EXPLORING SEQUENTIAL CHARACTERISTICS IN SPEAKER BOTTLENECK FEATURE FOR TEXT-DEPENDENT SPEAKER VERIFICATION

Liping Chen[1]*, Yong Zhao*[2]*, Shi-Xiong Zhang*[2]*, Jie Li*[1]*, Guoli Ye*[2]*,Frank Soong*[3]

[1] Microsoft Search Technology Center Asia, Beijing
[2]Microsoft Corporation, One Microsoft Way, Redmond, WA 98052
[3] Microsoft Research Asia

{lipch,yonzhao,zhashi,jliie,guoye,frankkps}@microsoft.com

ABSTRACT

In this paper, given the speaker bottleneck feature vectors extracted with speaker discriminant neural networks, we focus on using the sequential speaker characteristics for text-dependent speaker verification. In each evaluation trial, speaker supervectors are used as the representations of the sequential speaker characteristics rendered in the compared speech utterances. To this end, dynamic time warping is used to warp the variable-length speaker feature vector sequences of the utterances to the same length. Thereafter for every utterance, a speaker supervector can be obtained as the concatenation of its speaker feature vectors. We use Euclidean distance and support vector machine (SVM) to compute the decision score on the speaker supervectors. Our experiments on a Microsoft internal keyword-spotting database showed the effectiveness of the proposed speaker supervector for text-dependent speaker verification. Moreover, when SVM backend was used in scoring, the speaker supervector achieved the best EER performance 1.627%, better than the combination of i-vector and probabilistic linear discriminant analysis.

Index Terms— Text-dependent speaker verification, sequential speaker characteristics, speaker supervector, dynamic time warping

1. INTRODUCTION

As speaker verification is becoming more and more commercialized, the demand for speaker verification systems on short utterances is increasing. The inability of text-independent speaker verification methods to attain acceptable performance with utterances of short durations revived the application interests for text-dependent speaker verification. To name a few, Microsoft, Google and Apple have released their speaker verification products with the lexical contents of the speech utterances to be "Hey Cortana", "OK Google" and "Hey Siri" respectively. In such application scenarios, in order to obtain enough speech for acceptable performance, multiple utterances are always used for speaker enrollment. Previously, the methods in text-independent speaker recognition have been adjusted to the text-dependent scenario successfully, including *Gaussian mixture model – Universal background model* (GMM–UBM) [1, 2, 3], *joint factor analysis* (JFA) [4, 5, 6] and the combination of i-vector and *probabilistic linear discriminant analysis*(PLDA) [7, 8, 9, 10].

Recently, *neural networks* (NNs) trained for speaker discrimination have been explored in text-dependent speaker verification [11, 12, 13, 14]. In [11] and [12], the neural networks were trained to classify among the training speakers. In [13] and [14], the networks were trained in an end-to-end manner which inferred whether the speakers in the input utterance pairs were from the same or different speakers. Basically, whether the network is trained to classify among the predefined speaker set [11, 12] or to distinguish between the hypotheses of whether the input utterances are from the same speaker or not [13, 14], the hidden layers are endorsed with the capability of speaker discriminative information extraction. As such, usually speaker feature vectors can be extracted as the output of a bottleneck layer in a speaker discriminant neural network. How to make use of the speaker feature vectors for speaker verification forms an interesting issue that deserves further research. In [11], given the feature sequence extracted from a specific speech utterance, the outputs of the last hidden layer were averaged to be the so-called *d-vector*. Like i-vector, the d-vector contributes a kind of fixed-length representation of the speaker characteristics rendered in a speech utterance.

