HMM-Based Phrase-Independent i-Vector Extractor for Text-Dependent Speaker Verification

Hossein Zeinali, Hossein Sameti, and Lukáš Burget

***Abstract*—The low-dimensional i-vector representation of speech segments is used in the state-of-the-art text-independent speaker verification systems. However, i-vectors were deemed unsuitable for the text-dependent task, where simpler and older speaker recognition approaches were found more effective. In this work, we propose a straightforward hidden Markov model (HMM) based extension of the i-vector approach, which allows i-vectors to be successfully applied to text-dependent speaker verification. In our approach, the Universal Background Model (UBM) for training phrase-independent i-vector extractor is based on a set of monophone HMMs instead of the standard Gaussian Mixture Model (GMM). To compensate for the channel variability, we propose to precondition i-vectors using a regularized variant of within-class covariance normalization, which can be robustly estimated in a phrase-dependent fashion on the small datasets available for the text-dependent task. The verification scores are cosine similarities between the i-vectors normalized using phrase-dependent s-norm. The experimental results on RSR2015 and RedDots databases confirm the effectiveness of the proposed approach, especially in rejecting test utterances with a wrong phrase. A simple MFCC based i-vector/HMM system performs competitively when compared to very computationally expensive DNN-based approaches or the conventional relevance MAP GMM-UBM, which does not allow for compact speaker representations. To our knowledge, this paper presents the best published results obtained with a single system on both RSR2015 and RedDots dataset.**

***Index Terms*—Bottleneck features, DNN, hidden Markov model (HMM), i-vector, text-dependent speaker verification.**

I. Introduction

SPEAKER verification field has experienced rapid developments during the last decade and large improvements have been seen in terms of both computational complexity and accuracy. Newly introduced channel-compensation techniques, such as Joint Factor Analysis (JFA) [1], [2], have evolved in the i-vector paradigm [3], where each speech utterance is represented by a low-dimensional fixed-length vector. To verify speaker identity, the similarity between i-vectors can be measured by simple cosine similarity or using a more elaborate Bayesian model such as Probabilistic Linear Discriminant Analysis (PLDA) [4]–[6]. Fostered primarily by the NIST Speaker Recognition Evaluation (SRE) campaigns, most of this research has focused on the *text-independent* speaker verification task.

TABLE I
Trial Types in Text-Dependent Speaker Verification [8]

	Target Speaker	Imposter Speaker
Correct Pass-Phrase	Target-Correct	Imposter-Correct
Wrong Pass-Phrase	Target-Wrong	Imposter-Wrong

Recently, however, the demand for voice-based access control applications has increased, reviving interest in *text-dependent* speaker verification. Here the task is not only to verify the speaker of the tested utterance, but also to check whether the uttered phrase matches with the enrollment one (see Table I for types of errors). These systems can be *phrase-independent* (the user is given the freedom to enroll using any phrase of his choice) and *phrase-dependent* (the phrase is specified by the system). To complete the terminology, *text-prompted* speaker verification denotes a case where the phrases are composed from a limited predefined set of words.

Unfortunately, the modern techniques developed for *text-independent* speaker recognition were initially found quite ineffective for the *text-dependent* task — similar or better performance was usually obtained using slight modifications of simpler and older techniques such as Gaussian Mixture Model–Universal Background Model (GMM-UBM) [7], [8] or Nuisance Attribute Projection (NAP) compensated GMM mean super-vector scored using a Support Vector Machine (SVM) classifier [9]–[11].

The reason for the ineffectiveness of the *text-independent* techniques, and namely the i-vector approach, is the different nature of the data used for the *text-dependent* task: enrollment and test speech segments are typically very short, while the i-vector technique requires relatively long utterances in order to obtain reliable speaker representation. Large amounts of *text-independent* data from thousands of speakers are usually used for training the i-vector extractor, but it has been found difficult to leverage such data for the *text-dependent* task. Instead, it looks essential to train the *text-dependent* systems on a large amount of matching data from a predefined set of possible phrases. However, only small datasets (a couple of hundreds of speakers) are usually available for this purpose. Moreover, with a predefined

Manuscript received July 15, 2016; revised November 28, 2016 and March 2, 2017; accepted March 29, 2017. Date of publication April 17, 2017; date of current version June 5, 2017. The work was supported in part by Iranian Ministry of Science and in part by Czech Ministry of Interior project no. VI20152020025 "DRAPAK," European Union's Horizon 2020 project no. 645523 BISON and Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602." The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Bin Ma. *(Corresponding author: Hossein Zeinali.)*

H. Zeinali and H. Sameti are with the Department of Computer Engineering, Sharif University of Technology, Tehran 11365/8639, Iran, visiting Brno University of Technology, Czech Republic (e-mail: zeinali@ce.sharif.edu).

L. Burget is with the Faculty of Information technology, Brno University of Technology, Brno-stäed 601 90, Czech Republic (e-mail: burget@fit.vutbr.cz).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2017.2694708

set of phrases, the resulting system is not *phrase-independent*. Therefore, Stafylakis *et al.* [12] deemed the low-dimensional i-vector based representation as unsuitable for the *text-dependent* task, except perhaps for very impractical scenarios, where abundant data comprising only a limited number of possible phrases are available for the UBM and i-vector extractor training.

In this work, we propose a straightforward extension of the standard i-vector approach and we show that the low-dimensional representation of utterances can be successfully used in *text-dependent* speaker verification. Hidden Markov Model (HMM) based UBM is used in our i-vector extraction model in order to account for the left-to-right temporal structure of phrases in the *text-dependent* task. HMM models are trained for individual phonemes. For each enrollment or test utterance, phoneme specific HMMs are concatenated into a phrase specific HMM, which — instead of the conventional GMM-UBM — is in turn used to collect sufficient statistics for i-vector extraction. It should be noted that while there is a specific HMM built for each phrase, there is only one set of Gaussian components (Gaussians from all the HMM states of all phone models) corresponding to a single *phrase-independent* i-vector extraction model. The i-vector extractor is trained and used in the usual way, except that it benefits from better alignment of frames to Gaussian components which are constrained by the HMM model.

To compensate for the channel variability, we propose to normalize i-vectors using phrase-dependent Within-Class Covariance Normalization (WCCN) [13] derived from within-class covariance matrix regularized by adding a small constant to its diagonal. Simple cosine similarity scoring followed by phrase- and gender-dependent s-norm [6] score normalization was used for our experiments. We will show that this approach provides state-of-the-art performance on RSR2015 data [8].

Techniques making use of deep neural network (DNNs) have been recently devised in order to improve *text-independent* speaker verification: in one of the approaches, a DNN trained for phone classification is used to partition the feature space instead of the conventional GMM-UBM. In other words, DNN outputs are used to define the alignment for collecting the sufficient statistics for the i-vector extraction [14]–[17]. Another DNN-based approach, successful in *text-independent* speaker verification—as well as in other fields of speech processing [18]–[22]—is to use DNN for extracting frame-by-frame speech features. Typically, a bottleneck (BN) DNN is trained for phone classification, where the features are taken from a narrow hidden layer that compresses the relevant information into low dimensional feature vectors [22], [23]. Such features are then used as an input to the usual i-vector based system. In our previous works [24]–[26], we have shown that similar approaches can be also successfully applied for the *text-dependent* task.

The main objective of this paper is to describe and analyze the basic 'tricks' that are necessary to make i-vectors work for *text-dependent* speaker verification. Therefore, we focus on the simple configuration with mel-frequency cepstral coefficients (MFCC) features and HMM based alignment. Nevertheless, in order to make the story more complete, we also show how the simpler i-vector/HMM method compares to and combines with the aforementioned neural network approaches. For this purpose, we present selected results corresponding to the best performing DNN-based systems from our previous works [24], [26].

To the best of our knowledge, this paper reports the best published results obtained with individual systems on both RSR2015 and RedDots dataset with and without using the DNN-based techniques.

Let us now position our work within existing i-vector and HMM approaches to text-dependent speaker recognition and point out common points and changes.

To the best of our knowledge, the first successful attempt to use *i-vectors* for text-dependent speaker recognition was done by Stafylakis *et al.* [27] with a phrase-dependent PLDA. The authors found that the biggest problem was the uncertainty of i-vector estimation and tried to cope with it by considering the i-vector, not only as a point estimate, but as a whole distribution. Despite these efforts, the i-vector/PLDA system was only able to match the performance obtained with a simple GMM-UBM approach. Similar results were obtained by Larcher *et al.* [8] with i-vectors applied on the RSR2015 database. In our approach, we are coping with uncertainty by using an HMM instead of GMM. By applying an HMM with a temporal structure, we actually decrease the uncertainty of i-vector estimation (see section V-B) and can successfully use only the i-vector point estimates as obtained from a phrase-independent i-vector extractor.

In another branch of text-dependent speaker verification research, Kenny *et al.* concentrated on the use of JFA in place of the i-vector feature extractor[1] (see [28] and [29]). The JFA model decomposes utterance information into two low-dimensional vectors: channel vector $\mathbf{x}$, and speaker vector $\mathbf{y}$, and a high-dimensional residual vector $\mathbf{z}$. While the authors naturally discarded $\mathbf{x}$, they found that $\mathbf{y}$ was not enough for providing sufficient speaker recognition performance, probably due to the fact that in this vector, the phrase information is averaged out. To recover some of the information around the phrase structure, which is important for the text-dependent task, they had to also use $\mathbf{z}$ encoding of the information about the occupation of Gaussian components. Unfortunately, $\mathbf{z}$ is of high dimensions, and the UBM must be phrase-dependent in order to achieve good performance. In our approach, we target low-dimensional representations and ensure the phrase-specificity by properly concatenating the HMMs representing the phrase. There is still only one set of HMMs used for all phrases resulting in the phrase-independent i-vector extractor.

HMMs have a long tradition of deployment in text-dependent speaker recognition. In [30], Yu Kin *et al.* trained a separate HMM model for each digit and each one was scored separately. For creating speaker models, an HMM was trained from scratch for each digit. At this epoch, elaborate scoring techniques were still to be developed, so the system relied on simple likelihood scores. In [31], Chi Che *et al.* proposed a phoneme-based method: for each phoneme, an HMM model was created. While enrolling a speaker, these phoneme models were reestimated (not adapted). At the test stage, a phrase model was created by concatenating phoneme models and the score was calculated using Viterbi forced alignment. In [32], Toledano *et al.* first trained a phoneme recognizer using TIMIT. Speaker-dependent phoneme models were then created with two methods: reestimation or MLLR adaptation. Similarly as in [31], phrase models were created using concatenation. In [33], Dong *et al.* first trained speaker-independent HMM models. Then, for each speaker and phrase, first a phrase model was created using concatenation of initial HMMs and this model was then adapted

[1]Note that the typical use of JFA is for scoring.

to the speaker using maximum a posteriori (MAP) adaptation. The authors initially used LLR scoring, but then investigated stacking all HMM means to a super-vector and SVM scoring. The super-vector is different for each phrase and cannot be used in the phrase-independent mode.

