed based on  $k$  nearest neighbor sample information. The NDA transform is then formed using eigenvectors of  $\hat{S}_w^{-1}\hat{S}_b$ . It has been shown that as the number of nearest neighbors  $k$  approaches the number of samples in each class, the NDA essentially becomes the LDA projection. Based on the finding in [22], NDA outperformed LDA due to



**Fig. 2.** The duration of test segments in the development set after dropping non-speech frames.

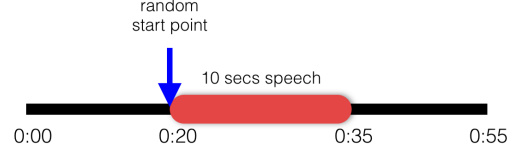
the ability in capturing the local structure and boundary information within and across different speakers. We applied a  $600 \times 400$  NDA projection matrix computed using the 10 nearest sample information on centered  $i$ -vectors. The resulting dimensionality reduced  $i$ -vectors are then whitened using both the training data and the unlabelled development set.

#### 4.2. Short-Duration Variability Compensation

The enrolment condition of the development set is supposed to provide at least 60 seconds of speech data for each target speaker. Nevertheless, our SAD indicates that the speech content is as low as 26 seconds in some cases. The test segments duration which ranges from 9 to 60 seconds of speech material can result in poor performance for lower duration segments. As indicated in Figure 2, more than one third of the test segments have speech duration of less than 20 seconds. We have addressed this issue by proposing a short duration variability compensation method. The proposed method works by first extracting from each audio segment in the unlabelled development set, a partial excerpt of 10 seconds of speech material with random selection of the starting point (Figure 3). Each audio file in the unlabelled development set, with the extracted audio segment will result in two 400-dimensional  $i$ -vectors, one with at most 10 seconds of speech material. Considering each pair as one class, we computed a  $400 \times 390$  LDA projection matrix to remove directions attributed to duration variability. Moreover, the projected  $i$ -vectors are also subjected to a within-class covariance normalization (WCCN) using the same class labels.

#### 4.3. Language Normalization

Language-source normalization is an effective technique for reducing language dependency in the state-of-the-art  $i$ -vector/PLDA speaker recognition system [14]. It can be implemented by extending SN-LDA [15] in order to mitigate



**Fig. 3.** Partial excerpt of 10 second speech duration from an audio speech file.

variations that separate languages. This can be accomplished by using the language label to identify different sources during training. Language Normalized-LDA (LN-LDA) utilizes a language-normalized within-speaker scatter matrix  $\hat{\mathbf{S}}_W$  which is estimated as the variability not captured by the between-speaker scatter matrix,

$$\hat{\mathbf{S}}_W = \mathbf{S}_T - \hat{\mathbf{S}}_B, \quad (1)$$

where  $\mathbf{S}_T$  and  $\hat{\mathbf{S}}_B$  are the total scatter and normalized between-speaker scatter matrices respectively, and are formulated as follows:

$$\mathbf{S}_T = \sum_{n=1}^N \mathbf{w}_n \mathbf{w}_n^T, \quad (2)$$

where  $N$  is the total number of  $i$ -vectors and

$$\hat{\mathbf{S}}_B = \sum_{l=1}^L \sum_{s=1}^{S_l} n_s^l (\bar{\mathbf{w}}^l(s) - \bar{\mathbf{w}}^l) (\bar{\mathbf{w}}^l(s) - \bar{\mathbf{w}}^l)^T, \quad (3)$$

where  $L$  is the number of languages in the training set,  $S_l$  is the number of speakers in language  $l$ ,  $\bar{\mathbf{w}}^l(s)$  is the mean of  $n_s^l$   $i$ -vectors from speaker  $s$  and language  $l$  and finally  $\bar{\mathbf{w}}^l$  is the mean of all  $i$ -vectors in language  $l$ . We applied a  $390 \times 300$  SN-LDA projection matrix to reduce the  $i$ -vector dimensions down to 300.