The models of speaker characteristics aforementioned are built on the accumulated statistics on the acoustic and speaker feature vectors while omitting the sequential correlation among the speech frames. Unlike them, in earlier researches, the sequential correlation was modeled and used in text-dependent speaker verification such as *dynamic time warping* (DTW) [15] and *hidden Markov model*(HMM) [16, 17]. Among others, DTW [18] provided an efficient and effective speaker comparison in a sequential manner. Nevertheless, the feature used in [18] was the acoustic feature which is rich in phonetic information, such as *perceptual linear prediction* (PLP) and *Mel-frequency cepstral coefficients* (MFCC). In such a manner, the alignment and speaker comparison in DTW was dominated by phonetic information. Although it has been widely acknowledged that the speaker comparison should be phone-related, template matching on acoustic feature still suffers from the vulnerability brought by the pronunciation and articulation mismatch between the compared speech utterances. Moreover, when the data conditions of the speech utterances are complex, e.g., variant channels and noise are involved, due to the lack of channel compensation techniques, the performance of the acoustic feature vectors will be unacceptable.

In this paper, we further pursue how to better make use of the speaker feature vectors extracted with NNs for text-dependent speaker verification. In an evaluation trial, given the sequences of speaker feature vectors extracted from the enrollment and test speech utterances, we firstly resort to DTW to align them to the same length. Then, a speaker supervector, which is the concatenation of the speaker feature vectors for each utterance, is used as the representation of the sequential speaker characterisitcs rendered in it. Compared with d-vector, the advantage of the speaker supervector is that the sequential correlation among the frames can be exploited

for speaker comparison. Besides, compared to the acoustic features as used in [18], the advantages of the speaker supervector lies in two aspects. Firstly, the alignment and speaker comparison between the enrollment and test utterances is dominated by the speaker characteristics instead of the pronunciation and articulation. Secondly, since the speaker discriminant neural network which is used for speaker feature extraction sees variant channel and noise conditions rendered in the training utterances, the speaker feature vectors can be robust to channel and noise variation. As for the testing phase, we apply two scoring methods on the speaker supervectors. One is to use the scaled Euclidean distance directly as the final decision score. The other is to use the support vector machine (SVM) for discriminative scoring [19]. We carried out our experiments on a Microsoft internal keyword-spotting database with the keyword "Hey Cortana". Six utterances were used for speaker enrollment and one for testing. The results showed that the proposed speaker supervector could achieve better performance than d-vector; and the sequential modeling on speaker feature was better than acoustic feature in speaker comparison. Moreover, when combined with SVM as backend, the speaker supervector outperformed the i-vector/PLDA cascade.

The rest of this paper is organized as follows. Section 2 gives a brief description of the existing vectors for speaker characteristic representation. In Section 3, we propose the speaker supervector. The experimental results are presented in Section 4. We reach our conclusions and describe the future work in Section 5.

2. PRIOR WORKS

2.1. I-vector

Given the feature sequence extracted from an utterance $\mathcal{O}$, its mean supervector $\mathbf{m}$ can be modeled as follows

$$\mathbf{m} = \mathbf{m}_0 + \mathbf{T}\mathbf{w} \tag{1}$$

where $\mathbf{T}$ is the total variability matrix and $\mathbf{w}$ is the latent variable whose prior distribution is a standard normal distribution as $\mathbf{w}_r \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. $\mathbf{m}_0$ is the mean supervector of the UBM. The i-vector of $\mathcal{O}$ is estimated as the posterior mean of $\mathbf{w}$ which provides a fixed-length and always reduced-dimension representation for the speech utterance $\mathcal{O}$.

2.2. D-vector

Fig. 1 illustrates the graphical model of the neural network for speaker feature extraction. The neural network always takes acoustic feature vectors as input. The nodes in the output layer represent the set of speakers in the training set, denoted as $\{\text{spk}_1, \text{spk}_2, ..., \text{spk}_N\}$ where N is the number of training speakers. The neural network is trained to classify among the N speakers with the cross-entropy loss function. The last hidden layer is a bottleneck layer whose dimension is always smaller than the others, denoted as $\mathbf{b}$ in Fig. 1. Given an input frame, the output of the bottleneck layer is estimated as the representation of its speaker information, named as speaker feature vector in this paper. The hidden layers of the neural network can be full connection layers, *long short-term memory* (LSTM) cells and convolutional layers, resulting in DNN, *recurrent neural network* (RNN) and *convolutional neural network* (CNN) respectively.