Attempts have also been made to use new HMM like structure for text-dependent speaker verification. In the HiLAM architecture defined by Larcher *et al.* [8], [34], a text-independent GMM-UBM is trained first; then it is adapted to the target speaker. In the final step, it is cloned into a sequence of HMM states that are re-trained on the enrollment utterance. The split of the utterance is either uniform or governed by used digits. In contrast to this work, our approach is phonetically inspired, and rather than doing coarse initial division of utterance, it respects the content by the concatenation of appropriate phoneme models. We are also able to cope with any content (no limitation to digits or other pre-set words).

Stafylakis *et al.* [35] focuses on *text-prompted* speaker verification, where the prompts are concatenated digits from Part III of RSR2015 database. They define an HMM with a tied mixture model with a codebook shared across digits. A UBM trained on a sufficient amount of data from the Mixer corpus is adapted to RSR, and mixture weights are trained for each digit. The JFA model is used to provide **y** and **z** vectors for scoring, similar to [28], [29]. In a recent paper, [12], the above authors have generalized this scheme into a "multi-tier" structure allowing for the extraction of **y** and **z** vectors from either the whole utterance or individual HMM states. They investigate different ways of concatenating these features or fusing their scores. In our opinion, this work again does not come up with a compact representation and we also find it unfortunate that it ignores the phonetic structure of the pass-phrase which is generally known.

Our previous work [36] targeted *text-prompted* speaker verification as well. We investigated the HMM-based i-vector approach, where one HMM and one corresponding i-vector extractor were trained for each word (Farsi month names). For each enrollment or test utterance, word specific HMMs were concatenated into a phrase specific HMM, which — instead of the conventional GMM-UBM — was in turn used to collect sufficient statistics for i-vector extraction. Word-specific i-vectors were extracted for each word from a given phrase and the corresponding word-specific scores were fused (averaged). In this paper, we are extending this approach to the *text-dependent* task, where the HMMs are trained for individual phonemes rather than words, and where a single i-vector is extracted for a phrase given the phrase-independent i-vector extractor.

The paper is organized as follows: Section II reviews the classical i-vector based system for speaker verification. Section III introduces our i-vector/HMM based method and also deals with the necessary channel compensation. Section IV details the experimental setups while Section V presents the results and their analysis. We conclude in Section VI.

II. Conventional i-vector Based System

A. General i-Vector Extraction

Although thoroughly described in literature, let us review the basics of i-vector extraction in order to facilitate the comparison with the proposed techniques described in the following section. The main principle is that the utterance-dependent super-vector of concatenated GMM mean vectors $\mathbf{s}$ is modeled as

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

where $\mathbf{m} = [\boldsymbol{\mu}^{(1)^T}, \ldots, \boldsymbol{\mu}^{(C)^T}]^T$ is the GMM-UBM mean super-vector (of C components), $\mathbf{T} = [\mathbf{T}^{(1)^T}, \ldots, \mathbf{T}^{(C)^T}]^T$ is a low-rank matrix representing M bases spanning a subspace with an important variability in the mean super-vector space, and $\mathbf{w}$ is a latent variable of size M with standard normal distribution.

The i-vector $\boldsymbol{\phi}$ is the MAP point estimate of the variable $\mathbf{w}$. It maps most of the relevant information from a variable-length observation $\mathcal{X}$ to a fixed-dimensional vector. The closed-form solution for computing the i-vector can be expressed as a function of the *zero-* and *first-order statistics*: $\mathbf{n}_\mathcal{X} = [N_\mathcal{X}^{(1)}, \ldots, N_\mathcal{X}^{(C)}]^T$ and $\mathbf{f}_\mathcal{X} = [\mathbf{f}_\mathcal{X}{}^{(1)^T}, \ldots, \mathbf{f}_\mathcal{X}{}^{(C)^T}]^T$:

$$N_\mathcal{X}^{(c)} = \sum_t \gamma_t^{(c)} \tag{2}$$

$$\mathbf{f}_\mathcal{X}^{(c)} = \sum_t \gamma_t^{(c)} \mathbf{o}_t, \tag{3}$$

where $\gamma_t^{(c)}$ is the posterior (or occupation) probability of frame $\mathbf{o}_t$ being generated by the mixture component c. The tuple $\gamma_t = (\gamma_t^{(1)}, \ldots, \gamma_t^{(C)})$ is usually referred to as *frame alignment*. Note that this variable can be computed either using the GMM-UBM or by using a separate model [14], [22], [37]. In this paper, we compare the standard GMM-UBM frame alignment with the HMM-based approach, described in the following section. In section V-F, we also experiment with the DNN based alignment, where $\gamma_t^{(C)}$) are taken from the output of DNN trained for senone classification. The i-vector is the mean of the posterior distribution of $\mathbf{w}$ computed as:

$$\boldsymbol{\phi}_\mathcal{X} = \mathbf{L}_\mathcal{X}^{-1} \bar{\mathbf{T}}^T \bar{\mathbf{f}}_\mathcal{X} \tag{4}$$

where $\mathbf{L}_\mathcal{X}$ is the precision matrix of the posterior distribution of $\mathbf{w}$ computed as:

$$\mathbf{L}_\mathcal{X} = \mathbf{I} + \sum_{c=1}^{C} N_\mathcal{X}^{(c)} \bar{\mathbf{T}}^{(c)^T} \bar{\mathbf{T}}^{(c)}, \tag{5}$$

where c is the GMM-UBM component index, and the 'bar' symbols denote normalized variables:

$$\bar{\mathbf{f}}_\mathcal{X}^{(c)} = \boldsymbol{\Sigma}^{(c)-\frac{1}{2}} \left(\mathbf{f}_\mathcal{X}^{(c)} - N_\mathcal{X}^{(c)} \boldsymbol{\mu}^{(c)} \right) \tag{6}$$

$$\bar{\mathbf{T}}^{(c)} = \boldsymbol{\Sigma}^{(c)-\frac{1}{2}} \mathbf{T}^{(c)}, \tag{7}$$

where $\boldsymbol{\Sigma}^{(c)-\frac{1}{2}}$ is the matrix square root (or a symmetrical decomposition such as Cholesky decomposition) of an inverse of the GMM-UBM covariance matrix $\boldsymbol{\Sigma}^{(c)}$. Note that the *normalization GMM-UBM* (i.e., the $\boldsymbol{\mu}^{(c)}$ and $\boldsymbol{\Sigma}^{(c)}$ parameters) should be estimated via the same alignment as used in (2) and (3).

B. Scoring and Channel Compensation

Modeling or reducing channel effects (or, more generally, intra-speaker variability) is crucial for the good performance of any speaker verification system. For the text-independent task, the most successful model for comparing i-vector is PLDA,

where both within and between-speaker variability in the i-vector space is explicitly modeled. However, it has been shown in [27] that PLDA is not very effective in the case of text-dependent scenario. Therefore, we use the simple cosine similarity scoring, where the speaker verification score for a trial with enrollment i-vector $\boldsymbol{\phi}_e$ and test i-vector $\boldsymbol{\phi}_t$ is obtained as the cosine similarity between the i-vectors

$$s_{e,t} = \frac{\boldsymbol{\phi}_e^T \boldsymbol{\phi}_t}{\|\boldsymbol{\phi}_e\| \|\boldsymbol{\phi}_t\|}. \tag{8}$$

As is usual in the case of cosine similarity scoring, we precondition i-vectors using WCCN in order to compensate for the channel variability. More precisely, i-vector distribution is whitened by multiplying each i-vector using transformation matrix $\mathbf{S}_w^{-\frac{1}{2}}$, which is the inverse of the square root (or a symmetrical decomposition) of the within-class covariance matrix

$$\mathbf{S}_w = \frac{1}{S} \sum_{s=1}^{S} \frac{1}{N_s} \sum_{n=1}^{N_s} (\mathbf{w_s}^n - \overline{\mathbf{w}}_s)(\mathbf{w_s}^n - \overline{\mathbf{w}}_s)^T, \tag{9}$$

with S being the total number of speakers (i.e., classes), N_s the number of training samples for speaker s, $\mathbf{w}_s^n$ the n^{th} training i-vector from speaker s and $\overline{\mathbf{w}}_s = \frac{1}{N_s} \sum_{n=1}^{N_s} \mathbf{w}_s^n$ the mean of the training i-vectors for speaker s.

Before the WCCN, we optionally apply Linear Discriminant Analysis (LDA) trained with the speaker labels as classes in order to preserve only the dimensions with high between speaker variability. Note that LDA and WCCN transformation needs to be estimated in a phrase-dependent fashion (i.e., only on training data of matching phrases) in order to obtain satisfactory results.

When measuring distances between i-vectors in speaker verification, it was found to be more important to compare the angles between i-vector rather than the Euclidean distances. This is also the reason for using the cosine similarity for scoring. When PLDA is used for scoring in text-independent speaker verification, i-vectors are usually normalized to unity length [38] for similar reasons. Such *length normalization* makes the resulting distribution of i-vectors more Gaussian-like and places i-vectors with a small angular distance close to each other. Although, the *length normalization* is implicit for calculating the cosine similarity (i.e., it is a dot product of two length normalized vectors), we still found it useful to explicitly normalize i-vectors before estimating or applying WCCN or LDA transformation. In summary, in order to produce the verification score for a pair of i-vectors, we first normalize the i-vectors to unity length, optionally apply LDA dimensionality reduction, apply WCCN preconditioning and calculate the cosine distance similarity score. Finally, we normalize the score as described in the next section.

In the official RSR2015, Part-1 experimental setup, each speaker enrollment (model definition) consists of three enrollment utterances of the same phrase. In order to obtain only one i-vector (speaker model) per enrollment, we simply average the three i-vectors extracted from each utterance.