#### 4.4. PLDA

Probabilistic Linear Discriminant Analysis (PLDA) provides a powerful mechanism to distinguish between-speaker variability, separating sources which characterizes speaker information, from all other sources of undesired variability that characterize distortions. Since  $i$ -vectors are assumed to be generated by some generative model, we can break it down into statistically independent speaker- and session-components with Gaussian distributions [5, 8]. Although it has been shown that their distribution follow Student's  $t$  rather than Gaussian [8] distributions, length normalizing the entire set of  $i$ -vectors as a pre-processing step can approximately Gaussianize their distributions [5] and as a result improve the performance of Gaussian PLDA to that of heavy-tailed PLDA

[8]. A standard Gaussian PLDA assumes that an  $i$ -vector  $\mathbf{w}$ , is modelled according to

$$\mathbf{w} = \mathbf{m} + \mathbf{V}\mathbf{y} + \varepsilon. \quad (4)$$

where,  $\mathbf{m}$  is the mean of  $i$ -vectors, the columns of matrix  $\mathbf{V}$  contains the basis for the between-speaker subspace, the latent identity variable  $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  denotes the speaker factor that represents the identity of the speaker and the residual  $\varepsilon$  which is normally distributed with zero mean and full covariance matrix  $\Sigma$ , represents within-speaker variability.

For each acoustic feature we have trained two PLDA models. The first out-domain PLDA ( $\mathbf{V}_{out}, \Sigma_{out}$ ) is trained using the training set presented in Table 1, and the second in-domain PLDA ( $\mathbf{V}_{in}, \Sigma_{in}$ ) was trained using the unlabelled development set. Our efforts to cluster the development set (e.g using the out-domain PLDA) was not very successful as it sounds that almost all of them are uttered by different speakers. Therefore, each  $i$ -vector was considered to be uttered by one speaker. We also set the number of speaker factors to 200.

#### 4.5. Domain Adaptation

Domain adaptation has gained considerable attention with the aim of compensating for cross-speech-source variability of in-domain and out-of-domain data. The framework presented in [6] for unsupervised adaptation of out-domain PLDA parameters resulted in better performance for in-domain data. Using in-domain and out-domain PLDA trained in Section 4.4, we interpolated their parameters as follow:

$$\begin{aligned} \mathbf{V}_{adapt} &= \alpha \mathbf{V}_{in} + (1 - \alpha) \mathbf{V}_{out} \\ \Sigma_{adapt} &= \alpha \Sigma_{in} + (1 - \alpha) \Sigma_{out}. \end{aligned} \quad (5)$$

We chose  $\alpha = 0.10$  for making our submission.

#### 4.6. Score Computation and Normalization

For the one-segment enrolment condition, the speaker model is the length normalized  $i$ -vector of that segment, however, for the three-segment enrolment condition, we simply used a length-normalized mean vector of the length-normalized  $i$ -vectors as the speaker model. Each speaker model is tested against each test segment as in the trial list. For each two trial  $i$ -vectors  $\mathbf{w}_1$  and  $\mathbf{w}_2$ , the PLDA score is computed as

$$s = \mathbf{w}_1^T \mathbf{Q} \mathbf{w}_1 + \mathbf{w}_2^T \mathbf{Q} \mathbf{w}_2 + 2 \mathbf{w}_1^T \mathbf{P} \mathbf{w}_2 + c, \quad (6)$$

in which

$$\mathbf{Q} = \mathbf{S}_T^{-1} - (\mathbf{S}_T - \mathbf{S}_B \mathbf{S}_T^{-1} \mathbf{S}_B)^{-1}, \quad (7)$$

$$\mathbf{P} = \mathbf{S}_T^{-1} \mathbf{S}_B (\mathbf{S}_T - \mathbf{S}_B \mathbf{S}_T^{-1} \mathbf{S}_B)^{-1}. \quad (8)$$