In [11], the neural network is specified to be DNN. Given the sequence of acoustic feature vectors extracted from a speech utterance $\mathcal{O}$, the speaker feature vectors $\mathcal{B} = \bigcup_{t=1}^{T} \mathbf{b}_t$ can be

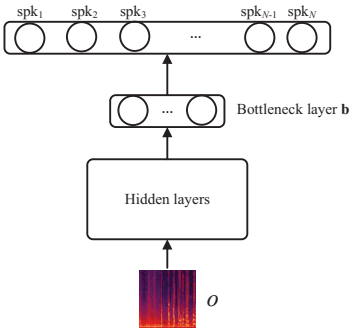

Fig. 1. Paradigm of neural networks for speaker classification, where o is the input observation; N is the number of training speakers and $\mathbf{b}$ is the bottleneck layer.

estimated where T is be the number of frames. The sequence of speaker feature vectors are then averaged to be the d-vector as a compact representation of the speaker characteristics rendered in $\mathcal{O}$. Note that from the perspective of modeling, the d-vector can be seen as the statistical mean vector taken from a Gaussian which models the speaker characteristics distribution in $\mathcal{O}$.

3. SPEAKER SUPERVECTOR

In this section, based on the speaker feature vectors extracted with the neural network illustrated in Fig. 1, we describe how to obtain the speaker supervectors for the utterances in an evaluation trial and how to score them for speaker similarity measurement.

3.1. Speaker supervector

For simplicity, in this subsection, we focus on the scenario where only one utterance is used for speaker enrollment. Denote the two speech utterances for identity claim as $\mathcal{O}_1$ and $\mathcal{O}_2$ and the speaker feature vectors extracted from them as $\mathcal{B}_1 = \bigcup_{t=1}^{T_1} \mathbf{b}_1(t)$ and $\mathcal{B}_2 = \bigcup_{t=1}^{T_2} \mathbf{b}_2(t)$, with T_1 and T_2 to be the number of frames respectively. Choosing $\mathcal{B}_1$ to be the template, the best warping function that warps the axis from $\mathcal{B}_1$ to $\mathcal{B}_2$ is the one that results in the smallest distance p^* which is mathematically defined as:

$$p^* = \min_{\{w(t)\}} \sum_{t=1}^{T_1} E(\mathbf{b}_1(t), \mathbf{b}_2(w(t))) \tag{2}$$

Here, E is the distance function, specified to be Euclidean distance in the paper. $m_t = w(t)$ is the warping function on the axises from $\mathcal{B}_1$ to $\mathcal{B}_2$, resulting in the warped $\mathcal{B}_2$ to be as follows,

$$\mathcal{B}_2 \leftarrow \bigcup_{t=1}^{T_1} \mathbf{b}_2(m_t) \tag{3}$$

The warped $\mathcal{B}_2$ is in the same length as the template $\mathcal{B}_1$, i.e., T_1.

The speaker feature vectors in $\mathcal{B}_1$ and the warped $\mathcal{B}_2$ are then concatenated respectively to be the speaker supervectors, i.e., $\mathbf{v}_1 = \left[(\mathbf{b}_1(1))^\mathsf{T}, (\mathbf{b}_1(2))^\mathsf{T}, ..., (\mathbf{b}_1(T_1))^\mathsf{T}\right]^\mathsf{T}$ and $\mathbf{v}_2 = \left[(\mathbf{b}_2(m_1))^\mathsf{T}, (\mathbf{b}_2(m_2))^\mathsf{T}, ..., (\mathbf{b}_2(m_{T_1}))^\mathsf{T}\right]^\mathsf{T}$, representing the sequential speaker characteristics rendered in $\mathcal{O}_1$ and $\mathcal{O}_2$. It's worth

pointing out that in our work, the speaker supervectors here are obtained among the compared utterances in individual evaluation trials. So different from our common sense in the GMM–UBM framework where the Gussian mean supervectors are of fixed length for all utterances, the lengths of the speaker supervectors here are the same within each trial but always different among different trials.