C. Score Normalization

Various score normalization methods are used in speaker verification. In our work, phrase- and gender-dependent s-norm [6] was experimentally found to perform the best. Normalized verification score is defined as

$$\tilde{s}_{e,t} = \frac{s_{e,t} - \mu_{e,p}}{\sigma_{e,p}} + \frac{s_{e,t} - \mu_{p,t}}{\sigma_{p,t}}, \tag{10}$$

where $s_{e,t}$ is the unnormalized verification score (8) and $\mu_{s,p}$ and $\sigma_{s,p}$ represent the mean and standard deviation of scores obtained by scoring the enrollment i-vector $\boldsymbol{\phi}_e$ against a cohort set of imposter i-vectors. Similarly, $\mu_{p,t}$ and $\sigma_{p,t}$ are obtained from scoring the test i-vector $\boldsymbol{\phi}_t$ against the i-vector cohort.[2]

III. Proposed i-Vector/HMM Method

A. HMMs in Text-Dependent i-Vector Systems

In Section I, we discussed the advantages of the i-vector/HMM approach to text-dependent speaker verification. This section presents its details. The first step is to train a phoneme recognizer with 3-state, GMM-based, left-to-right mono-phone HMMs: let F be the total number of mono-phones, $S = 3F$ the number of all states, G the number of Gaussian components per state, $C = SG$ the number of all individual Gaussians and let (s, g) denote g^{th} Gaussian component in state s. Then, for each phrase (based on the transcribed sequence of phonemes in that phrase), a new phrase-specific HMM is constructed by concatenating the corresponding mono-phone HMMs. This step is shown at the top of Fig. 1.

For extracting zero- and first-order statistics, the Viterbi algorithm is used to obtain the alignment of frames to the states of the phrase-specific HMM. Since the Viterbi algorithm provides hard alignment, each frame t is aligned to exactly one HMM state s. Within the HMM state s, soft alignment $\gamma_t^{(s,g)}$ is calculated as the posterior (or occupation) probability of Gaussian component g. Note that the same phone can occur in a given phrase multiple times. Therefore, the concatenated phrase-specific HMM can contain multiple replicas of the same phone specific HMM. For example, Fig. 1 shows a phrase HMM with two copies of phoneme "G". To collect the sufficient statistics, however, we consider only a single HMM per phone, and all the frames aligned to different replicas of a phone are, in fact, aligned with the states of the same phone HMM. For example, in Fig. 1, the first and seventh states of the phrase model correspond to the first HMM state of phone "G". Therefore, all frames aligned by the Viterbi algorithm to those two phrase-specific states are, in fact, aligned to the first HMM state of phone "G". A similar strategy is used to accumulate statistics in the standard "embedded" HMM training.

We can now re-interpret the pair (s, g) as one out of C Gaussians and we can substitute $\gamma_t^{(c)}$ in (2) and (3) by $\gamma_t^{(s,g)}$, so that the zero- and first-order statistics can be written as:

$$\mathbf{n}_{\mathcal{X}} = \left[N_{\mathcal{X}}^{(1,1)}, \ldots, N_{\mathcal{X}}^{(s,g)}, \ldots, N_{\mathcal{X}}^{(S,G)} \right]^T$$

$$\mathbf{f}_{\mathcal{X}} = \left[\mathbf{f}_{\mathcal{X}}^{(1,1)^T}, \ldots, \mathbf{f}_{\mathcal{X}}^{(s,g)^T}, \ldots, \mathbf{f}_{\mathcal{X}}^{(S,G)^T} \right]^T,$$

[2] To be more precise, when estimating $\mu_{p,t}$ and $\sigma_{p,t}$, each i-vector (non-target speaker model) in the cohort is estimated on three utterances in the same manner as the enrollment i-vector as described in Section II-B.

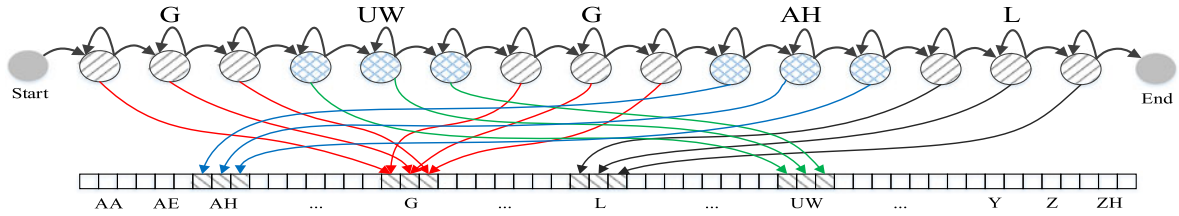

Fig. 1. The process of estimating sufficient statistics: In the top, the left-to-right phrase-specific model is shown. Each state is considered as a small GMM model. The vector in the bottom shows one of the zero or first order statistic vectors. Here, each cell shows a part of the statistics associated with state s of one of the phone HMM. Obviously, in this modeling, only the value of the subparts of this vector that are connected to the states of the phrase model will accumulate non-zero values.

where,

$$N_{\mathcal{X}}^{(s,g)} = \sum_t \gamma_t^{(s,g)} \tag{11}$$

$$\mathbf{f}_{\mathcal{X}}^{(s,g)} = \sum_t \gamma_t^{(s,g)} \mathbf{x}_t \ , \tag{12}$$

Note that the resulting vector of statistics has a fixed size and is structure independent of the actual phrase (as also demonstrated in Fig. 1), which can be used to train the phrase-independent i-vector extractor. Note also that in (11) and (12), due to the typically short duration of phrases, not all phonemes are used in the phrase-specific HMM. Therefore, the alignment of frames to the Gaussian components is often sparse and most of the $\gamma_t^{(s,g)}$ values are zero.

Equations (5) to (7) can now be changed to use the HMM alignment $\gamma_t^{(s,g)}$:

$$\mathbf{L}_{\mathcal{X}} = \mathbf{I} + \sum_{s=1}^{S}\sum_{g=1}^{G} N_{\mathcal{X}}^{(s,g)} \bar{\mathbf{T}}^{(s,g)^T} \bar{\mathbf{T}}^{(s,g)} \tag{13}$$

$$\bar{\mathbf{f}}_{\mathcal{X}}^{(s,g)} = \mathbf{\Sigma}^{(s,g)-\frac{1}{2}} \left(\mathbf{f}_{\mathcal{X}}^{(s,g)} - N_{\mathcal{X}}^{(s,g)} \boldsymbol{\mu}^{(s,g)} \right) \tag{14}$$

$$\bar{\mathbf{T}}^{(s,g)} = \mathbf{\Sigma}^{(s,g)-\frac{1}{2}} \mathbf{T}^{(s,g)} \ . \tag{15}$$

B. Regularized WCCN

For reducing channel effects, we can simply use LDA to reduce the dimensionality of i-vectors as explained in Section II-B. However, the number of available speakers (classes) to estimate this transformation is usually low in text-dependent speaker verification (around 100-200 in RSR2015) and LDA reduces i-vectors dimensionality to an utmost number of classes minus one. Such dimensionality reduction causes losing important information as demonstrated in our experiments. Therefore, we decided to use WCCN instead of LDA, which is able to preserve all dimensions. Still we need to deal with a relatively small amount of training data in RSR2015 dataset, preventing us from robustly estimating WCCN transformation in a straightforward way.

In [39], the authors proposed to add a fraction of the total covariance matrix to the covariance matrix of each class in order to obtain a more robust estimate of LDA transformation. Inspired by this work, we use Regularized WCCN (RWCCN), which is based on the regularized estimation of the within-class covariance matrix. Since the prior distribution of i-vectors is assumed to be normal with an identity covariance matrix, we add a fraction of the identity matrix (instead of the total covariance matrix) to the within-class covariance matrix estimate. In other words, instead of (9), we estimate the regularized within-class covariance matrix

$$\mathbf{S}_w = \alpha \mathbf{I} + \frac{1}{S}\sum_{s=1}^{S}\frac{1}{N_s}\sum_{n=1}^{N_s}(\mathbf{w_s}^n - \overline{\mathbf{w}_s})(\mathbf{w_s}^n - \overline{\mathbf{w}_s})^T), \tag{16}$$

where, $\mathbf{I}$ is the identity matrix and α is regularization coefficient, which must be determined on the development set. Other notations are identical to the ones used in (9). Note again that i-vectors are length-normalized prior to estimating or applying LDA, WCCN or RWCCN.

C. HMM Alignment for Enrollment and Test Utterances

In our experiments, we assume that the pass-phrase transcriptions are known for the enrollment utterances. This allows us to construct the phrase-specific HMM and obtain the Viterbi alignment for extraction of the zero- and first-order statistics. In order to correctly accept a test utterance, it also has to contain the same enrollment pass-phrase. Therefore, to extract the zero- and first-order statistics for the test utterance, we also use the same phrase-specific HMM (i.e., Viterbi alignment is obtained using the same enrollment pass-phrase). If the test utterance really contains the correct phrase, we obtain good frame alignment and the extracted test i-vector is appropriate for scoring. On the other hand, if the speaker uttered a wrong phrase instead of the pass-phrase (Target-Wrong and Imposter-Wrong trials in Table I), frames are aligned to the wrong states (i.e., Gaussians) of the phrase-specific model and the extracted statistics become incorrect. In the results, we will see that such trials were easily rejected by our i-vector/HMM technique. On the other hand, if GMM was used instead of HMM, such trials may be accepted due to common phonemes between two utterances.

IV. Experimental Setups

A. Data

We used the RSR2015 data set [8] for most of our experiments. It consists of recordings from 300 speakers (157 males and 143 females), each in 9 sessions. The data is divided into three sets (background, development and evaluation). It was also divided into three parts based on different lexical constraints. Part-1 is used for text-dependent speaker verification, where enrollment and test utterances contains one of the 10 predefined phrases. Part-2 is suitable for command controls and verification speakers with short commands. Part-3 focuses on text-prompted speaker verification where speakers are prompted to say a random sequence of predefined words (English digits). In this paper, we focus on text-dependent speaker verification, so we only use Part-1. In this part, each speaker uttered 30 different phrases

from TIMIT in 9 sessions. For each phrase, three repetitions from different sessions were used to enroll a single i-vector as a speaker model (see section II-B) and other phrases were used for testing in accordance with the RSR2015 trial definition.

In all experiments, the RSR2015 background set was used for UBM (including the phoneme recognizer and GMM) and i-vector extractor training. The evaluation set was used for testing (except where indicated otherwise). Training was done in a gender-independent manner (except where indicated otherwise). We used all speakers from the background set for gender independent RWCCN and gender-dependent score normalization (except where stated otherwise). Based on our experimental results, we decided to use phrase-dependent RWCCN and score normalization in all experiments (except where stated otherwise). RWCCN and/or s-norm estimated in a phrase-independent manner actually degrades the performance compared to not applying any WCCN or s-norm as was also demonstrated in the experiments from section V-H. This is consistent with the phrase-dependent PLDA proposed in [27]. The RSR2015 development set was only used to tune the RWCCN regularization parameter α unless stated otherwise.

We used a part of freely available LibriSpeech data (i.e., Train-Clean-100) [40], with 251 speakers and about 100 hours of speech. In this dataset, each speaker reads several books and each recording was split into short segments ranging from one to several sentences. For each segment, there is a word-level transcription. This dataset was used in two experiments investigating the effects of heterogeneous, text-independent data for UBM and i-vector extractor training and also for RWCCN and score normalization.