and  $\mathbf{S}_B = \mathbf{V}_{adapt} \mathbf{V}_{adapt}^T$  and  $\mathbf{S}_T = \mathbf{S}_B + \Sigma_{adapt}$ . It has been shown and proved in our experiments that score normalization can have a great impact on the performance of the

recognition system. We used the symmetric s-norm proposed in [8] which normalizes the score  $s$  of the pair  $(w_1, w_2)$  using the formula

$$\hat{s} = \frac{s - \mu_1}{\sigma_1} - \frac{s - \mu_2}{\sigma_2} \quad (9)$$

where the means  $\mu_1, \mu_2$  and standard deviations  $\sigma_1, \sigma_2$  are computed by matching  $w_1$  and  $w_2$  against the unlabelled set as the impostor speakers, respectively.

#### 4.7. Quality Measure Function

It has been shown that there is a dependency between the value of the  $C_{det}^{min}$  threshold and the duration of both enrolment and test segments. Applying the quality measure function (QMF) [18] enabled us to compensate for the shift in the  $C_{det}^{min}$  threshold due to the differences in speech duration. We conducted some experiments to estimate the dependency between the  $C_{det}^{min}$  threshold shift on the duration of test segment and used the following QMF for PLDA verification scores:

$$QMF(t) = -0.2\sqrt{t} \quad (10)$$

where  $t$  is the duration of the test segment in seconds.

#### 4.8. Calibration

In the literature, the performance of speaker recognition is usually reported in terms of calibrated-insensitive equal error rate (EER) or the minimum decision cost function ( $C_{det}^{min}$ ). However, in real applications of speaker recognition there is a need to present recognition results in terms of calibrated log-likelihood-ratios. We have utilized the BOSARIS Toolkit [1] for calibration of scores.  $C_{det}^{min}$  provides an ideal reference value for judging calibration. If  $C_{det} - C_{det}^{min}$  is minimized, then the system can be said to be well calibrated.

The choice of target probability ( $P_{tar}$ ) had a great impact on the performance of the calibration. However, we set  $P_{tar} = 0.0001$  for our primary submission which performed the best on the development set. For our secondary submission  $P_{tar} = 0.001$  was used.

## 5. RESULTS AND DISCUSSION

In this section we present the results obtained on the protocol provided by NIST on the development set which is supposed to mirror that of evaluation set. The results are shown in Table 2. The first part of the table indicates the result obtained by the primary system. As can be seen, the fusion of MFCC and PLP (a simple sum of both MFCC and PLP scores) resulted in a relative improvement of almost 10%, as compared to MFCC alone, in terms of both  $C_{det}$  and  $C_{det}^{min}$ . In order to quantify the contribution of the different system components we have defined different scenarios. In scenario A, we have analysed the effect of using LDA instead of NDA. As can be seen from the results, LDA outperforms NDA in the case of

**Table 2.** Performance comparison of the Intelligent Voice speaker recognition system with various analysis on the development protocol of NIST SRE 2016.

Acoustic Features	Unequalized			Equalized		
	EER	$C_{det}^{min}$	$C_{det}$	EER	$C_{det}^{min}$	$C_{det}$
<b>Primary</b>						
MFCC	16.49	0.6633	0.6754	15.83	0.6650	0.6749
PLP	17.87	0.6857	0.6977	16.84	0.6914	0.6982
Fusion	<b>16.04</b>	<b>0.6012</b>	<b>0.6107</b>	<b>14.93</b>	<b>0.6011</b>	<b>0.6267</b>
<b>Scenario A</b>						
MFCC	16.82	0.6658	0.6794	16.42	0.6890	0.7021
PLP	16.98	0.6691	0.6881	16.28	0.6903	0.7092
Fusion	15.73	0.6153	0.6369	15.12	0.6587	0.6964
<b>Scenario B</b>						
MFCC	16.55	0.6735	0.6880	16.10	0.6755	0.6945
PLP	18.27	0.6938	0.7141	16.97	0.7018	0.7299
Fusion	16.31	0.6075	0.6299	14.70	0.6259	0.6482
<b>Scenario C</b>						
MFCC	17.08	0.6767	0.6889	16.77	0.6677	0.6927
PLP	17.98	0.6857	0.6968	17.21	0.7001	0.7192
Fusion	16.59	0.6176	0.6264	15.70	0.6363	0.6680
<b>Scenario D</b>						
MFCC	17.42	0.6694	0.6833	16.54	0.6639	0.6820
PLP	18.49	0.6851	0.7062	17.46	0.6852	0.7054
Fusion	17.03	0.6171	0.6315	15.73	0.6243	0.6410
<b>Scenario E</b>						
MFCC	16.65	0.6976	0.7124	16.24	0.6972	0.7122
PLP	18.48	0.7182	0.7324	17.49	0.7263	0.7480
Fusion	16.82	0.6343	0.6500	15.52	0.6471	0.6737