Note that, we use the linear output of the bottleneck layer as our speaker feature vector. Before being aligned by DTW, the speaker feature vectors are normalized with rank ordering within each frame [20]. To be specific, a speaker feature vector $\mathbf{d}$ can be specified to its dimensions as $\mathbf{b} = [b_1, ..., b_D]^{\mathsf{T}}$ with D to be its dimensionality. The D elements are firstly ranked with the rank order of the d-th element to be R_d. The normalized value for b_d will be:

$$b_d \leftarrow \frac{D + \frac{1}{2} - R_d}{D} \tag{4}$$

where $d = 1, ..., D$.

3.2. Scoring

Given the N ($N \geq 1$) enrollment utterances and the test utterance in an evaluation trial, denote the sequences of speaker feature vectors extracted from them as $\bigcup_{n=1}^{N} \mathcal{B}_n^{\mathrm{e}}$ and $\mathcal{B}^{\mathrm{t}}$ where the superscripts $^{\mathrm{e}}$ and $^{\mathrm{t}}$ stands for enrollment and test separately. One utterance is randomly chosen from the enrollment utterances to be the template, which has T frames. The rest enrollment and test sequences are then warped to the template. Via concatenation, the speaker supervectors for the enrollment and test utterances can be obtained to be $\bigcup_{n=1}^{N} \mathbf{v}_n^{\mathrm{e}}$ and $\mathbf{v}^{\mathrm{t}}$ respectively. Two scoring methods are then applied for decision score computation, i.e. Euclidean distance scoring and SVM backend.

In Euclidean distance scoring, the enrollment supervectors are averaged to be the speaker supervector of the enrollment speaker as $\mathbf{v}^{\mathrm{e}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{v}_n^{\mathrm{e}}$. The Euclidean distance between $\mathbf{v}^{\mathrm{e}}$ and $\mathbf{v}^{\mathrm{t}}$ is computed and then scaled by the number of frames T to be the final decision score as follows:

$$s = \frac{\left\|\mathbf{v}^{\mathrm{e}} - \mathbf{v}^{\mathrm{t}}\right\|_2}{T} \tag{5}$$

On the speaker supervectors, SVM can be applied as the backend for discriminative speaker modeling and scoring. Firstly, an SVM is trained using the enrollment supervectors $\bigcup_{n=1}^{N} \mathbf{v}_n^{\mathrm{e}}$ to be the enrollment speaker model with the discriminant function to be $s = f_{\mathrm{e}}(\mathbf{v}; \theta_{\mathrm{e}})$ where θ_{e} denotes the set of model parameters. Given the test supervector $\mathbf{v}^{\mathrm{t}}$, $s = f_{\mathrm{e}}\left(\mathbf{v}^{\mathrm{t}}; \theta_{\mathrm{e}}\right)$ is computed to be the final decision score. See [19] for more details about SVM.

4. EXPERIMENTS

4.1. Experimental setup

Our experiments were carried out on the Microsoft keyword-spotting live speech dataset. The speech segments of the text "Hey Cortana" were identified and separated from the speech utterances using a keyword spotter. As for the keyword spotter, we used a DNN for acoustic modeling. The input layer was the current frame spliced with 2 frames on its left and right sides respectively. The DNN was trained to discriminate among the 9 monophones in "Hey Cortana". Together with a node for silence, there were 10 nodes in the output layer, leading the structure of the DNN to be $190(5 \times 38) - 512 - 128 - 64 - 10$. The first subgraph in Fig. 2 illustrates the duration distribution of the keyword speech segments

