Denoising autoencoder-based speaker feature restoration for utterances of short duration

Hitoshi Yamamoto and Takafumi Koshinaka

Information and Media Processing Laboratory, NEC Corporation, Japan

h-yamamoto@bc.jp.nec.com

10.21437/Interspeech.2015-283

Abstract

This paper describes a speaker feature restoration method for improving text-independent speaker recognition with short utterances. The method employs a denoising autoencoder (DAE) to compensate speaker features of a short utterance which contains limited phonetic information. It first estimates phonetic distribution in the utterance as posteriors based on speech models and then transforms an i-vector of the utterance using DAE along with the phonetic posteriors. The DAE-based transformation is able to produce a reliable speaker feature with help of supervised training using pairs of long and short speech segments. Speaker recognition experiments on an NIST SRE task demonstrate a 37.9% error reduction.

Index Terms: speaker recognition, short utterance, i-vector, feature compensation, denoising autoencoder

1. Introduction

Text-independent speaker recognition technologies have constantly been improved over the past several decades [1]. A speaker recognition system extracts speaker characteristics from an utterance and then estimates or validates its speaker ID. As a feature representing speaker characteristics, i-vectors [2] have been widely used. Probabilistic linear discriminant analysis (PLDA) [3] has also become a common recognizer for such i-vector-based speaker representation. The latest speaker recognition evaluation campaign, "the i-vector challenge," coordinated by the National Institute of Standards and Technology (NIST), has focused on such state-of-the-art technologies [4].

This paper describes a restoration method of speaker features for improving text-independent speaker recognition with short utterances. Generally, speaker recognition systems perform worse as the duration of utterances become shorter. This is mainly because the phonetic distribution becomes more unbalanced and a speaker feature estimated on such an imbalanced phone distribution becomes statistically less reliable. Conventional GMM- or SVM-based systems suffer from this problem [5, 6]. Degradation has also been observed in a series of experiments using i-vectors [7]. The error rate was reported to increase as the duration of utterances decreased (e.g., from 2.5 minutes to 2 seconds). In practical speaker verification applications, often only short speech segments are observed during testing, even when long speech recordings are available during enrollment.

The problem of performance degradation for i-vector-based speaker recognition with short utterances has been addressed in the literature. The most common solution is to incorporate the duration of utterances into a process of speaker recognition, for instance, i-vector extraction [8], i-vector normalization [9, 10], PLDA modeling [11, 12, 13], and score normalization [14, 15]. These methods utilize the duration itself or a variance of the i-vectors as an indicator of variability.

Another solution focuses on phonetic information as a characteristic of short utterances. Each utterance contains different phonemes, so such phonetic structures represent each short utterance in more detail than the duration information. For the GMM-based systems, for example, text-constrained recognizer [16] or frame selection [17] were investigated to distinguish speakers by means of specific words or syllables. For i-vectors, phonetically-constrained methods [18, 19] have harnessed such features to improve text-dependent system. Content matching method has shown improvement in text-independent systems when the lexical content of a given test utterance is pronounced in the enrollment data [20].

In the machine learning field, a new methodology named deep learning represents a major advance. Neural networks trained with deep learning outperform the conventional statistical models in several pattern recognition applications such as image recognition [21] or speech recognition [22]. A denoising autoencoder (DAE) [23] is a kind of neural network typically used to reduce the noise factor in input signals. For example, DAE can be applied to speech enhancement for noisy [24] or reverberant [25] speech signals.

In this paper, we propose a DAE-based speaker feature restoration method for text-independent speaker recognition using short utterances. The method employs the DAE to compensate the phonetic imbalance in a short utterance. For each utterance, we first estimate not only its i-vector but also a phonetic vector which represents its phonetic distribution. Then, the DAE transforms them into its canonical i-vector which could be obtained when its phonetic distribution is balanced. The DAE learns various kinds of phonetic distributions by its training using many pairs of short and long speech segments sampled from enrollment data. In addition, the subsequent speaker recognition does not require any lexical constraints used in the conventional methods, so existing text-independent systems are available without any modification.