In order to verify our approach on different data, we also used the RedDots data set [41] in several experiments. RedDots contains 62 speakers (49 males and 13 females). 41 speakers are the target speakers (35 males and 6 females) and the other ones are considered as unseen imposters. RedDots consists of four parts; in our work, we used only Part-01. In this part, each speaker uttered 10 common phrases. The definition of RedDots trials distinguishes three types of incorrect trials (see Table I), so we report results for each type separately. The RedDots evaluation data comes without any development set, which would contain recordings of the same phrases as used for enrollment and testing. Therefore, RSR2015 and LibriSpeech were used as training data in RedDots experiments .

The DNN for the experiments with DNN-based alignment and BN features were trained on the Switchboard-1 (Phase-1 Release 2). Note that while 16 kHz speech data is used in our text-dependent speaker recognition experiments, the DNNs are trained only on 8 kHz telephone speech. Therefore, whenever processing speech by DNNs, it is first downsampled to 8 kHz. See [26] for text-dependent speaker recognition experiments comparing DNNs trained on 8 kHz and 16 kHz speech.

The number of speakers in RSR2015, LibriSpeech and RedDots data sets are shown in Table II. Note that we use exactly the same training and test sets as [29], where some of the trials of very short duration and low signal to noise ratios were removed from the original RSR2015 setup [8].[3] Therefore, our results should be directly comparable with the best results reported in [29, Table 6]. We also use the same HTK-based MFCC features as in [29]. However, we use our own voice activity detection (VAD) different from [29]. The number of evaluation trials for each trial type is listed in Table III.

TABLE II
DATASETS, PARTS AND NUMBERS OF SPEAKERS [8], [40], [41]

Dataset	Subset	# Males	# Females
RSR2015	Background	50	47
	Development	50	47
	Evaluation	57	49
LibriSpeech	Train-Clean-100	126	125
RedDots	Part-01	49	13

TABLE III
NUMBERS OF EVALUATION TRIALS IN PART-1 OF RSR2015 AFTER REMOVING UTTERANCES AS IN [12], [27], [29]

Trial Type	Male	Female
Target-Correct	9670	8670
Target-Wrong	297076	255143
Imposter-Correct	541301	416166
Imposter-Wrong	8318132	6123417

B. Features

Different types of features were used. The 60-dimensional MFCCs were selected as the primary features, which are used for most of the experiments. In addition, several experiments were carried out using 60-dimensional PLPs and 39-dimensional MFCCs and PLPs. All features were extracted from 25 ms Hamming windowed signals with 15 ms overlaps using HTK [42]. Contrary to text-independent systems, simple VAD could not be used, as VAD errors would harm the Viterbi alignment. Therefore, we used a silence HMM model to model silences at the beginning and end of each utterance. The frames aligned to this silence model are dropped (i.e., not used in the following estimation of statistics and i-vector extraction). We assumed that there is no silence in the middle of utterances; this is a plausible assumption as the utterances are very short.[4] Finally, cepstral mean and variance normalization was applied to trimmed utterances.

Beside the cepstral features, 80-dimensional BN features are used in our experiments. The bottleneck neural network refers to DNN with a specific topology, where one of the hidden layers has a significantly lower dimensionality than the surrounding layers. A bottleneck feature vector is generally understood as a by-product of forwarding a primary input feature vector through the DNN, while reading the vector of values at the output of the bottleneck layer. In this work, we use more elaborate architecture for BN features called Stacked Bottleneck Features described in detail in [22], [43]. With this architecture, each output feature vector is effectively extracted from 30 frames (300 ms) of the input features in the context around the current frame. We have used this architecture as it proved to be very effective in our previous *text-independent* speaker recognition experiments [22]. However, our more recent experiments indicate that similar results can be obtained with simpler BN neural

[3]We thank the authors of [29] for sharing their enrollment and trial lists.

[4]Only a slight improvement was obtained when properly modeling phrases with an optional silence after each word. Therefore, we decided to report results with the simpler model dropping only initial and final silence regions.

networks. The BN DNNs are trained to classify 8802 triphone tied states (senones).

C. Models for i-Vector Extraction

We modeled each phoneme by an HMM with 3 states and 8 Gaussian components per state. The total number of phonemes is 39 and the total number of Gaussian components in the whole model is $3 \times 8 \times 39 = 936$. The frames aligned to the extra model for silence (see above) were dropped. To compare the proposed method with the classical GMM-UBM, we used a GMM with 1024 components. We used 400-dimensional i-vectors unless stated otherwise.

The RWCCN regularization coefficients α from (16) was tuned for the best performance on the development set of RSR2015, as this data was not used for training of any other parameter in our systems. The best value for α was found to be 0.001.

For the experiment with DNN-based alignment, we use the same DNN architecture as for the BN feature extraction described in the previous section. In this case, however, the DNN is trained to classify 1011 senones and the outputs of the DNN (i.e., the senone posteriors) are used as $\gamma_t^{(c)}$ in (2) and (3)).

V. Results

In all experiments, if we use a different configuration from the one mentioned in earlier sections, we explicitly mention it. We report results in terms of Equal Error Rate (EER) and Normalized Detection Cost Function as defined for NIST SRE08 ($\mathrm{NDCF}_{\mathrm{old}}^{\mathrm{min}}$) and NIST SRE10 ($\mathrm{NDCF}_{\mathrm{new}}^{\mathrm{min}}$). In all DET curves, the square and star markers correspond to $\mathrm{NDCF}_{\mathrm{old}}^{\mathrm{min}}$ and $\mathrm{NDCF}_{\mathrm{new}}^{\mathrm{min}}$ operating points, respectively.

A. Features

Different features and their dimensionalities were investigated with the i-vector/HMM based method. We just compare the two most common features (MFCC and PLP) here and show the performance of their score fusion.

From the first section of Table IV, it can be seen that there is no remarkable performance difference between the different features. The 39 dimensional features work a little better for females, while there is no general rule for males. Interestingly, as can be seen from the second section of the table (and Fig. 2), large improvements in performance can be obtained from the simple score level fusion of two systems trained on different features. These results are consistent with our results on other text-dependent datasets. The combination of 60- and 39-dimensional features has better performance than other combinations. Based on these results, we select MFCC60 as the primary features for other experiments.

B. Comparison of GMM and HMM Alignment Methods

In this section, we compare the i-vector/HMMs with the standard i-vector/GMMs. Table V shows the results and Figs. 3 and 4 present the DET curves for males and females, respectively. The score-domain fusion of both techniques is reported as well. In order to compare the proposed HMM-based method with GMM, we report the results separately for three conditions, each considering only one of the three types of non-target trials from Table I. Target trials are the same for all three conditions.

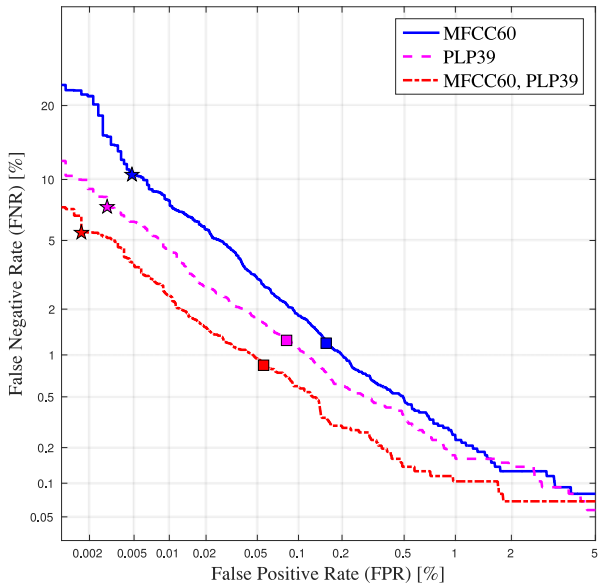

Fig. 2. DET curves of two different features and their score domain fusion for Imposter-Correct as non-target trials of females. From the worst plot to the best: 1) MFCC60, 2) PLP39, 3) Score domain fusion.

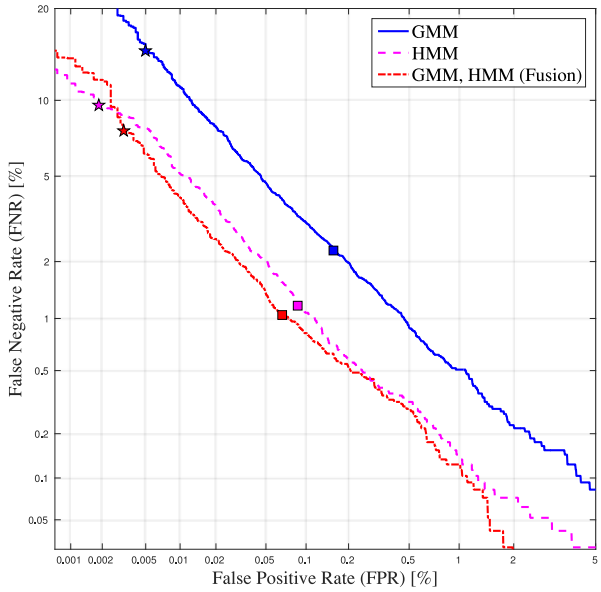

Fig. 3. DET curves of the proposed i-vector/HMM, standard i-vector/GMM and their score domain fusion for Imposter-Correct as non-target trials of males. From the worst plot to the best: 1) GMM, 2) HMM, 3) Score domain fusion.

For the *Imposter-Correct trials*, the proposed method reduced the error in all criteria and for both genders too. HMM outperforms GMM due to better frame alignment to Gaussian components — in the proposed method, the Viterbi alignment aligns many consecutive frames to the same HMM state and the posterior probabilities of all such frames are forced to be non-zero for only a few Gaussian components. Compared to GMM, this leads to a more restricted and sparser assignment of frames to the Gaussian components. Consequently, for HMM, the entropy of the frame posteriors is lower, and, as mentioned in the Introduction, this reduces the uncertainty of i-vector for short utterances (the trace of the

TABLE IV
PERFORMANCE COMPARISON OF THE PROPOSED METHOD WITH DIFFERENT FEATURES AND THEIR SCORE DOMAIN FUSION FOR IMPOSTER-CORRECT AS NON-TARGET TRIALS

	Features	Male			Female		
		EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
Single System	MFCC39	0.40	0.0212	0.1087	**0.40**	**0.0197**	0.1200
	MFCC60	**0.37**	**0.0204**	0.1142	0.49	0.0275	0.1533
	PLP39	0.41	0.0217	0.1103	0.42	0.0207	**0.1029**
	PLP60	0.52	0.0232	**0.1023**	0.44	0.0238	0.1300
Score Fusion	MFCC39, PLP39	0.31	0.0152	0.0874	0.27	0.0148	0.0731
	MFCC60, PLP60	0.32	0.0157	0.0782	0.33	0.0190	0.1107
	MFCC39, PLP60	0.29	0.0142	**0.0619**	**0.24**	**0.0127**	0.0773
	MFCC60, PLP39	**0.25**	**0.0123**	0.0712	0.27	0.0139	**0.0721**

RWCCN and s-norm were used. Numbers concatenated to each feature show dimensionality of them.