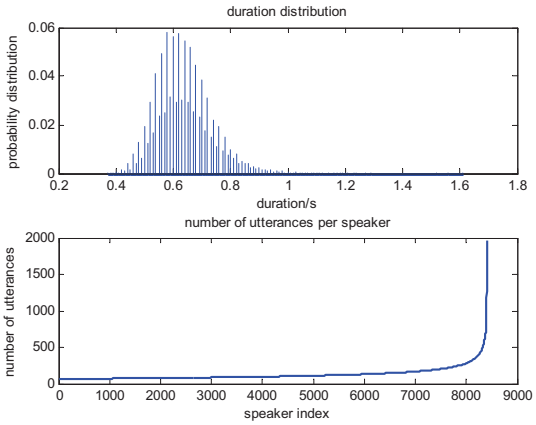

Fig. 2. Duration distribution of the keyword speech segments and the number of utterances per speaker in the training subset

detected by the keyword spotter. The average duration per segment was about 0.6 seconds. A training and an evaluation sets were built exclusively to make sure that the speakers in the two subsets were not overlapped. The training set was composed of 193 hours of utterances from $8,404$ speakers with at least 70 utterances for each speaker. The second subgraph in Fig. 2 presents the number of utterances per speaker in the training subset. The number of utterances per speaker ranges from 70 to $1,968$. Among them, $8,320$ speakers have less than 500 utterances. In evaluation, $1,000$ speakers were selected as the enrollment speakers. 6 utterances were used for speaker enrollment and 1 was used for testing in each verification trial. On average, 3 target and 17 nontarget trials were composed for every enrollment speaker, resulting in $2,987$ target and $16,036$ nontarget trials in total. The *equal error rate* (EER) and *detection cost function* (DCF) [21] were used as the performance criteria. For DCF, we considered the operation points of NIST SRE'08, SRE'10 and SRE'12.

The acoustic feature we used was 13-dimensional MFCC static coefficients appended with the first and second derivatives. After appending, C_0 was removed, leaving the 38-dimensional raw MFCC feature vectors. However, according to the results reported in [22], in order to achieve the new state-of-the-art performances on the systems that were to be compared in our experiments, the raw MFCC feature vector was concatenated with its corresponding phonetic bottleneck vector. The phonetic bottleneck vector was extracted with the DNN used in our keyword spotter. The linear output of the last hidden layer of dimension 64 was estimated for each frame and concatenated with its MFCC feature vector. Thereafter, *principal component analysis* (PCA) [23] was exerted on the 102-dimensional concatenated vectors to reduce its dimension to 72. The dimensionality of 72 was chosen from a set of experiments as it gave the best performance. In the following, the 72-dimensional combination of MFCC and phonetic bottleneck vector is referred to BN–MFCC for brevity.

4.2. Speaker supervector

The DNN for speaker feature extraction was trained on the 193-hour training dataset using the 38-dimensional raw MFCC feature. Every frame was spliced with its contextual 20 frames on both left and right sides as the input to the DNN. There were $8,404$ nodes in the output

softmax layer, each representing a training speaker. There were 4 hidden layers of $1,024$ nodes, succeeded by a bottleneck layer of 80 nodes. Above all, the structure of the network was $1558\,(41 \times 38) - 1024 - 1024 - 1024 - 1024 - 80 - 8404$. 80 was chosen since it gave the best performance in our experiments regarding the size of the bottleneck layer. In the warping between the speaker vector sequences, the maximum and minimum slopes were 2 and $1/2$.

The performance of the speaker supervector was compared with two systems. The first one was d-vector system where the 6 d-vectors of enrollment utterances were averaged to represent the enrollment speaker and its Euclidean distance with the test d-vector was used as the score. The other system is similar with speaker supervector, except for the feature vectors. In this experiment, we replaced the speaker feature vectors with the 72-dimensional BN–MFCC feature vectors. Following the same operations that constructed speaker supervectors, we obtained BN–MFCC supervectors from the BN–MFCC feature vectors for the enrollment and test utterances in an evaluation trial. Euclidean distance as described in Section 3.2 was used for decision score computation on the two kinds of supervectors. It's worthy to mention that, in our experiments, we found that rank ordering normalization on the speaker and BN–MFCC feature vectors as described in Section 3.1 was crucial to the performances of the corresponding systems. Hence, rank ordering normalization was exerted on the speaker and BN–MFCC feature vectors before they were averaged to d-vector or warped to supervector. The results of the three vectors, i.e., d-vector, BN–MFCC and speaker supervectors, are presented in Table 1.

