A Segmental DNN/i-vector Approach for Digit-Prompted Speaker Verification

Jie Yan*, Xie Lei*‡, Guangsen Wang† and Zhong-Hua Fu*
* Northwestern Polytechnical University, Xian, China
E-mail: {jyan,lxie}@nwpu-aslp.org, mailfzh@nwpu.edu.cn
† Tencent AI Lab, Shenzhen, China
E-mail: guangsenw@gmail.com

***Abstract*—DNN/i-vectors have achieved state-of-the-art performance in text-independent speaker verification systems. For such systems, the UBM posteriors are replaced with the DNN posteriors when training the i-vector extractor to better model the phonetic space. However, the DNN/i-vector systems have limited success on text-dependent speaker verification systems as the lexical variabilities, which are important for such applications, are suppressed in the utterance-level i-vectors. In this paper, we propose a segmental DNN/i-vector approach for the digit-prompted speaker verification task. Specifically, we segment the utterance into digits and model each digit using an individual DNN/i-vector system. By modeling the variability for each digit independently, we can focus more on the speaker characteristics for each digit. To take into consideration the uncertainties in the DNN posteriors, we propose a confidence measure based weighting method. On the RSR2015 dataset, the proposed approach yields an equal error rate of 3.44%, compared to 5.76% of the baseline utterance-level DNN/i-vector system and 4.54% of the joint factor analysis (JFA) system.**

I. Introduction

There are two major categories of the speaker verification systems, namely text-independent (TI) and the text-dependent (TD). Under the former circumstance, the user is free to speak anything during verification. On the other hand, for TD systems, the user is required to speak the passphrase in each verification. Various TI speaker recognition schemes were proposed under the assumption that the speaker characteristic is independent of the spoken content when the utterances are sufficiently long [1], such as GMM/UBM [2], joint factor analysis (JFA) [3] and the total variability model [4]. The i-vectors [4] derived from the total variability model have been widely used in the TI speaker verification systems and shown to be extremely effective. With the success in speech recognition [5], deep neural networks (DNNs) have also been used in the i-vector framework by replacing the universal background model (UBM) posteriors with the DNN senone posteriors [6] [7] leading to state-of-the-art performance on several benchmarking tasks. The success of such systems (DNN/i-vector) are mainly attributed to the better phonetic space modeling ability of the DNNs.

Although i-vectors have become state-of-the-art technique for TI speaker recognition, they have very limited success for the TD systems because the lexical variabilities which are important for TD systems are suppressed in the utterance-level i-vectors. For TD systems, the passphrase is often in very short duration, where speaker characteristic has shown significant dependency on the lexical content [8, 9, 10]. Therefore, it is essential for TD speaker verification systems to model both the speaker variabilities and the lexical variabilities of the passphrase. In [8], an HMM based approach called HiLAM was proposed, where each speaker-passphrase model was trained as an HMM via adaptation to model the speaker and the passphrase jointly.

As a variant of the TD system, text-prompted speaker verification is more challenging in which user is prompted to provide utterance of random text every time the system is used. Thus we can prevent playback attack. The prompted texts could be random sequences of keywords from a constrained set (e.g., digits). In this paper, we aim to investigate the digit-prompted task on the publicly available RSR2015 data set [8]. A combination of phone adaptation and speaker adaptation was proposed in [11] for text-variable speaker recognition. In the same vein, a phoneme adaption scheme [12] was used within the JFA framework for the digit-prompted speaker verification task. The authors use local vector models a segment in an utterance, while a global vector models the whole utterance. Maximum a posteriori (MAP) adaptation was applied to a phonetic independent UBM to model the speaker and digits jointly. On the same digit prompted task, the authors in [13] advocate the use of HMM for joint modeling of speaker characteristic and lexical content and develop a scoring scheme to differentiate the score contributions of the lexical and speaker components. In [9], the authors show that an i-vector can be decomposed into segments of local variability vectors, each corresponding to a monophone, where each local vector models session variability given the phonetic context.