TABLE V
COMPARISON OF GMM AND HMM ALIGNMENT METHODS AND THEIR SCORE DOMAIN FUSION (THIRD SECTION) FOR THREE NON-TARGET TRIAL TYPES

Method	Non-Target Trial Type	Male			Female		
		EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
i-vector/GMM	Imposter-Correct	0.67	0.0382	0.1983	0.62	0.0355	0.1991
	Target-Wrong	2.25	0.1461	0.7036	0.72	0.0483	0.3704
	Imposter-Wrong	0.10	0.0060	0.0507	0.03	0.0020	0.0129
i-vector/HMM	Imposter-Correct	0.37	0.0204	0.1142	0.49	0.0275	0.1533
	Target-Wrong	**0.30**	**0.0162**	**0.1111**	**0.13**	**0.0072**	**0.0620**
	Imposter-Wrong	0.03	0.0013	0.0088	0.06	0.0011	0.0045
i-vector/GMM + i-vector/HMM	Imposter-Correct	**0.35**	**0.0170**	**0.1080**	**0.32**	**0.0184**	**0.1108**
	Target-Wrong	0.54	0.0318	0.1887	**0.13**	0.0076	0.0892
	Imposter-Wrong	**0.01**	**0.0008**	**0.0082**	**0.02**	**0.0006**	**0.0015**

The total number of Gaussian components are 1024 and 936 for GMM and HMM respectively. RWCCN and s-norm were used for all experiments.

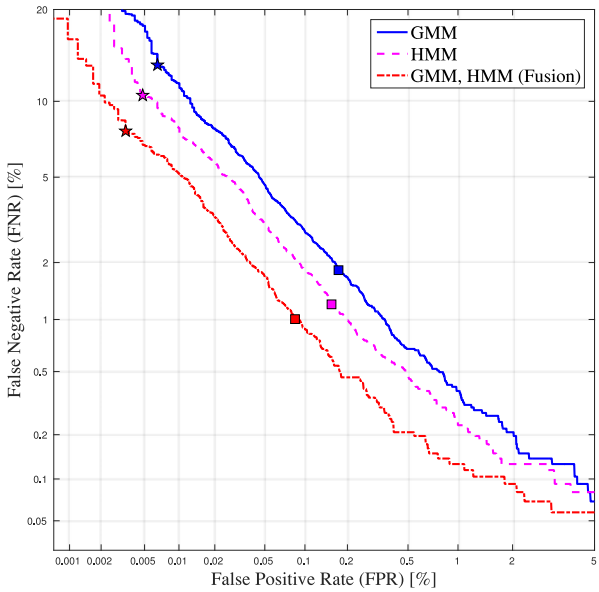

Fig. 4. DET curves of the proposed i-vector/HMM, standard i-vector/GMM and their score domain fusion for Imposter-Correct as non-target trials of females. From the worst plot to the best: 1) GMM, 2) HMM, 3) Score domain fusion.

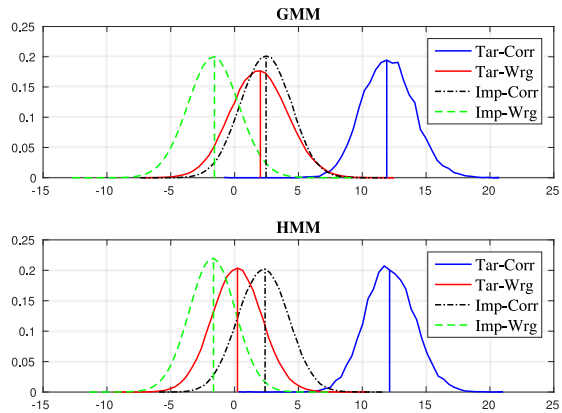

Fig. 5. Comparison of score distributions of GMM and HMM for females. The vertical lines show the means of normal distributions fitted to the scores. In each figure, plots from the left are for Imp-Wrg, Tar-Wrg, Imp-Corr and Tar-Corr respectively.

covariance matrix (5) is reduced from 67 to 51). Score distributions in Fig. 5 give us another insight into the performance for this trial type: the overlap between Target-Correct and Imposter-Correct distributions for HMM is lower than for GMM.

One of the main advantages of the proposed method is its ability to reject *Target-Wrong* trials. In a practical application, the text-dependent speaker verification system must be able to reject this type of trial in order to prevent the replay attacks. In the proposed method, Viterbi forced alignment must respect the pass-phrase transcription and as a result aligns the frames from

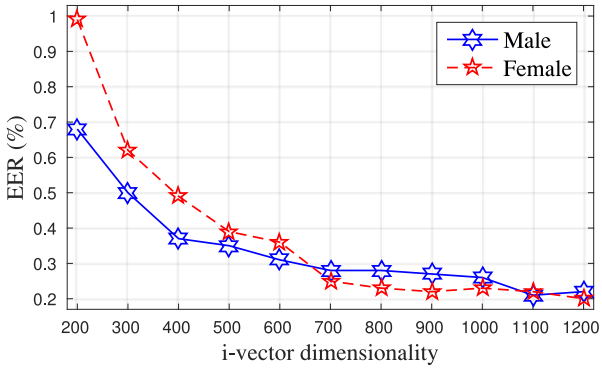

Fig. 6. EER versus i-vectors dimensionality for Imposter-Correct as non-target trials. The line with hexagram marker shows male and line with pentagram marker shows female. The trends for $\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{old}}$ and $\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{new}}$ are very similar.

a wrong phrase to wrong HMM states. This results in incoherent zero- and first-order statistics. An i-vector extracted from these statistics has a large distance to the enrollment one and is therefore rejected easily. On the contrary, the GMM-based system has no control over the order of frames, therefore, for this method, this type of trial is more difficult and more often accepted. Our method reduced the average error in this trial type by more than 84%. Fig. 5 clearly shows that for the proposed method, the distance between Target-Correct and Target-Wrong scores distributions is bigger than for GMM; therefore, the rejection of the Target-Wrong trials is easier.

For *Imposter-Wrong* trials, better results are obtained with the HMM-based method in almost all cases. However, both GMM and HHM produce only a few errors and the differences between their performance is not statistically significant when evaluated on the small RSR2015 data set.

Note that the GMM-based i-vectors are able to discriminate between phrases (and therefore to reject wrong phrases) only thanks to the short phrases in the RSR2015 data set. For longer and phonetically rich phrases, information about the phonetic content of the utterances tends to average out in the i-vector/GMM representation. In contrast, the i-vector/HMM method does not suffer from this problem. Therefore, in terms of rejecting the wrong phrases, we expect an even larger performance gap between the HMM and GMM based techniques for longer phrases.

These experiments show that the performance of the proposed method is consistently better than the GMM-based one, even though the HMM-based system has a slightly smaller number of parameters (936 vs. 1024 Gaussian components). The fusion of both GMM- and HMM-based methods worked as expected: for Imposter-Correct and Imposter-Wrong trials, the performance of the fusion system improved, while for Target-Wrong it degraded (especially for males) because of the much worse performance of the GMM-based subsystem.

C. The Effects of the i-Vector Dimensionality

The aim of these experiments is to show the effect of i-vector dimensionality on the performance of the proposed i-vector/HMM method. In this part, the RWCCN and s-norm were used in all experiments.

From Fig. 6, it is clear that the performance of the proposed method is improved by increasing the i-vector dimension for both genders. The best results come from i-vector with 1200 dimensions. The reduction in all criteria (EER, $\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{old}}$ and $\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{new}}$) decreases while increasing the i-vector dimensionality and most of them saturate after 1200 dimensions. Based on the computation constraints, we selected 400 dimensional i-vectors for other experiments. Note that the performance improves with the increasing i-vector dimensionality, especially for females, and while the performance for males is quite better compared to females for low dimensional i-vector, the trend reverses for higher dimensional i-vectors.

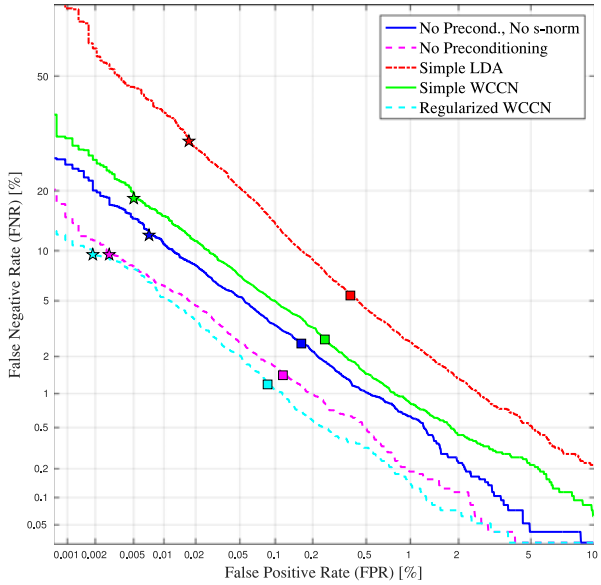

Fig. 7. DET curves of different preconditioning transformations for Imposter-Correct as non-target trials of males. From the worst plot to the best: 1) Simple LDA, 2) Simple WCCN, 3) No Preconditioning and No s-norm, 4) No Preconditioning, 5) Regularized WCCN.

D. Regularized WCCN vs. Simple LDA, WCCN and PLDA

In Section III-B, we explained that we cannot robustly estimate LDA and WCCN with the limited amount of training data available for the text-dependent speaker verification task. In order to prove it experimentally, we compare the performance of LDA, WCCN and RWCCN in the first half of Table VI and in Figs. 7 and 8. We also present the results without any preconditioning (i.e., without LDA, WCCN or RWCCN). The results indicate that using simple LDA and WCCN lead to a big degradation in performance. Especially with LDA dimensionality reduction, where the i-vector dimensionality is reduced from 400 to the number of speakers minus one, it seems that lots of the important information is lost. This is in contrast with the text-independent task where LDA with dimensionality reduction is often beneficial [3], [44]. Comparison of simple WCCN and RWCCN shows the advantage of using the regularization in within-class covariance estimation. However, when the preconditioning transformation is estimated only on the small RSR2015 background set, RWCCN is effective only for males. For females, the same (or slightly better) performance can be obtained without any preconditioning. By default, s-norm is applied in all experiments. For completeness, we also present results for no preconditioning and no score normalization in the first line of Table VI showing that the application of s-norm is an important step.