This paper is organized as follows: Section 2 introduces key technologies composing the baseline system of speaker recognition; Section 3 presents our method, i-vector restoration based on a denoising autoencoder; and Section 4 describes experimental evaluation results for speaker recognition in an NIST SRE task. In Section 5, we summarize our work and discuss future issues.

2. i-vector and PLDA system

This section briefly presents two technologies widely used in state-of-the-art speaker recognition systems, i-vectors and Probabilistic Linear Discriminant Analysis (PLDA).

2.1. i-vectors

An i-vector is a feature for speaker recognition that is extracted from an utterance based on factor analysis. As described in [2], factor analysis is used to define a new low-dimensional space named a total variability space. In this new space, a given speech utterance is represented by a vector named the i-vector.

Given an utterance, its feature vector $\boldsymbol{M}$ such as a GMM supervector [26] is written as follows,

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w}, \tag{1}$$

where $\boldsymbol{m}$ is the speaker- and channel-independent supervector typically taken from the universal background model (UBM), the total variability matrix $\boldsymbol{T}$ is a rectangular matrix of low rank and $\boldsymbol{w}$ is an i-vector.

The i-vector for a given utterance u can be obtained using the following equation,

$$\boldsymbol{w}(u) = (\boldsymbol{I} + \boldsymbol{T}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{N}(u)\boldsymbol{T})^{-1}\boldsymbol{T}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{F}(u), \tag{2}$$

where $\boldsymbol{\Sigma}$ is a diagonal covariance of the supervector estimated during factor analysis training. This equation loads two statistics, $\boldsymbol{N}(u)$ and $\boldsymbol{F}(u)$, which consist of the elements written as follows. When the Gaussian mixture model is used as a UBM, the elements on UBM mixture component c are

$$\boldsymbol{N}_c(u) = \sum_{t=1}^{L} p(c|\boldsymbol{u}_t), \tag{3}$$

$$\boldsymbol{F}_c(u) = \sum_{t=1}^{L} p(c|\boldsymbol{u}_t)(\boldsymbol{u}_t - \boldsymbol{m}_c), \tag{4}$$

where $p(c|\cdot)$ corresponds to the posterior probability of mixture component c, $\boldsymbol{u}_t$ is a t-th frame of utterance u that has L frames, and $\boldsymbol{m_c}$ is the mean of the component c. In this way, the i-vector is based not only on its speakers' characteristics but also on its phonetic characteristics, i.e., what words or phrases are spoken in the utterance.

Because an utterance with short duration does not always contain various kinds of phonemes, the posterior $p(c|u)$ would naturally get close to zero for components corresponding to phonemes that never appear in the utterance u. Thus, for such a short utterance, the i-vector would have large variability, and this is the issue we are going to tackle in this paper.

2.2. PLDA

PLDA [3] is a probabilistic model that is widely utilized as a generative model of i-vectors in the speaker recognition field. The model is written as follows,

$$\boldsymbol{w} = \bar{\boldsymbol{w}} + \boldsymbol{\Phi}\boldsymbol{\beta} + \boldsymbol{\Gamma}\boldsymbol{\alpha} + \boldsymbol{\epsilon}, \tag{5}$$

where $\boldsymbol{w}$ is an i-vector of a given utterance, $\bar{\boldsymbol{w}}$ is the global offset in i-vector space, $\boldsymbol{\Phi}$ and $\boldsymbol{\Gamma}$ are matrices of eigenvoices and eigenchannels, $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ are speaker- and channel- factors, and $\boldsymbol{\epsilon}$ is residual noise.