In this paper, we investigate the DNN/i-vector approach for digit-prompted speaker verification. Inspired by the way of modeling on a smaller granularity acoustic unit in [9] and [12], we model the speaker variabilities on the digit level considering the suppression of lexical components in utterance level i-vectors and the lack of data in phone level local i-vectors. In practice, we segment the training utterances into digits by forced-alignment from an automatic speech recognizer and

‡This work was supported by the National Natural Science Foundation of China (Grant No. 61571363). Corresponding author.

build a DNN/i-vector system for each single digit individually. For verification, we propose a confidence measure(CM) based i-vector weighting scheme to compensate the uncertainty of the ASR for segmenting the digits. Experiments show that our digit local i-vector achieves better results in this task than the previous work.

II. DNN/I-VECTORS EXTRACTION

A. DNN/HMM system for digit segmentation

To train the digit level DNN/i-vector systems, we need the training data for each individual digit. To this end, a hybrid DNN/HMM system [14] used for speech recognition was trained from the raw unsegmented utterances. Specifically, in our digit-prompted task, it is natural to train a simple digit-string recognition system, where each digit is modeled as a J-state HMM and the HMM states for all the digits (10 digits in this paper) are used as the targets for the hybrid DNN/HMM system. The hybrid system is then used to do the forced alignment on the training utterances to segment them into digits. Moreover, the hybrid system can also produce the DNN posteriors for training the DNN/I-vector systems for each individual digit detailed in the following section.

B. I-vector systems from DNN posteriors

A standard GMM-UBM i-vector system [4] is used as one of the baselines in our experiments. Given an utterance, the speaker and channel-dependent GMM supervector is written as follows:

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

where $\mathbf{M}$ is the mean supervector, $\mathbf{m}$ is the mean supervector of UBM. The matrix $\boldsymbol{T}$ is the total variability matrix projecting mean supervector to obtain i-vectors ω . In this model, the *t*-th speech frame $\boldsymbol{x}_t^i$ from the *i*-th speech segment is assumed to be generated by the following distribution:

$$\boldsymbol{x}_t^i \sim \sum_k \gamma_{kt}^i N(\mu_k + \boldsymbol{T}_k \omega, \sum_k) \tag{2}$$

μ_k and $\sum_k$ are the mean and covariance of the *k*-th Gaussian, and γ_{kt}^i are the alignments of $\boldsymbol{x}_t^i$. In general, we represent the alignments by the posterior of the *k*-th Gaussian, given by $\gamma_{kt}^i = p(k|\boldsymbol{x}_t^i)$. The zeroth and first order sufficient statistics used to train the subspace $\boldsymbol{T}$ and extract the i-vector ω can be computed using the posterior probabilities of the classes.

Traditionally, the Gaussian in UBM define the classes *k* in (2) and the posteriors for the classes are computed from the likelihoods of the Gaussians using the Bayes rule [4]. In the DNN/i-vector framework [6] [7], a DNN trained for ASR to discriminate the senones (the HMM states in our system) is used to define the classes *k* instead of the Gaussian in a GMM. The means and covariance of the senone *k*, which is analogous to a Gaussian component in the traditional UBM, can be computed as:

$$\mu_k^i = \frac{\sum_t \gamma_{kt}^i \boldsymbol{x}_t^i}{\sum_t \gamma_{kt}^i} \tag{3}$$

$$\Sigma_k^i = \frac{\sum_t \gamma_{kt}^i \boldsymbol{x}_t^i \boldsymbol{x}_t^{i\,T}}{\sum_t \gamma_{kt}^i} - \mu_k^i \mu_k^{i\,T} \tag{4}$$

where the alignment is represented by the posterior of the DNN corresponding to output unit *k* for frame $\boldsymbol{x}_t^i$. This set of means and covariance are used to compute statistics used in the i-vector extraction. The rest of the process remains the same as in the conventional method, except that the posteriors are always computed using the DNN.