Table 1. Performance comparison of d-vector, BN–MFCC supervector and speaker supervector

		EER(%)	DCF08	DCF10	DCF12
	d-vector	2.099	0.102	0.407	0.312
supervector	BN–MFCC	3.407	0.134	0.402	0.310
	speaker	**1.893**	**0.098**	**0.380**	**0.293**

From the results, we can see that the proposed speaker supervector definitely achieved the best performance among the three methods. In terms of EER, it was 9.7% and 44.4% relatively better than d-vector and BN–MFCC supervector respectively. The superiority of speaker supervector over d-vector informs us that the sequential speaker information within the speech utterances is a part of speaker characteristics and can be used for better speaker verification performance. Meanwhile, the better performance achieved by speaker supervector than BN–MFCC supervector infers us that for sequential speaker characteristic modeling and comparison, speaker feature is better than BN–MFCC. According to our analysis, the reason for such a superiority lies three aspects. Firstly, the alignment between the compared speech utterances is dominated by speaker information. Secondly, the similarity score is computed on the features vectors representing speaker characteristics. Thirdly, since the speaker feature vector was extracted with the DNN trained on a large scale of speech utterances, it's more robust to channel variation and noise in our live dataset than the MFCC feature.

4.3. SVM backend

We further experimented on SVM as the backend on the speaker supervector. Given the multiple enrollment speaker supervectors in an evaluation trial, an SVM was trained as the enrollment speaker model using the libsvm toolkit [24]. Linear kernel was used in the SVM. $1,200$ utterances from the speakers that didn't appear in evaluation were used as imposters in SVM training.

In addition, the performances of the conventional GMM–UBM and i-vetcor/PLDA cascade were also compared. The BN–MFCC feature vector was used in the two experiments. The UBM was composed of 128 Gaussian components with diagonal covariance matrices. The rank of the total variability matrix was 400. In the following PLDA, the speaker and channel subspaces were of ranks 200 and 400 respectively and the residual covariance matrix was diagonal. In order to keep a balance among the training speakers and also be time-efficient, 70 utterances were randomly chosen per speaker from the $8,404$ training speakers to train the UBM, total variability matrix and PLDA. In our multi-session scenario, the statistics computed on the 6 enrollment utterances were accumulated for the adaptation from UBM to speaker-dependent GMM. In the i-vector system, an i-vector was estimated for each of the 6 enrollment utterances and their mean was then taken to be the i-vector of the enrollment speaker [25].

The performances of the Euclidean distance and SVM backend on speaker supervectors are given in Table 2 together with the GMM–UBM and i-vector/PLDA. From the results, we can see that when Euclidean distance is used for scoring, the speaker supervector outperformed GMM–UBM. For the speaker supervector, the SVM backend achieved better performance than Euclidean distance, showing the effectiveness of the discriminative model for speaker modeling and scoring. Moreover, the combination of speaker supervector and SVM achieved 4.5% relative gain in EER compared with the i-vector/PLDA cascade, validating it to be an effective method for text-dependent speaker verification.

Table 2. Performance comparisons among GMM–UBM, i-vector/PLDA and the Euclidean and SVM scorings on speaker supervector

		EER(%)	DCF08	DCF10	DCF12
GMM–UBM		2.445	0.010	0.363	0.296
i-vector/PLDA		1.737	0.086	0.396	0.287
speaker	Euclidean	1.893	0.098	0.380	0.293
supervector	SVM	**1.627**	**0.073**	**0.246**	**0.191**

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed the speaker supervector to represent the sequential speaker characteristics rendered in a speech utterance for text-dependent speaker verification. To this end, the deep neural network trained for speaker classification was used as speaker feature extractor and dynamic time warping was used to map the variable-length sequences to the same length. Our experiments conducted in the Microsoft internal keyword spotting dataset validated the superiorities of the sequential modeling on speaker feature vectors over both the statistical mean on the speaker vectors and the sequential modeling on acoustic feature vectors. Moreover, using support vector machine for backend speaker modeling and scoring on the speaker supervectors, we can obtain better performance than the i-vector/PLDA cascade.