A PLDA-based speaker verification system measures the similarity between two given i-vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ based on the likelihood ratio defined as,

$$s(\boldsymbol{w}_1, \boldsymbol{w}_2) = \frac{p(\boldsymbol{w}_1, \boldsymbol{w}_2|H_1)}{p(\boldsymbol{w}_1|H_0)p(\boldsymbol{w}_2|H_0)}, \tag{6}$$

where the hypothesis H_1 represents the claim that the two i-vectors belong to the same speaker, while H_0 represents the claim that they are from different speakers.

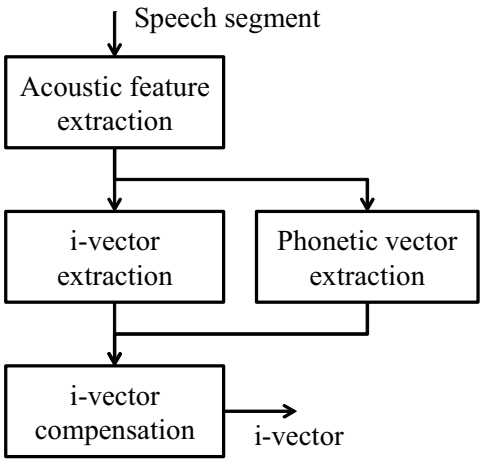

Figure 1: *The framework of our speaker feature restoration.*

3. Speaker Feature Restoration

The framework of our speaker feature restoration method is shown in Figure 1. This system consists of four processing units and works as follows. The acoustic feature extraction unit processes an input utterance and produces a time series of acoustic feature vectors such as MFCC. The i-vector extraction unit extracts an i-vector as a speaker feature of the utterance from the acoustic feature vectors. The phonetic vector extraction unit calculates the posterior distribution over phonetic classes using a speech model such as a Gaussian mixture model. Then, a new i-vector compensation unit accepts both the i-vector and the phonetic vector and transforms them with a denoising autoencoder to restore a desirable i-vector.

3.1. Phonetic vector

Short utterances are supposed to contain limited phonemes (to miss some phonemes), as analyzed in [12], for instance. Thus, to utilize such characteristics of short utterances in detail, we propose using phonetic posterior probabilities in an utterance as its feature. We call it a *phonetic vector*, and make use of the speech model such as the Gaussian mixture model to estimate it of a given utterance u as follows,

$$p_c(u) = \frac{1}{L}\sum_{t=1}^{L} p(c|\boldsymbol{u}_t), \tag{7}$$

where $p_c(u)$ is an element of the phonetic vector $\boldsymbol{p}(u)$ corresponding to the Gaussian index c. In this way, the phonetic vector is obtained without text information of the utterance.

3.2. i-vector compensation using denoising autoencoder

This subsection presents our i-vector compensation method that uses a denoising autoencoder (DAE). The autoencoder (AE) is a kind of neural network generally used for learning efficient distributed representations for a set of data. The DAE is an extension of the AE, which is trained to reconstruct the input from a corrupted version [23]. In the speech processing field, the DAE has been applied to noisy or reverberant speech to enhance target speech signals [24, 25].

Our DAE for i-vector compensation is depicted in Figure 2. The input layer accepts both i-vector $\boldsymbol{w}(u)$ and pho-