III. SEGMENTAL I-VECTORS FOR DIGIT-PROMPTED SPEAKER VERIFICATION

In either a GMM/i-vector system or a DNN/i-vector system, the i-vector is extracted at the utterance level. That is, the statistics computed for extracting i-vector are accumulated on all the frames of the utterance. If directly applying such utterance level i-vectors to digit-prompted speaker verification, the text constraints imposed on the speaker are not fully exploited. In [9], the authors develop a phone-centric local vector, but it may suffer from the lack of phoneme data. And co-articulation may have influence in local i-vectors modeling. In view of these questions, we model the total variabilities using a finer granularity at the digit level rather than the sentence level. More specifically, we segment the training utterances into digits and a DNN/i-vector system is built for each single digit. We call it segmental DNN/i-vector approach.

A. Digit i-vectors modeling

In practice, we segment all the training data into digits using forced alignment from the ASR system and gather the segments into several groups by the same digit label. A DNN/i-vector system is built for each single digit group. By modeling the variability for each digit independently, we can focus more on the speaker characteristics for each digit. We believe that the $\mathbf{T}$-matrix trained with single digit data can model not only speaker variance but also digit content.

Each of the enrollment utterances is a sequence of randomly generated ten numbers covering all digits from zero to nine. We then extract digit i-vectors for all the segments with the corresponding DNN/i-vector model and average the i-vectors of all the segments corresponding to the same digit. Thus we can get a total of *N* (usually *N*=10) enrollment digit i-vectors $\mathbf{u}_1, \mathbf{u}_2, ...\mathbf{u}_n$ for a speaker.

During verification, the test digit i-vectors $\mathbf{v}_1; \mathbf{v}_2; ...\mathbf{v}_n$ corresponding to the n randomly generated digits of the test utterance can also be extracted in the same way. Finally we splice the test digit local i-vectors into a final test i-vector ω_{test} from left to right in the prompt sequence (e.g.0, 1, n)

$$\omega_{test} = [\mathbf{v}_1; \mathbf{v}_2; ...\mathbf{v}_n] \tag{5}$$

The enrollment i-vector ω_{enroll} is also produced by splicing the enrollment digit i-vectors according to the same sequence:

$$\omega_{enroll} = [\mathbf{u}_1; \mathbf{u}_2; ...\mathbf{u}_n] \tag{6}$$

The scores are then computed as the cosine distance between the enrollment i-vector ω_{enroll} and test i-vector ω_{test} .