Besides the proposed speaker supervector, it's still an interesting topic to see how other successful models on acoustic feature can be applied on the speaker feature vectors, such as GMM–UBM, i-vector/PLDA and another sequential model GMM–HMM. This is will be a point for future research.

6. REFERENCES

[1] A. Larcher, K.A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[2] A. Larcher, K.A. Lee, B. Ma, and H. Li, "Imposture classification for text-dependent speaker verification," in *Proc. ICASSP*, 2014, pp. 739–743.

[3] A. Larcher, K.A. Lee, B. Ma, and H. Li, "Modelling the alternative hypothesis for text-dependent speaker verification," in *Proc. ICASSP*, 2014, pp. 734–738.

[4] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1448–1460, 2007.

[5] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

[6] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, 2005.

[7] P. Kenny, T. Stafylakis, J. Alam, and J. Kockmann, "JFA modeling with left-to-right structure and a new backend for text-dependent speaker recognition," in *Proc. ICASSP*, 2015, pp. 4689–4693.

[8] T. Stafylakis, P. Kenny, M. J. Alam, and M. Kockmann, "Speaker and channel factors in text-dependent speaker recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 1, pp. 65–78, 2016.

[9] T. Stafylakis, P. Kenny, P. Ouellet, J. Perez, M. Kockmann, and P. Dumouchel, "Text-dependent speaker recognition using PLDA with uncertainty propagation," in *Proc. Interspeech*, 2013, pp. 3684–3688.

[10] L. Chen, K.A. Lee, B. Ma, W. Guo, H. Li, and L.-R. Dai, "Phone-centric local variability vector for text-constrained speaker verification.," in *Proc. Interspeech*, 2015, pp. 229–233.

[11] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, 2014, pp. 4052–4056.

[12] G. Bhattacharya, J. Alam, T. Stafylakis, and P. Kenny, "Deep neural network based text-dependent speaker recognition: preliminary results," in *Proc. Odyssey: Speaker and Language Recognition Workshop*, 2016.

[13] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," *arXiv preprint arXiv:1701.00562*, 2017.

[14] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. ICASSP*, 2016, pp. 5115–5119.

[15] R. Bellman, "On the theory of dynamic programming," *Proceedings of the National Academy of Sciences*, vol. 38, no. 8, pp. 716–719, 1952.

[16] A. Poritz, "Linear predictive hidden markov models and the speech signal," in *Proc. ICASSP*, 1982, vol. 7, pp. 1291–1294.

[17] N. Z. Tisby, "On the application of mixture ar hidden markov models to text independent speaker recognition," *Transactions on Signal Processing*, vol. 39, no. 3, pp. 563–570, 1991.

[18] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[19] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, Wiley, New York, 1973.

[20] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Odyssey:The Speaker and Language Recognition Workshop*, 2001, pp. 213–218.

[21] N. Brümmer and J. du Preez, "Application-independent evaluation of speaker detection," *Computer & Speech Language*, vol. 20, no. 2, pp. 230–275, 2006.

[22] P. Matějka, O. Glembek, O. Novotnỳ, O. Plchot, F. Grézl, L. Burget, and J. H. Cernockỳ, "Analysis of DNN approaches to speaker identification," in *Proc. ICASSP*, 2016, pp. 5100–5104.

[23] C. Bishop, *Pattern recognition and machine learning*, Springer, New York, 2006.

[24] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 27, 2011.

[25] L. Chen, K. A. Lee, B. Ma, W. Guo, H. Li, and L.-R. Dai, "Minimum divergence estimation of speaker prior in multi-session PLDA scoring," in *Proc. ICASSP*, 2014, pp. 4007–4011.