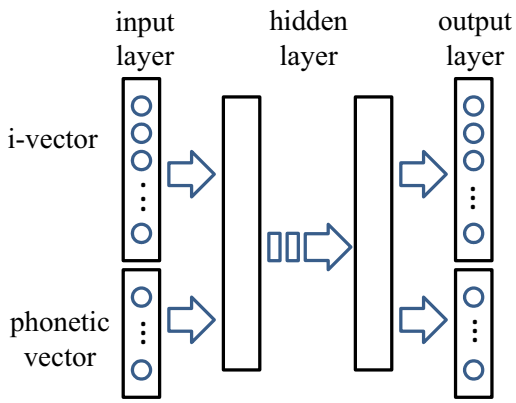

Figure 2: *Denoising autoencoder for i-vector compensation.*

netic vector $\boldsymbol{p}(u)$ extracted from utterance u. The input vector $\boldsymbol{x}_1(u) = [\boldsymbol{w}_1(u), \boldsymbol{p}_1(u)] = [\boldsymbol{w}(u), \boldsymbol{p}(u)]$ is propagated to the succeeding layers with linear and/or non-linear transformations. Such a transformation of the vector $\boldsymbol{x}_i(u)$ to $\boldsymbol{x}_{i+1}(u)$ in one propagation step from i-th layer to $(i+1)$-th layer is written as follows,

$$\boldsymbol{x}_{i+1}(u) = g(\boldsymbol{W}_{(i,i+1)}\boldsymbol{x}_i(u) + \boldsymbol{b}_{(i,i+1)}), \quad (8)$$

where $g()$ is a non-linear function such as the sigmoid, $\boldsymbol{W}_{(i,i+1)}$ and $\boldsymbol{b}_{(i,i+1)}$ are parameters of the DAE. Finally, at the output N-th layer, a vector $\boldsymbol{x}_N(u) = [\boldsymbol{w}_N(u), \boldsymbol{p}_N(u)]$ is produced, and we call $\boldsymbol{w}^c(u) = \boldsymbol{w}_N(u)$ a *compensated i-vector*.

The DAE can be trained in a supervised manner with the training data which is a set of pairs of target vectors and corrupted vectors. The procedure of preparing the training data is:

1. Prepare long speech recordings such as an enrollment data in standard speaker verification systems.
2. For each speech recording S_i, extract the i-vector and the phonetic vector, and form the target vector $\boldsymbol{x}(S_i)$.
3. Sample many short segments from S_i. Then, for each segment $s_{i,j}$, extract the i-vector and the phonetic vector, and form the corrupted vector $\boldsymbol{x}(s_{i,j})$.
4. Append a pair $(\boldsymbol{x}(S_i), \boldsymbol{x}(s_{i,j}))$ to the training data.

The parameters of the DAE can be optimized with a numerical method to minimize objective functions such as the mean square error between the target vector $\boldsymbol{x}(S_i)$ and the DAE compensated vector of the corrupted vector $\boldsymbol{x}(s_{i,j})$.

The DAE learns the transformation between i-vectors of short and long utterances, by using these training data including various kinds of phonetic distributions. With help of the phonetic vector, the DAE produces different i-vectors even when i-vectors of short utterances are similar to each other. Such property would contribute to reduce mis-restoration. In addition, it works without any modification in the existing text-independent speaker recognition system. Thus, our i-vector restoration is expected to improve text-independent speaker recognition using short utterances.

3.3. i-vector combination

Once the compensation process is completed, two i-vectors $\boldsymbol{w}$ and $\boldsymbol{w}^c$ are available in the subsequent speaker recognition process. Thus, we consider a combination method of the two i-vectors to make use of these features as much as possible.

One such combination is an i-vector level combination. For example, simple linear interpolation is applied to produce a fused i-vector $\boldsymbol{w}^f(u)$ of utterance u as follows,

$$\boldsymbol{w}^f(u) = (1-\alpha)\boldsymbol{w}(u) + \alpha\boldsymbol{w}^c(u), \;\; 0 \leq \alpha \leq 1, \quad (9)$$

where $\boldsymbol{w}$ is an i-vector, and $\boldsymbol{w}^c$ is its compensated i-vector.

Another approach is a score level fusion that combines scores calculated by speaker recognition systems. For example, the fused score $s^f(u)$ of utterance u can be calculated as follows,

$$s^f(u) = (1-\alpha)s(\boldsymbol{w}(u)) + \alpha s(\boldsymbol{w}^c(u)), \;\; 0 \leq \alpha \leq 1, \quad (10)$$

where $s()$ is a scoring function of a speaker recognition system with calibration. The interpolation weight and calibration parameters in this equation can be optimized with supervised training [27].