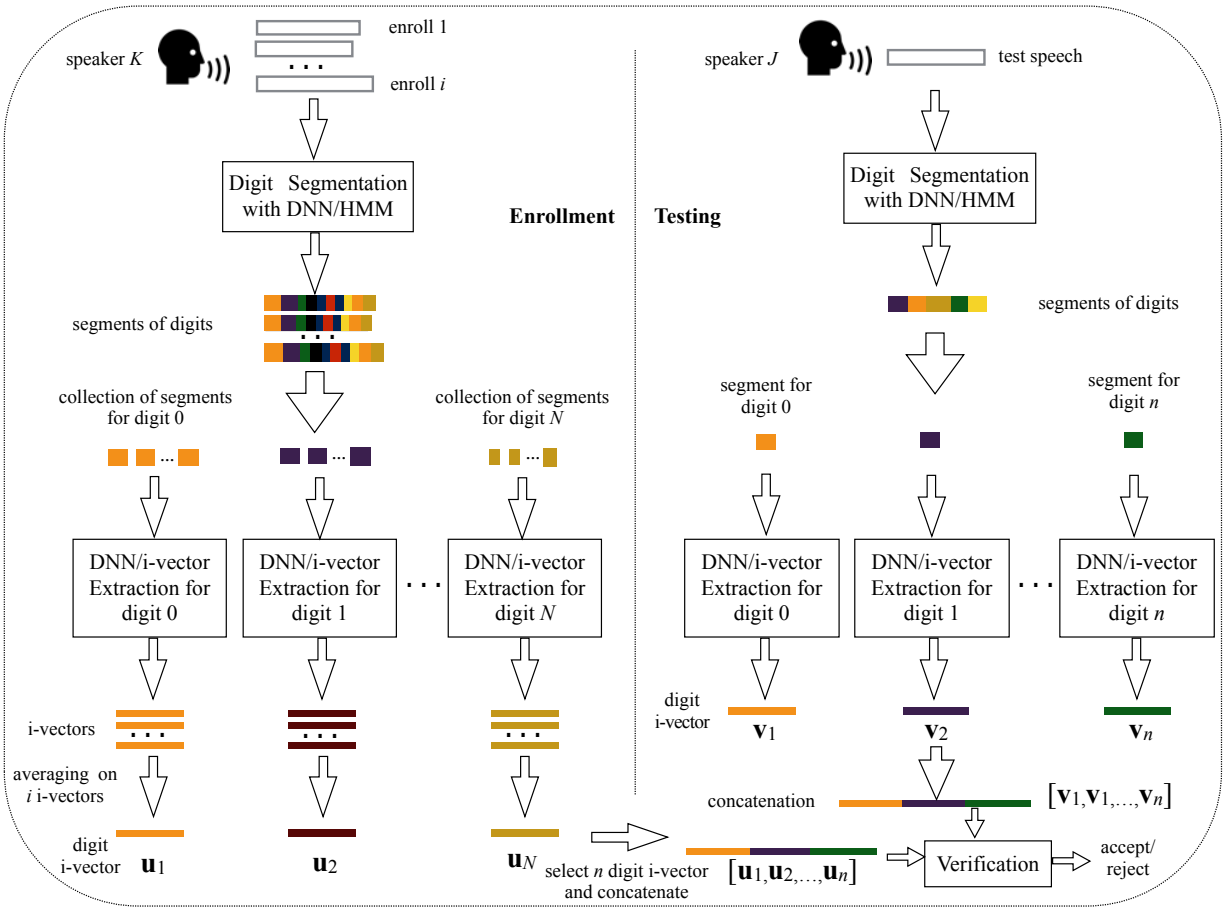

Fig. 1. Enrollment and testing for the segmental DNN/I-Vector approach.

Fig.1 shows the flow diagram of our system. Modeling at the digit level brings obvious benefits. On the one hand, parameters size of models can be greatly reduced since the digit-level i-vector usually has a smaller dimension. On the other hand, model simplification directly results in the compression of computation and space complexity. Despite the number of i-vectors we need to extract increases from one to the number of digits containing in test utterance, the extra computation and storage do not matter if we do the extractions in parallel.

B. Backend models

1) Confidence measure based i-vector weighting: After extracting all the digit i-vectors for a test utterance, we concatenate them into one vector from left to right according to the prompt sequence as described in Section A. However, the ASR system used to segment the digits cannot guarantee perfect frame boundary for all the digits. It is desirable to have an independent measure on how good the hypothesis is relative to the ground truth. To take into account the uncertainties in the DNN posteriors, confidence measure (CM) [15] [16] is used to weight the digit i-vectors before concatenation. Specifically, we compute the confidence measure by posterior probability.

The typical ASR algorithm uses the maximum a posteriori (MAP) decision rule to find the most likely sequence of words $\hat{W}$ which achieves the maximum posterior probability $p(W|X)$ given any acoustic observation X by $\hat{W} = argmax_W\, p(W|X) = argmax_W\, p(W)p(X|W)$. After being normalized by $p(X)$, the posterior probability $p(W|X)$ can serve as a confidence measure since it represents the absolute quantitative measure of the match between X and W.

An effective way to normalize $p(X)$ is calculating through the word lattice $\mathscr{X}$ generated by an ASR decoder for the utterance. The word lattice is represented as a directed, acyclic, weighted graph. Each arc is labeled by w_s^e, representing a hypothesized word w attached to the arc, which starts at frame s and ends at e. Given a lattice, the posterior probability of any arc a through a complete path C is calculated as a ratio between the total probability of all complete paths passing through the arc a to that of all complete paths in $\mathscr{X}$:

$$p(a|\mathscr{X}) = \frac{\sum_{C \subset \mathscr{X}, a \in C} p(C|\mathscr{X}))}{\sum_{C \subset \mathscr{X}} p(C|\mathscr{X}))} \tag{7}$$

The posterior probability $p(a|\mathscr{X})$ can be directly used as the confidence measure c_{word} for the recognized word. To further take into account other arcs which have the same word id w but slightly different s and e, when calculating confidence measure for the word w in an arc, we sum over all arcs in word graph which have the same word id and intersect with the current arc in the time domain.