PLP, however, in fusion we can see that NDA resulted in better performance in terms of the primary metric. In scenario B, we analysed the effect of using the short-duration compensation technique proposed in Section 4.2. Results indicate superior performance using this technique. In scenario C, we investigated the effects of language normalization on the performance of the system. If we replace LN-LDA with simple LDA, we can see performance degradation in MFCC as well as fusion, however, PLP seems not to be adversely affected. The effect of using QMF is also investigated in scenario D. Finally in scenario E, we can see the major improvement obtained through the use of the domain adaptation technique explained in Section 4.5. For our secondary submission, we incorporated a disjoint portion of the labelled development set (10 out of 20 speakers) in either LN-LDA and in-domain PLDA training. We evaluated the system on almost 6k out of 24k trials from the other portion to avoid any over-fitting, particularly important for the domain adaptation technique. This resulted in a relative improvement of 11% compared to the primary system in terms of the primary metric. However, the results can be misleading, since the recording condition may

be the same for all speakers in the development set.

## 6. TIME ANALYSIS

This section reports on the CPU execution time (single threaded), and the amount of memory used to process a single trial, which includes the time for creating models from the enrolment data and the time needed for processing the test segments. The analysis was performed on an Intel(R) Xeon(R) CPU E5-2670 2.60GHz. The results are shown in Table 3. We used the time command in Unix to report these results. The *user time* is the actual CPU time used in executing the process (single thread). The *real time* is the wall clock time (the elapsed time including time slices used by other processes and the time the process spends blocked). The *system time* is also the amount of CPU time spent in the kernel within the process. We have also reported the memory allocated for each stage of execution. The most computationally intensive stage is the extraction of *i*-vectors (both MFCC- and PLP-based *i*-vectors), which also depends on the duration of the segments. For enrolment, we have reported the time

**Table 3.** CPU execution time and the amount of memory required to process a single trial.

	segment dur	speech dur	stage	user time	system time	real time	memory(MB)
Enrolment	140s	60s	features	0.99s	0.02s	1.02s	12
			SAD	8.10s	0.23s	2.26s	25.0
			<i>i</i> -vectors	27.47s	2.20s	7.83s	4,014
Test	36s	25s	features	0.52s	0.01s	0.54s	7.5
			SAD	2.26s	0.09s	0.94s	25.35
			<i>i</i> -vectors	26.02s	2.25s	7.9s	4,013

required to extract a model from a segment with a duration of 140 seconds and speech duration of 60 seconds. The time and memory required for front-end processing are negligible compared to the *i*-vector extraction stage, since they only include matrix operations. The time required for our SAD is also reported which increases linearly with the duration of segment.

## 7. CONCLUSIONS AND PERSPECTIVES

We have presented the Intelligent Voice speaker recognition system used for the NIST 2016 speaker recognition evaluation. Our system is based on a score fusion of MFCC- and PLP-based *i*-vector/PLDA systems. We have described the main components of the system including, acoustic feature extraction, speech activity detection, *i*-vector extraction as front-end processing, and language normalization, short-duration compensation, channel compensation and domain adaptation as back-end processing. For our future work, we intend to use the ALISP segmentation technique [2] in order to extract meaningful acoustic units so as to train supervised GMM or DNN models.