TABLE VI
PERFORMANCE COMPARISON OF DIFFERENT PRECONDITIONING TRANSFORMATIONS WITH THE PROPOSED I-VECTOR/HMM METHOD FOR IMPOSTER-CORRECT AS NON-TARGET TRIALS

Training Data	Method	Male			Female		
		EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
Background (97 speakers)	No Precond., No s-norm	0.78	0.0413	0.1925	1.07	0.0497	0.2060
	No Preconditioning	0.49	0.0257	0.1233	**0.47**	**0.0248**	**0.1284**
	Simple LDA	1.65	0.0925	0.4886	2.24	0.1537	0.6574
	Simple WCCN	0.92	0.0515	0.2357	1.34	0.0844	0.4022
	Regularized WCCN	**0.37**	**0.0204**	**0.1142**	0.49	0.0275	0.1533
	PLDA	1.92	0.1106	0.4426	1.91	0.1010	0.4653
Background, Development (194 speakers)	No Preconditioning	0.48	0.0251	0.1047	0.39	0.0179	0.1041
	Simple LDA	0.52	0.0293	0.2036	0.47	0.0324	0.2492
	Simple WCCN	0.40	0.0211	0.1061	0.29	0.0173	0.0837
	Regularized WCCN	**0.29**	**0.0162**	**0.0894**	**0.28**	**0.0141**	**0.0746**
	PLDA	1.39	0.0762	0.3448	1.36	0.0759	0.3584

For LDA, the dimensionality after transformation is the number of speakers minus one. Only the background part of RSR2015 is used to train UBM and i-vector extractor in all these experiments.

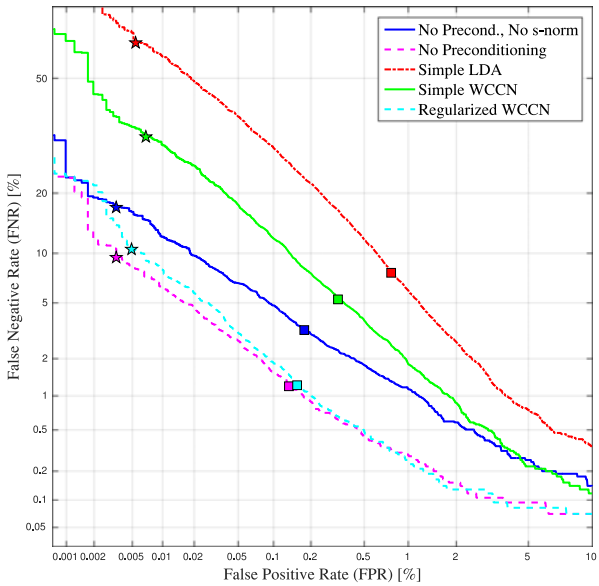

Fig. 8. DET curves of different preconditioning transformations for Imposter-Correct as non-target trials of females. From the worst plot to the best: 1) Simple LDA, 2) Simple WCCN, 3) No Preconditioning and No s-norm, 4) Regularized WCCN, 5) No Preconditioning.

For comparison, we also include results where PLDA is used for scoring rather than the default cosine distance. In this case, no preconditioning is used and PLDA is trained on the training data instead. No score normalization is used for PLDA as it was found to only degrade the performance. As can be seen, PLDA performs significantly worse compared to any of the cosine distance based systems.

To demonstrate the effects of increasing the number of training speakers, we also include experiments where the RSR2015 development set is added to the training data for the estimation of preconditioning transformations and score normalization (see the second half of Table VI). This roughly doubles the amount of such training data. We can see significant improvements for all the results where any preconditioning transformation is applied (especially for females) as the transformations can be more robustly estimated. We still benefit from the regularization in RWCCN estimation. However, the improvements compared to simple WCCN are less pronounced for the larger training set as expected. The results with no preconditioning indicate that s-norm does not significantly benefit from the increased size of the cohort set. This is also expected as only a few parameters (two means and two standard deviations) need to be estimated on the cohort set for each trial.

E. Comparison With Other State-of-the-Art Techniques

To the best of our knowledge, the best results on the RSR2015, Part-1 published prior to our work on i-vector/HMM are reported in [29], where JFA is used to extract the i-vector-like fixed-length representations of utterances. In Table VII, the line *z-vectors/JFA* shows the results for their best system (best average results for both genders from [29, Table 6]), which uses the high-dimensional residual vector $\mathbf{z}$ as the speaker representation, the cosine similarity for scoring and s-norm for score normalization. As described in section IV-A, we use the experimental setup from [29] for all our experiments, unless otherwise stated. Therefore, our results should be directly comparable with those from [29]. As can be seen from Table VII, our i-vector/HMM systems are clearly superior to the z-vectors/JFA system especially for female trials. The relative improvements obtained with the 1200 dimensional i-vector/HMM system are 50% and 67% on EER and 61% and 67% on $NDCF_{old}$ for males and females, respectively. Meanwhile, the i-vector/HMM system is simpler as it does not need any adaptation to the phrase, speaker and channel as required for the JFA system. The dimension of our i-vector is much lower than for the $\mathbf{z}$ vector (1200 vs. 30720). Since both methods use cosine similarity for scoring, ours is approximately 25 times faster and memory efficient at the scoring stage.

We also compare the i-vector/HMM approach to another two simpler and more conventional baseline methods: the line in Table VII denoted as *Rel. MAP/GMM* corresponds to the traditional approach based on Relevance MAP adaptation of GMM-UBM (1024 components) and log-likelihood ratio scoring [45], which is still the standard approach to text-dependent speaker recognition. In the case of *Rel. MAP/HMM*, the same phone-based phrase specific HMMs are used to obtain alignment of frames to Gaussian components as in the case of the i-vector/HMM system. However, the same relevance MAP

TABLE VII
COMPARISON OF THE PROPOSED METHOD WITH THE BEST RESULTS PUBLISHED ON RSR2015 PART-1 [29] AND TWO RELEVANCE MAP BASED SYSTEMS FOR IMPOSTER-CORRECT AS NON-TARGET TRIALS

	Male			Female		
Method	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
z-vector/JFA[29]	0.44	0.028	–	0.61	0.027	–
Rel. MAP/GMM	0.40	0.020	0.106	0.15	0.008	0.035
Rel. MAP/HMM	0.23	0.014	0.073	**0.12**	**0.006**	0.022
i-vector/HMM(400)	0.37	0.020	0.114	0.49	0.028	0.153
i-vector/HMM(1200)	0.22	0.011	0.075	0.20	0.009	0.033
Rel. MAP/HMM, i-vector/HMM(1200)	**0.14**	**0.009**	**0.057**	**0.12**	**0.006**	**0.018**

The last row shows score-level fusion of Rel. MAP/HMM and i-vector/HMM.

adaptation and log-likelihood ratio scoring is used as in the case of *Rel. MAP/GMM*. Although, these simpler methods lack any channel compensation, their performance is comparable to (or, for females, even slightly better than) the i-vector/HMM system.[5] Note, however, that the trends are different on RedDots data as presented in the following. Again, the advantage of the i-vector/HMM approach is that it results in much more compact representations of speaker models and test utterances compared to the Relevance MAP based techniques. Out of the two Relevance MAP based systems, *Rel. MAP/HMM* again benefits from the better HMM-based alignment and outperforms the simpler *Rel. MAP/GMM* system.

The last row of Table VII presents the ultimate, best performing score-level fusion of the *Rel. MAP/HMM* system and the proposed i-vector/HMM technique with 1200 dimensional i-vector.

F. Comparison With DNN Based Methods

In this section, we present results for models making use of neural networks. The approaches used in these experiments were originally proposed for text-independent speaker verification, and were also recently found successful for the text-dependent task. A more detailed study of these approaches in a text-dependent task can be found in our previous works [24], [26], where the result presented here are selected from. We select here only the best performing DNN based systems to show how the simpler i-vector/HMM method compares to and combines with the neural network approaches.

In order to facilitate the comparison, the first section of Table VIII simply repeats the results for the MFCC-based i-vector/HMM system from Table V. The second section corresponds to a i-vector system where, instead of the HMM model, DNN trained for senone classification defines the alignment of feature frames to Gaussian components (i.e., the DNN output defines $\gamma_t^{(c)}$ from (2) and (3)). The DNN alignment performs just comparably for Imposter-Correct trials. However, since it does not rely on the true transcription, it is not able to reject Target-Wrong trials as reliably as the HMM alignment.

The third and fourth sections of Table VIII present results for i-vector/GMM and i-vector/HMM systems where input features are MFCCs concatenated with a neural network based BN features (MFCC+BN). When the BN features are used the performance greatly improves, especially for the Target-Wrong trials. Since the BN features are trained for phone discrimination, they produce very phrase specific i-vectors, which are very good for discrimination between phrases and, therefore, for rejecting wrong phrase trials. We still benefit from using the HMM alignment (rather than the GMM one), however, the improvement is rather small compared to the case with only MFCC features.

The last two sections of Table VIII again show performance with the concatenated MFCC+BN features. This time, however, the old fashioned relevance MAP adaptation with log-likelihood scoring is used. Just like in the case i-vector systems with BN features, we do not really benefit from the HMM alignment. With the exception of female Imposter-Correct condition, the i-vector systems with BN features performs better (or at least the same) compared to the relevance MAP based ones. Therefore, we can benefit from using the more compact i-vector representation without scarifying the performance.

From the presented results, it might seem that the best option is to use the MFCC+BN features and the i-vector system with the simpler GMM based alignment. However, as we will later see in Section V-I, which is deals with the experiments on RedDots data, BN features can fail to provide good performance on the most difficult Imposter-Correct. On this condition, the BN features perform well only if the data for training UBM and i-vector extractor contains the same closed set of phrases as used in the evaluation data (i.e., phrases in the enrollment and test utterances). This is the case for our experiments on RSR2015 data, but not for RedDots. See [26] for a more detailed analysis of this problem. Note that such a constraint might be too limiting when building practical systems. Furthermore, we might not be able to train suitable neural network for BN feature extraction in the case of building a speaker verification system for some of the low-resource language. In such cases, the i-vector/HMM based system trained only on MFCC can be the best option.

G. Gender Dependent (GD) and Gender Independent (GI) Training of the UBM and i-Vector Extractor

The aim of these experiments is to compare the performance of the proposed methods based on the amount of training data, in the following scenarios:

1) Train both UBM and i-vector extractor as gender independent.
2) Train both as independent, and add development set to their training data.
3) Train both of them as gender dependent.
4) Train both of them as gender dependent and add development set to their training data.

[5] In general, channel compensation techniques are known to be ineffective on RSR2015 data, where the channel variability is very limited.