In this paper, we use the CM calculated for the digits to

weight the digit i-vectors while splicing into the test i-vector:

$$\boldsymbol{\omega}_{test} = [c_1\mathbf{v}_1, c_2\mathbf{v}_2, ...c_n\mathbf{v}_n] \quad (8)$$

where the $c_1, ...c_n$ are the confidence measures for digits $1, 2...n$ containing in the test utterance and $\mathbf{v}_1, ...\mathbf{v}_n$ are the test digit local i-vectors. The enrollment i-vector remains unchanged when computing the cosine distance. In other words, the CM weight for enrollment digit local i-vectors is always regarded as one.

2) Dimension reduction and score normalization: In the i-vector framework, length normalization is usually used, simply by projecting the vectors onto the unit-sphere without pre-whitening. Furthermore, linear discriminant analysis (LDA) is often used for dimension reduction [4]. The LDA aims at maximizing between-class variance and minimizing intra-class variance. In this paper, we train ten LDA matrices for the ten digits. Each class is made up of all the segments to the target number of a single speaker. LDA dimension reduction is used prior to any other processing of the i-vectors, so the new test i-vector is written as:

$$\boldsymbol{\omega}_{test} = [c_1\mathbf{A}_1^T\mathbf{v}_1, c_2\mathbf{A}_2^T\mathbf{v}_2, ...c_n\mathbf{A}_n^T\mathbf{v}_n] \quad (9)$$

where the $\mathbf{A}_n$ is the LDA projection matrix of the target digit. The enrollment digit i-vectors are transformed in the same way. In addition to the confidence measure based backend method, score normalization is also applied for a better s-core distribution. In this paper we use test-dependent zero normalization (TZNorm) [17] for score normalization just like previous work in [12].

IV. EXPERIMENTS

A. Dataset

We use the RSR2015 part 3 dataset [8], which is devoted to speaker verification using randomly-prompted English digit strings, to evaluate the proposed approach. The dataset consists of 300 speakers (157 males and 143 females) with ages between 17 and 42. Each speaker contains 9 sessions and 13 utterances per session. According to the protocol in [8], session 1, 4 and 7 recorded with the same handset are chosen for enrollment so that each speaker contributes three different speaker models. The speaker model is enrolled with three 10-digit utterances. The test utterance from the remaining 6 sessions contains a random 5-digit string. For all of the utterances, the sequence of digits is known to the system and the speaker verification system can use it directly. Since the proposed method is aiming at speaker verification, the experiment results are provided by the target-correct trials and the non-target from the imposter-correct trials. The validity of the lexical content is judged by forced alignment.

B. Experimental setup

The RSR2015 Part 3 training and development sets are used to train the DNN and UBM models. We first train a GMM/HMM model to perform the force alignment to derive the digit state labels for the following DNN training. For the GMM/HMM training, since the corpus contains only ten digits from zero to nine, digit-based HMM models are adopted. Each digit is modeled by an HMM with 9 states. The input features are the standard 20-dimensional MFCC with its first and second derivatives, yielding a dimension of 60. No voice activity detection (VAD) is applied in all the systems. Using the forced-alignments from the GMM/HMM system, we train a five-hidden-layer feed-forward DNN with 9 frames context input (4+1+4) to predict totally 103 HMM states. The number of nodes in each hidden layer is set to 240. The Kaldi toolkit [18] is used for model training. In the GMMUBM i-vector baseline system, we train a gender-independent UBM of 512 mixture and **T** matrix of 400 dimensions with all the background set data. For comparison, a segmental GMM/i-vector system is also built with UBM of 64 mixtures and **T** matrix of 40 dimensions, for each digit using the segmented data. In our segmental DNN/i-vector system, the size of **T** matrices are also set to 40. Under these configurations, we can keep the amount of the parameters in digit dependent i-vector model is at the same level of traditional i-vector system.In summary, we have four systems for comparison.