## 8. REFERENCES

- [1] N. Brümmer and E. de Villiers. The bosaris toolkit user guide: Theory, algorithms and code for binary classifier score processing. *Documentation of BOSARIS toolkit*, 2011.
- [2] G. Chollet, J. Černocký, A. Constantinescu, S. Deligne, and F. Bimbot. Toward alisp: A proposal for automatic language independent speech processing. In *Computational Models of Speech Pattern Processing*, pages 375–388. Springer, 1999.
- [3] 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*, 19(4):788–798, 2011.
- [4] K. Fukunaga and J. Mantock. Nonparametric discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):671–678, 1983.
- [5] D. Garcia-Romero and C. Y. Espy-Wilson. Analysis of *i*-vector length normalization in speaker recognition systems. In *INTERSPEECH*, pages 249–252, 2011.
- [6] 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*, Joensuu, Finland, 2014.
- [7] C. S. Greenberg, D. Bansé, G. R. Doddington, D. Garcia-Romero, J. J. Godfrey, T. Kinnunen, A. F. Martin, A. McCree, M. Przybocki, and D. A. Reynolds. The nist 2014 speaker recognition *i*-vector machine learning challenge. In *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, Joensuu, Finland, 2014.
- [8] P. Kenny. Bayesian speaker verification with heavy-tailed priors. In *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, page 14, 2010.
- [9] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam. Deep neural networks for extracting baum-welch statistics for speaker recognition. In *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, pages 293–298, Joensuu, Finland, 2014.
- [10] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel. Plda for speaker verification with utterances of arbitrary duration. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7649–7653. IEEE, 2013.
- [11] A. Khosravani and M. Homayounpour. Linearly constrained minimum variance for robust *i*-vector based speaker recognition. In *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, pages 249–253, Joensuu, Finland, 2014.
- [12] E. Khoury, L. El Shafey, M. Ferras, and S. Marcel. Hierarchical speaker clustering methods for the nist *i*-vector challenge. In *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, Joensuu, Finland, 2014.

- [13] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1695–1699. IEEE, 2014.
- [14] M. McLaren, M. I. Mandasari, and D. A. van Leeuwen. Source normalization for language-independent speaker recognition using i-vectors. In *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, pages 55–61, Singapore, 2012.
- [15] M. McLaren and D. Van Leeuwen. Source-normalized lda for robust speaker recognition using i-vectors from multiple speech sources. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(3):755–766, 2012.
- [16] N. Mesgarani, M. Slaney, and S. A. Shamma. Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):920–930, 2006.
- [17] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Vesely, and P. Matejka. Developing a speech activity detection system for the darpa rats program. In *INTERSPEECH*, pages 1969–1972, 2012.
- [18] S. Novoselov, T. Pekhovsky, and K. Simonchik. Stc speaker recognition system for the nist i-vector challenge. In *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, pages 231–240, Joensuu, Finland, 2014.
- [19] T. Pfau, D. P. Ellis, and A. Stolcke. Multispeaker speech activity detection for the icsi meeting recorder. In *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*, pages 107–110. IEEE, 2001.
- [20] 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*. IEEE Signal Processing Society, 2011.
- [21] N. Ryant, M. Liberman, and J. Yuan. Speech activity detection on youtube using deep neural networks. In *INTERSPEECH*, pages 728–731, 2013.
- [22] S. O. Sadjadi, S. Ganapathy, and J. Pelecanos. The ibm 2016 speaker recognition system. In *Odyssey 2016: The Speaker and Language Recognition Workshop*, pages 174–180, Bilbao, Spain, June 21–24 2016.
- [23] B. Vesnicer, J. Zganec-Gros, S. Dobrisek, and V. Struc. Incorporating duration information into i-vector-based speaker-recognition systems. In *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, pages 241–248, Joensuu, Finland, 2014.