TABLE VIII
COMPARISON WITH DNN BASED METHODS (DNN ALIGNMENT, CONCATENATED MFCC+BN FEATURES)

Method	Non-Target Trial Type	Male			Female		
		EER [%]	$\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{old}}$	$\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{new}}$	EER [%]	$\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{old}}$	$\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{new}}$
MFCC i-vector/HMM	Imposter-Correct	0.37	0.0204	0.1142	0.49	0.0275	0.1533
	Target-Wrong	0.30	0.0162	0.1111	0.13	0.0072	0.0620
	Imposter-Wrong	0.03	0.0013	0.0088	0.06	0.0011	0.0045
MFCC i-vector/DNN	Imposter-Correct	0.36	0.0198	0.1196	0.38	0.0232	0.1562
	Target-Wrong	1.13	0.0806	0.5709	0.42	0.0284	0.2133
	Imposter-Wrong	0.03	0.0025	0.0224	0.03	0.0013	0.0077
MFCC+BN i-vector/GMM	Imposter-Correct	0.31	0.0176	0.0955	0.28	0.0144	0.0898
	Target-Wrong	0.08	0.0054	0.0330	**0.07**	0.0025	0.0236
	Imposter-Wrong	**0.01**	**0.0002**	**0.0020**	**0.02**	**0.0004**	**0.0008**
MFCC+BN i-vector/HMM	Imposter-Correct	**0.30**	**0.0148**	**0.0927**	0.27	0.0134	0.0809
	Target-Wrong	**0.07**	**0.0042**	**0.0295**	0.09	0.0026	0.0114
	Imposter-Wrong	**0.01**	0.0005	0.0024	0.03	0.0008	0.0021
MFCC+BN Rel. MAP/GMM	Imposter-Correct	0.31	0.0161	0.0998	**0.17**	**0.0091**	**0.0405**
	Target-Wrong	0.29	0.0102	0.0322	0.09	0.0043	0.0227
	Imposter-Wrong	0.17	0.0050	0.0123	0.03	0.0009	0.0027
MFCC+BN Rel. MAP/HMM	Imposter-Correct	0.36	0.0193	0.1253	**0.17**	0.0108	0.0523
	Target-Wrong	0.26	0.0110	0.0367	**0.07**	**0.0024**	**0.0099**
	Imposter-Wrong	0.13	0.0037	0.0097	0.03	0.0009	0.0017

The results are reported on RSR2015 dataset for three types of non-target trials. The first column specifies the used features and model.

TABLE IX
COMPARISON OF GENDER DEPENDENT (GD) AND GENDER INDEPENDENT (GI) TRAINING FOR IMPOSTER-CORRECT AS NON-TARGET TRIALS

Gender	Strategy	Use Dev	EER [%]	$\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{old}}$	$\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{new}}$
Male	GI	×	0.37	0.0204	0.1142
	GI	✓	**0.26**	0.0137	**0.0575**
	GD	×	0.54	0.0255	0.1172
	GD	✓	0.32	**0.0136**	0.0687
Female	GI	×	0.49	0.0275	0.1533
	GI	✓	**0.23**	**0.0126**	0.0850
	GD	×	0.60	0.0315	0.1552
	GD	✓	0.34	0.0136	**0.0723**

The third column says whether the development set is used for training or not.

The results in Table IX show that the number of speakers in the training data is more crucial for the system performance than gender dependency. The performance improved considerably by increasing the number of speakers for both training strategies. In case these two strategies used approximately the same amount of training data (compare rows 1 and 4), it is better to use gender dependent modeling; however, if there is not enough training data for both genders, gender independent modeling is preferred.

Comparing males and females results shows that adding new data is more effective for females than males, similar to Table VI where the amount of data for training the preconditioning transformations was investigated.

H. Out of Domain Training

The last question we sought to resolve is whether it is possible to use another dataset to train the models and to handle the preconditioning transformations and score normalization as we did in the text-independent case. Several experiments were done in different scenarios using LibriSpeech dataset. Testing was performed on RSR.

Table X contains four sections. The first one (RSR dataset used for both UBM/i-vector and transformations/normalization training) is borrowed from Table VI for ease of comparison. The second section shows a scenario in which UBM and i-vector extractor are trained on a heterogeneous text-independent dataset (LibriSpeech), while the target RSR dataset is used for RWCCN and s-norm training. Comparing these two sections shows that the performance does not change much: we have a slight degradation for males, while the performance improved a little for females. Without any transformation and normalization, the performance is obviously degraded. These results prove that we can have a text-dependent system built independently of users' pass-phrases. However, we have to record some utterances from each pass-phrase for re-estimating the transformations and score normalization, or we can accept performance degradation at the price of having a totally phrase-independent system.

In the third section, we use both RSR and LibriSpeech datasets for UBM and i-vector extractor training. Comparing these results with previous sections shows that increased training data, even if from text-independent datasets, can improve the performance. Similar to other experiments, adding new training data is more effective for females.

The last section of Table X shows the most pessimistic scenario, in which we have no access to target data at all. All training (UBM and i-vector extractor training, RWCCN and s-norm parameter estimation) is done on LibriSpeech in a text-independent manner (i.e., we also use phrase-independent RWCCN and s-norm in this case). The results show that the text-dependent speaker verification performance drops significantly with channel compensation and score normalization method trained in a text-independent (phrase-independent) manner. The main reason for this is the short duration of the utterances causing significant variations of the phonetic content. Comparing the last row

TABLE X
TRAINING ON HETEROGENEOUS DATA

Training Data	Transformation Data	Method	Male			Female		
			EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
RSR	–	–	0.79	0.0413	0.1925	1.06	0.0497	0.2060
	RSR	s-norm, RWCCN	**0.37**	0.0204	0.1142	0.49	0.0275	0.1533
Libri	–	–	1.08	0.0532	0.2039	1.50	0.0686	0.2589
	RSR	s-norm, RWCCN	0.50	0.0239	0.1242	0.43	0.0233	**0.1136**
RSR, Libri	–	–	0.97	0.0482	0.1875	1.52	0.0692	0.2454
	RSR	s-norm, RWCCN	0.40	**0.0193**	**0.0942**	**0.36**	**0.0213**	0.1193
Libri	Libri	RWCCN	1.17	0.0559	0.2032	1.51	0.0684	0.2693
	Libri	s-norm	1.29	0.0635	0.2676	1.57	0.0733	0.2831
	Libri	s-norm, RWCCN	1.17	0.0595	0.2481	1.36	0.0656	0.2443

The first column specifies the data used for training the models and i-vector extractor, the second column details the data used for preconditioning transformation and score normalization training. These results are reported on RSR2015 dataset for Imposter-Correct as non-target trials.

of this table (everything done on LibriSpeech) with the third one (UBM and i-vector extractor trained on LibriSpeech, no channel compensation, no normalization), we see a little degradation for males and minor improvement for females.

The LibriSpeech results have shown that for a text-dependent speaker verification with good performance, adding a small amount of data from the target domain for any processing steps (adaptation, channel modeling and compensation and score normalization) is beneficial. At the same time, using text-independent data allows us to acceptably collect less training data.

I. Results on RedDots Data

The results on RSR2015 encouraged us to test the proposed technique on a more challenging dataset — RedDots. Table XI compares four systems introduced in previous sections. The results are again reported separately for the three conditions corresponding to three types of non-target trials.

The RedDots evaluation data comes without any development set, which would contain recordings of the same phrases as used for the enrollment and test. Therefore, we have to use training data from other datasets (RSR2015 and LibriSpeech) with mismatched phrases. In section V-H, we have shown that such a mismatch makes the channel compensation and score normalization techniques ineffective in the case of text-dependent speaker verification. Therefore, all the reported results for the RedDots dataset are based on the simple cosine distance scoring and without any score normalization. 600-dimensional i-vectors are used in these experiments.

First, let us focus on the first half of Table XI corresponding to the systems trained on the simple MFCC features. As before, the relevance MAP based systems use likelihood ratio verification scores. The results on Imposter-Correct trials show that the best performance is achieved with HMMs — in most cases, the *Rel. MAP/HMM* system is the best one followed by the proposed i-vector/HMM.

Target-wrong trials are more interesting — the gap between HMM-based and GMM-based methods is considerably wider. As we saw in the RSR2015 data (Table V), the HMM forced alignment makes rejecting these trials much easier. Contrary to Imposter-Correct trials, *Rel. MAP/HMM* does not work well compared to *i-vector/HMM*. The main reason is the likelihood ratio scoring: when the HMM is used directly for calculating likelihoods, both the speaker model and UBM likelihoods obtained for the wrong phrase are low, their ratio is unreliable and causes a higher false acceptance rate. With the i-vector/HMM model, wrong HMM alignment simply produces a wrong i-vector which is easily rejected.

TABLE XI
COMPARISON OF THE PROPOSED METHOD WITH GMM ALIGNMENT AND TWO RELEVANCE MAP BASED SYSTEMS ON PART-01 MALES OF REDDOTS

Method	Trial Type	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
MFCC i-vector/GMM	Imp-Corr	2.07	0.0899	0.3105
	Tar-Wrg	3.76	0.1762	0.4275
	Imp-Wrg	0.43	0.0153	0.0435
MFCC i-vector/HMM	Imp-Corr	1.88	0.0809	0.2271
	Tar-Wrg	1.11	0.0338	0.0509
	Imp-Wrg	0.46	0.0106	0.0228
MFCC Rel. MAP/GMM	Imp-Corr	1.98	0.0848	0.2879
	Tar-Wrg	4.01	0.1733	0.4960
	Imp-Wrg	0.34	0.0135	0.0488
MFCC Rel. MAP/HMM	Imp-Corr	**1.48**	**0.0613**	**0.1722**
	Tar-Wrg	1.57	0.0567	0.1491
	Imp-Wrg	0.59	0.0137	0.0361
MFCC+BN i-vector/GMM	Imp-Corr	3.05	0.1385	0.5002
	Tar-Wrg	0.56	0.0226	0.0515
	Imp-Wrg	**0.19**	0.0045	0.0071
MFCC+BN i-vector/HMM	Imp-Corr	2.92	0.1295	0.4246
	Tar-Wrg	**0.37**	0.0081	0.0154
	Imp-Wrg	0.22	**0.0036**	**0.0059**
MFCC+BN Rel. MAP/GMM	Imp-Corr	2.59	0.1295	0.4423
	Tar-Wrg	0.46	0.0155	0.0549
	Imp-Wrg	0.28	0.0047	0.0201
MFCC+BN Rel. MAP/HMM	Imp-Corr	2.19	0.1019	0.3906
	Tar-Wrg	0.43	**0.0068**	**0.0068**
	Imp-Wrg	0.22	0.0046	0.0105

600-dimensional i-vectors were used for both i-vector based systems. The first half of this table reports results on the simple MFCC features while the second half of it shows results of similar systems with MFCC concatenated to BN features (i.e., MFCC+BN).