- **GMM/i-vector**: the conventional utterance-level GMM-UBM i-vector system;
- **SegGMM/i-vector**: the segmental GMM-UBM i-vector system built for each single digit;
- **DNN/i-vector**: the utterance-level DNN/i-vector system;
- **SegDNN/i-vector**: the proposed segmental DNN/i-vector system built for each single digit.

C. Experiment results

Table I presents the performance of the four systems. Note that the scores are computed as the cosine distance between enrollment and test i-vectors and score normalization are applied to all these systems. In general, the proposed segmental i-vector systems perform better than the traditional utterance level i-vector systems. Specifically, the EERs for SegDNN/i-vector system are 4.75%/3.61% on female/male data, improved by 17%/21% relatively compared to the traditional utterance level DNN/i-vector system. We believe that the finer granularity modeling on the digit-level helps capture the lexical variability and thus improves the performance. Table II shows the performances of various backend methods applied

TABLE I
PERFORMANCE COMPARISON USING RSR PART 3 EVALUATION SET IN EER (%) / DCF08*100.

System	Female	Male
GMM/i-vector	10.63/50.14	9.01/44.90
SegGMM/i-vector	7.28/38.60	6.13/27.33
DNN/i-vector	5.76/33.25	4.58/23.17
SegDNN/i-vector	**4.75/27.18**	**3.61/15.99**

TABLE II
COMPARISON OF BACKEND METHODS IN EER (%) / DCF08*100.

Backend Method	Female	Male
SegDNN/i-vector	4.75/27.18	3.61/15.99
SegDNN/i-vector + LDA	4.12/22.33	3.01/14.10
SegDNN/i-vector + LDA + CM	**3.44/16.41**	**2.47/13.14**

TABLE III
RESULT COMPARISON WITH HILAM AND JFA IN EER (%) / DCF08*100.

System	Female	Male
HiLAM [8]	10.87/46.86	5.32/32.58
JFA-JDB [12]	4.54 /22.9	2.65/13.6
SegDNN/i-vector	**3.44 /16.41**	**2.47/13.14**

to the SegDNN/i-vector system. The length of digit i-vectors is reduced from 40 to 25 after the LDA transformation, which results in a better EER (4.12%/3.01% for female/male), indicating that the backend methods for original utterance-level i-vectors can also be applied to digit-level i-vectors before they are concatenated for scoring. Finally, confidence measure (CM) based i-vector weighting further improves the EER to 3.44%/2.47% for female/male. This may indicate that the CM-based i-vector weighting is necessary for compensating the uncertainty of the ASR system when segmenting the digits. Fig. 2 shows the test DET curves for the three DNN/i-vector systems for the female gender. We can see the EER reductions with the proposed segmental DNN/i-vector approach and the backend methods. Table III compares the performances of the proposed system with two popular methods, i.e., HiLAM [8] and JFA [12], on the same test condition. We can see that the proposed SegDNN/i-vector approach achieves the lowest EER and DCF08.

These experiments show the effectiveness of our system. By modeling the variability for each digit independently, the digit i-vectors can model both the speaker variabilities and the lexical variabilities of the passphrase. This is essential in the text-dependent speaker verification task. Digit-level i-vectors can also make use of the important co-articulation information within the digits, which is a weakness of the phone-dependent local i-vector method. Besides, confidence measure plays a major role in improving performance by compensating the uncertainty while segmenting the digits.