4. Evaluation

4.1. Experimental Setup

We experimentally evaluated the performance of our method in a speaker verification task of the 2008 NIST SRE [1]. In the experiment, we used the "short2-10sec" condition as a trial set. In the trial, the enrollment data were a collection of conversational speech of approximately five minutes duration, and the testing data were a set of approximately 10 seconds of speech segments. Performance measures for the evaluation were the equal error rate (EER) and the minimum detection cost function (minDCF) on the trial calculated with the BOSARIS toolkit [2].

The speaker verification system in the experiments was based on an i-vector and PLDA framework described in Section 2. In the system, the input speech segment was first converted to a time series of acoustic feature vectors, each of which consist of 60 features (MFCC 1-20 and its Δ and $\Delta\Delta$) extracted from a frame of 25ms width every 10ms. Then, an i-vector of 400 dimensions was extracted as a speaker feature from the acoustic features, using a Gaussian mixture model with 2,048 mixture components as a universal background model (UBM) and a total variability matrix (TVM). Linear discriminant analysis (LDA) and length normalization [28] were applied to the i-vector, and finally an i-vector of 150 dimensions was evaluated in the PLDA model. The UBM, TVM, LDA, and PLDA models were trained with development data that were different from the data in the trial list. The development data were a combination of the SRE 2004 and the Fisher corpus, and they contained 5,444 speakers in total. We utilized the Kaldi speech recognition toolkit [3] [29] to run these steps.

Neural networks with 1 hidden layers were used as the denoising autoencoder (DAE) for our method. The input and output layers had 182 nodes to accept the i-vectors (150-dim) and phonetic vectors of 32 dimensions. The phonetic vector was also calculated for each speech segment based on a Gaussian mixture model (32 mixtures) that was trained with the development set. The one hidden layer had 200 nodes.

In the training phase, the DAE was optimized in two steps. In the first step, we used the entirety of the enrollment set of 1,270 speakers to train a speaker-independent DAE. Each enrollment segment was divided into short segments of 10 seconds. Then, the segments were converted to i-vectors that were

[1] http://www.itl.nist.gov/iad/mig/tests/sre/2008/
[2] http://sites.google.com/site/bosaristoolkit/
[3] http://sourceforge.net/projects/kaldi/

Table 1: *Equal error rates (EER, %) and minimum detection cost function (minDCF, in parentheses). All values were measured on condition 6 (telephone training and test) of the "short2-10sec" male trial list. Our DAE-based compensation outperformed the baseline when the two systems were fused in score level.*

System	EER(minDCF)
(a) Baseline	6.6 (0.362)
(b) DAE w/o phonetic vector	10.3 (0.481)
+ Baseline	6.4 (0.265)
(c) DAE w/ phonetic vector	9.3 (0.448)
+ Baseline	**4.1** (**0.196**)

regarded as corrupted examples of the enrollment i-vectors. The total number of pairs in the training set was 30,273. In the second step, the speaker-independent DAE was trained again with a small number of i-vector pairs (23.8 pairs in average) for each enrollment speaker to produce a speaker-dependent DAE. In both steps, its objective function was the mean square reconstruction error. Other parameters in training were configured as follows: corruption on the input layer was the binomial function that randomly applied zero to 20% of the nodes, the learning rate was set to 0.001, and the number of iterations was 40 and 10 in the first and second steps, respectively. We utilized the pylearn2 [4] machine learning library for implementation.

In the testing phase, the i-vector (150-dim) and phonetic vector (32-dim) were extracted from a testing speech segment in the trial list. Then, the speaker-dependent DAE of the claimed speaker transformed them and produced its compensated i-vector $\boldsymbol{w}^c$ of 150 dimensions. Moreover, two versions of fusion, Eq. (9) and (10), were applied to the input i-vector and the compensated i-vector. The weights for the score fusion were determined using 2-fold cross validation.

Note that the whole enrollment set is used in training of the speaker-independent DAE in our experiments. Use of information about the other target speakers is not allowed in the SRE 2008 regulations, although we think our approach would be reasonable in practical situations.

4.2. Experimental Results

Table 1 shows EERs and minDCFs of the series of speaker verification experiments. The EER with our DAE-based system (c) was 4.1%, and it outperformed the 6.6% of the baseline i-vector PLDA system (a). The reduction rate of the EER was 37.9% for this case. Although our method did not show improvement when used as a single system, it achieved better values when fused with the baseline system in the score level ("+ Baseline"). Figure 3 shows the EERs of the i-vector fusion systems of Eq. (9) with respect to the weight α of the compensated i-vector. The fused system performed slightly better than the baseline when α was from 0.1 to 0.3. These results indicate that the DAE-based compensation method provides supplemental speaker characteristics not included in the original i-vector. However, at the same time, it might lack some features to distinguish speakers. Thus, further improvement in the compensation procedure would help improve speaker recognition.

Table 1 also shows that the DAE-based system without phonetic vector (b) improved a little. This result means that the phonetic vector is helpful in improving the performance of feature

[4] http://deeplearning.net/software/pylearn2/

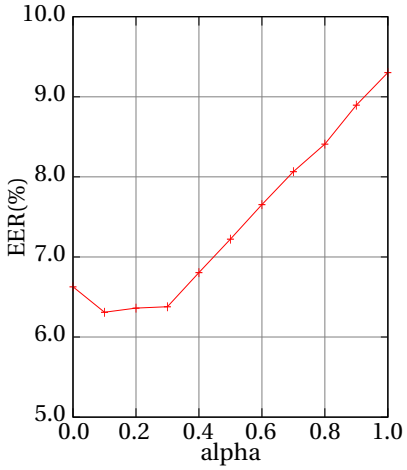

Figure 3: *Equal error rates (EER, %) of i-vector level fusion of original i-vector ($\alpha = 0$) and its compensated i-vector ($\alpha = 1$). The fused system outperformed the baseline.*

restoration. In this way, the speaker recognition system should perform better with our method than with the baseline i-vector PLDA system.

5. Conclusions and Future Work

In this paper, we described, for text-independent speaker recognition of short utterances, a speaker feature restoration method that utilizes a denoising autoencoder (DAE) to compensate the phonetic imbalance in a short utterance. The DAE transforms an i-vector into its canonical i-vector which could be obtained when its phonetic distribution is balanced. The phonetic vector that represents phonetic distribution in an utterance is also given to the DAE. The effectiveness of the method was demonstrated in a series of speaker verification experiments based on the NIST SRE task. It achieved a 37.9% reduction in equal error rates over those for a case without compensation.

Our future work is to extend the DAE modeling, e.g., deep structure, to improve the i-vector restoration, and we are also considering evaluating the effectiveness of our method under various duration-mismatch conditions.

6. Acknowledgements

The authors would like to thank Prof. Koichi Shinoda and Johan Rohdin of the Tokyo Institute of Technology for their helpful advice and comments.

7. References

[1] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

[2] N. Dehak, P. 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.

[3] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. of ICCV*, 2007.

[4] C. S. Greenberg, D. Bansé, G. R. Doddington, D. Garcia-Romero, J. J. Godfrey, T. Kinnunen, A. F. Martin, A. McCree, M. Przybocki, D. A. Raynolds, "The NIST 2014 speaker recognition i-vector machine learning challenge," in *Proc. of Odyssey: Speaker and Language Recognition Workshop*, pp. 224–230, 2014.

[5] B. Fauve, N. Evans, and J. Mason, "Improving the performance of text-independent short duration SVM-and GMM-based speaker verification," in *Proc. of Odyssey: Speaker and Language Recognition Workshop*, 2008.