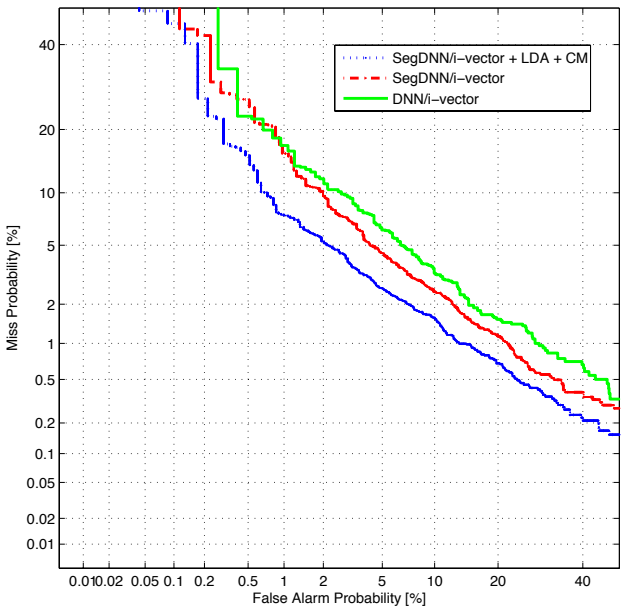

Fig. 2. DET curves on the female tests of RSR part 3.

V. CONCLUSIONS

In this paper, we proposed a segmental DNN/i-vector approach for digit-prompted verification task. The DNN/ivector systems were built using a finer granularity on the digit level rather than the sentence level for better lexical modeling. Moreover, we also explored a confidence measure based i-vector weighting method to compensate the uncertainty while segmenting the digits. Experiments were conducted on the text-prompted task of RSR2015. Experimental results show that our best system gives an EER of 2.47% and 3.44% for male and female genders respectively using the target-correct and imposter-correct trials, a significant improvement over the baseline utterance-level DNN/I-vector systems. The proposed method of i-vector modeling in the word level can also be applied to any other text-prompted speaker verification tasks.

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] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models" *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19-41, 2000.

[3] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," 2006.

[4] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio Speech & Language Processing IEEE Transactions* on, vol. 19, no. 4, pp. 788-798, 2011.

[5] H. Geoffrey, D. Li, Y. Dong, D. G. E., M. Abdelrahman, J. Navdeep, N. Patrick, and S. T. N, "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.

[6] Y. Lei, L. Ferrer, M. McLaren, and N. Scheffer, "A deep neural network speaker verification system targeting microphone speech," *INTERSPEECH*, pp. 681-685, 2014

[7] Y. Lei, N. Scheffer, L. Ferrer, and M. Mclaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," *ICASSP*, pp. 1695-1699, 2014.

[8] A. Larcher, A. L. Kong, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and rsr2015," *Speech Communication*, vol. 60, no. 3, pp. 56-77, 2014.

[9] L. Chen, A. L. Kong, B. Ma, W. Guo , H. Li, and R. Dai, "Phone-centric local variability vector for text-constrained speaker verification," *INTERSPEECH*, 2015.

[10] N. Scheffer and Y. Lei, "Content matching for short duration speaker recognition," *INTERSPEECH*, 2014.

[11] M. Tomoko and F. Sadaoki, "Concatenated phoneme models for text-variable speaker recognition," *ICASSP*, vol. 2, pp. 391-394,1993.

[12] T. Stafylakis, P. Kenny, J. Alam, and M. Kockmann, "JFA for speaker recognition with random digit strings," *INTERSPEECH*,2015.

[13] G. Wang, A. L. Kong, T. H. Nguyen, H. Sun, and B. Ma, "Joint speaker and lexical modeling for short-term characterization of speaker," *INTERSPEECH*, pp. 415-419, 2016.

[14] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," *INTERSPEECH*, 2011.

[15] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Communication*, vol. 45, no. 4, pp. 455-470, 2005.

[16] P. S. Huang, K. Kumar, C. Liu, and Y. Gong, "Predicting speech recognition confidence using deep learning with word identity and score features," *ICASSP*,pp. 7413-7417, 2013.

[17] F. Bimbot, J. Bonastre, C. Fredouille, G. Gravier, and e. a. I Magrin-Chagnolleau, "A tutorial on text-independent speaker verification," *EURASIP Journal on Advances in Signal Processing*,vol. 2004, no. 4, pp. 1-22, 2004.

[18] 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," *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.