[6] R. Vogt, B. Baker, and S. Sridharan, "Factor analysis subspace estimation for speaker verification with short utterances," in *Proc. of Interspeech*, pp. 853–856, 2008.

[7] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "i-vector based speaker recognition on short utterances," in *Proc. of Interspeech*, pp. 2341–2344, 2011.

[8] V. Hautamäki, Y.-C. Cheng, P. Rajan, and C.-H. Lee, "Minimax i-vector extractor for short duration speaker verification," in *Proc. of Interspeech*, pp. 3708–3712, 2013.

[9] A. Kanagasundaram, D. Dean, S. Sridharan, J. Gonzalez-Dominguez, J. Gonzalez-Rodriguez, and D. Ramos, "Improving short utterance i-vector speaker verification using utterance variance modelling and compensation techniques," *Speech Communication*, vol. 59, pp. 69–82, 2014.

[10] B. Vesnicer, J. Žganec-Gros, S. Dobrišek, and V. Štruc, "Incorporating duration information into i-vector-based speaker recognition systems," in *Proc. of Odyssey Speaker and Language Recognition Workshop*, pp. 241–248, 2014.

[11] A. K. Sarkar, D. Matrouf, P. M. Bousquet, and J. F. Bonastre, "Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification," in *Proc. of Interspeech*, pp. 2662–2665, 2012.

[12] T. Hasan, R. Saeidi, J. H. L. Hansen, and D. A. van Leeuwen, "Duration mismatch compensation for i-vector based speaker recognition systems," in *Proc. of ICASSP*, pp. 7663–7667, 2013.

[13] P. Kenny, T. Stafylakis, P. Ouellet, Md. J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Proc. of ICASSP*, pp. 7649–7653, 2013.

[14] M. I. Mandasari, R. Saeidi, M. McLaren, and D. A. van Leeuwen, "Quality measure functions for calibration of speaker recognition systems in various duration conditions," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 11, pp. 2425–2438, 2013.

[15] A. Nautsch, C. Rathgeby, C. Buschy, H. Reininger, and K. Kasperz, "Towards duration invariance of i-vector-based adaptive score normalization," in *Proc. of Odyssey Speaker and Language Recognition Workshop*, pp. 60–67, 2014.

[16] D. E. Sturim, D. A. Reynolds, R. B. Dunn, and T. F. Quatieri, "Speaker verification using text-constrained Gaussian mixture models," in *Proc. of ICASSP*, pp. 677–680, 2002.

[17] T. Bocklet and E. Shriberg, "Speaker recognition using syllable-based constraints for cepstral frame selection," in *Proc. of ICASSP*, pp. 4525–4528, 2009.

[18] A. Larcher, P.-M. Bousquet, K. A. Lee, D. Matrouf, H. Li, and J.-F. Bonastre, "I-vectors in the context of phonetically-constrained short utterances for speaker verification," in *Proc. of ICASSP*, pp. 4773–4776, 2012.

[19] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances," in *Proc. of. ICASSP*, pp. 7673–7677, 2013.

[20] N. Scheffer and Y. Lei, "Content matching for short duration speaker recognition," in *Proc. of Interspeech*, pp.1317–1321, 2014.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. of NIPS*, 2012.

[22] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[23] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. of ICML*, pp. 1096–1103, 2008.

[24] X. Lu, Y. Tsao, S. Matsuda, C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. of Interspeech*, pp. 436–440, 2013.

[25] T. Ishii, H. Komiyama, T. Shinozaki, Y. Horiuchi, and S. Kuroiwa, "Reverberant speech recognition based on denoising autoencoder," in *Proc. of Interspeech*, pp. 3512–3516, 2013.

[26] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[27] N. Brümmer and E. de Villiers, "The BOSARIS toolkit user guide: Theory, algorithms and code for binary classifier score processing," 2011.

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

[29] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. of Workshop on Automatic Speech Recognition and Understanding*, 2011.