The second half of Table XI shows results for similar systems. This time, however, the systems are trained on the concatenated MFCC+BN features. The BN features greatly improve performance again on the Target-Wrong trials for all used models. However, as already pointed out in section V-F, BN features fail to provide good performance on the most difficult and most

TABLE XII
COMPARISON OF SPEED OF DIFFERENT SYSTEMS ON 2 sec TEST SEGMENT

Method	i-vector HMM(MFCC) [ms]	i-vector GMM(MFCC+BN) [ms]	Rel. MAP GMM(MFCC+BN) [ms]
Features (2 s)	4.2	546.7	546.7
Statistics (2 s)	33.1	46.7	46.7
i-vector	35.9	39.4	–
Scoring	0.05	0.05	19.9
Overall (2 s)	73.2	632.8	613.3

important Imposter-Correct trials as the data for training UBM and i-vector extractor do not contain the same phrases as used for the evaluation. See [26] for a more detailed analysis of this problem.

J. Speed Comparison

Table XII compares the speed of selected speaker verification systems as measured using our speed optimized Matlab implementation on Intel Xeon CPU E5-2670 (2.60 GHz). We report the time in milliseconds spent on verifying one 2 sec long utterance, which is about the average length of test segments in the RSR2015 database. We further break the timing down into the individual phases of the verification process, which should allow the reader to get a good idea about the speed of any system described in this paper. In the table, we use (2 s) to mark phases that (linearly) depend on the duration of the test segment.

As can be seen, the simple MFCC based i-vector/HMM system, which still performs very competitively, is an order of magnitude faster than any system which makes use of BN features (or DNN alignment). BN features are very costly to extract.

We have shown that the proposed i-vector based system also provides competitive verification performance when compared the more conventional relevance MAP GMM-UBM (or HMM-UBM) systems. The relevance MAP based systems, however, do not allow for i-vector-like compact speaker representations, which also results in about two orders of magnitude slower scoring phase. This might pose a problem for an application where the same test segment needs to be scored against many speaker models. For a test segment, all the phases needs to be executed only once except for the scoring phase, which needs to be evaluated for each speaker model.

Note that the speed of the relevance MAP based system is reported for the case of an approximate fast linear scoring (see [46, eq. (20)] without the term $\mathbf{Vy}$). The full frame-by-frame evaluation of a speaker model (and also all models that form the s-norm cohort) would take more than 4 sec for one 2 sec test segment (i.e., it would be another order of magnitude slower).

VI. CONCLUSION

In this paper, we proposed a new HMM structure for text-dependent speaker verification, enabling us to use the potential of the HMM to model time sequences along with the established i-vector technique. We first trained a phoneme recognition system and then used its models to build a model for each phrase. With this HMM modeling, we could train a single phrase-independent i-vector extractor for all phrases. We also empirically showed the advantages of this method over GMM that is commonly used in text-independent and text-dependent speaker verification, mainly the ability to reject target-wrong trials. The Viterbi forced alignment produces invalid statistics for such trials and consequently they are rejected easily.

We explained and showed that due to a limited number of speakers, simple LDA and WCCN cannot be used for the text-dependent task. We suggested a regularized version of WCCN to solve this problem, and obtained better results with the proposed i-vector/HMM method.

We have performed comprehensive experiments addressing several aspects of the proposed method. We have found that the performance of the various cepstral features are not considerably different for the text-dependent task and investigated into feature fusion in the score domain.

Although the results of the i-vector/HMM are much better than those of i-vector/GMM, their fusion can still reduce the EER by 34% percent on average for imposter-correct trials.

We obtained our best results for the single system with 1200 dimensional i-vectors using RWCCN. Compared to the best published results on Part-1 of RSR [29], our technique can reduce the EER by 50% and 67% and $\mathrm{NDCF}_{\mathrm{old}}$ by 61% and 67% relative for males and females, respectively.

We have compared and combined the proposed technique with DNN-based approaches to speaker verification. For Target-Wrong trials, HMM-based alignment outperforms the one based on DNN. On the contrary, while BN features provide superior performance for rejecting Target-Wrong trials, they might fail on Imposter-Correct condition as demonstrated in the RedDots data set.

Experiments on out-of-domain training data show that we can use text-independent datasets to improve the performance, but we cannot use them for channel compensation and score normalization (as is usual in the text-independent case).

REFERENCES

[1] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 5, pp. 980–988, Jul. 2008.

[2] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[4] S. Ioffe, "Probabilistic linear discriminant analysis," in *Computer Vision—ECCV 2006*. New York, NY, USA: Springer, 2006, pp. 531–542.

[5] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE 11th Int. Conf. Comput. Vision*, 2007, pp. 1–8.

[6] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Proc. Odyssey*, 2010, p. 14.

[7] A. Larcher *et al.*, "Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7673–7677.

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

[9] H. Aronowitz, "Text dependent speaker verification using a small development set," in *Proc. Odyssey—The Speaker Lang. Recog. Workshop*, 2012, pp. 312–316.

[10] H. Aronowitz and O. Barkan, "On leveraging conversational data for building a text dependent speaker verification system," in *Proc. INTERSPEECH*, 2013, pp. 2470–2473.

[11] S. Novoselov, T. Pekhovsky, A. Shulipa, and A. Sholokhov, "Text-dependent GMM-JFA system for password based speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 729–737.

[12] T. Stafylakis, P. Kenny, M. J. Alam, and M. Kockmann, "Speaker and channel factors in text-dependent speaker recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 1, pp. 65–78, Jan. 2016.
[13] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. INTERSPEECH*, 2006, pp. 1471–1474.
[14] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 1695–1699.
[15] G.-R. D., Z. X., M. A., and P. D., "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *Proc. Spoken Lang. Technol. Workshop*, 2014, pp. 378–383.
[16] D. Garcia-Romero and A. McCree, "Insights into deep neural networks for speaker recognition," in *Proc. INTERSPEECH*, 2015, pp. 1141–1145.
[17] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Proc. Odyssey-The Speaker Lang. Recog. Workshop*, 2014, pp. 293–298.
[18] F. Grezl, M. Karafiát, and L. Burget, "Investigation into bottle-neck features for meeting speech recognition," in *Proc. INTERSPEECH*, 2009, pp. 2947–2950.
[19] S. Yaman, J. Pelecanos, and R. Sarikaya, "Bottleneck features for speaker recognition," in *Proc. Odyssey-The Speaker Lang. Recog. Workshop*, 2012, vol. 12, pp. 105–108.
[20] P. Matejka *et al.*, "Neural network bottleneck features for language identification," in *Proc. Odyssey—The Speaker Lang. Recog. Workshop*, 2014, pp. 299–304.
[21] K. Vesely, M. Karafiát, F. Grezl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proc. Spoken Lang. Technol. Workshop*, 2012, pp. 336–341.
[22] P. Matejka *et al.*, "Analysis of DNN approaches to speaker identification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 5100–5104.
[23] F. Richardson, D. A. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," in *Proc. INTERSPEECH*, 2015, pp. 1146–1150.
[24] H. Zeinali, L. Burget, H. Sameti, O. Glembek, and O. Plchot, "Deep neural networks and hidden Markov models in i-vector-based text-dependent speaker verification," in *Proc. Odyssey-The Speaker Lang. Recog. Workshop*, 2016. pp. 24–30.
[25] H. Zeinali, H. Sameti, L. Burget, J. Cernocky, N. Maghsoodi, and P. Matejka, "i-vector/HMM based text-dependent speaker verification system for reddots challenge," in *Proc. INTERSPEECH*, 2016, pp. 440–444.
[26] H. Zeinali, H. Sameti, and Č. J. Burget, Lukáš, "Text-dependent speaker verification based on i-vectors, deep neural networks and hidden Markov models," *Comput. Speech Lang.*, submitted for publication.
[27] 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.
[28] P. Kenny, T. Stafylakis, P. Ouellet, and M. J. Alam, "JFA-based front ends for speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 1705–1709.
[29] P. Kenny, T. Stafylakis, J. Alam, P. Ouellet, and M. Kockmann, "Joint factor analysis for text-dependent speaker verification," in *Proc. Odyssey—The Speaker Lang. Recog. Workshop*, 2014, pp. 200–207.
[30] Y. Kin, J. Mason, and J. Oglesby, "Speaker recognition using hidden Markov models, dynamic time warping and vector quantisation," *IEE Proc. Vision, Image Signal Process.*, no. 5, pp. 313–318, 1995.
[31] C. ChiWei, Q. Lin, and D.-S. Yuk, "An HMM approach to text-prompted speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1996, pp. 673–676.
[32] D. T. Toledano, C. Esteve Elizalde, and J. Gonzalez-Rodriguez, "Phoneme and sub-phoneme t-normalization for text-dependent speaker recognition," in *Proc. Odyssey—The Speaker Lang. Recog. Workshop.*, International Speech Communication Association, 2008, paper 029.
[33] C. Dong, Y. Dong, J. Li, and H. Wang, "Support vector machines based text dependent speaker verification using HMM supervectors," in *Proc. Odyssey—The Speaker Lang. Recog. Workshop*, 2008, paper 031.
[34] A. Larcher, J.-F. Bonastre, and J. S. Mason, "Constrained temporal structure for text-dependent speaker verification," *Digital Signal Process.*, vol. 23, no. 6, pp. 1910–1917, 2013.
[35] T. Stafylakis, P. Kenny, J. Alam, and M. Kockmann, "JFA for speaker recognition with random digit strings," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 190–194.
[36] H. Zeinali, E. Kalantari, H. Sameti, and H. Hadian, "Telephony text-prompted speaker verification using i-vector representation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4839–4843.
[37] Y. Tian, M. Cai, L. He, and J. Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification," in *Proc. INTERSPEECH*, 2015, pp. 1151–1155.
[38] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. INTERSPEECH*, 2011, pp. 249–252.
[39] J. H. Friedman, "Regularized discriminant analysis," *J. Amer. Statist. Assoc.*, vol. 84, no. 405, pp. 165–175, 1989.
[40] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5206–5210.
[41] K. A. Lee *et al.*, "The RedDots data collection for speaker recognition," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 2996–3000.
[42] S. Young *et al.*, *The HTK book*. Cambridge, U.K.: Entropic Cambridge Research Laboratory, 1997, vol. 2.
[43] M. Karafiát, F. Grézl, K. Veselý, M. Hannemann, I. Szőke, and J. Černocký, "BUT 2014 Babel system: Analysis of adaptation in NN based systems," in *Proc. INTERSPEECH*, 2014, pp. 3002–3006.
[44] P. Matějka *et al.*, "Full-covariance ubm and heavy-tailed plda in i-vector speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 4828–4831.
[45] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Process.*, vol. 10, no. 1, pp. 19–41, 2000.
[46] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny, "Comparison of scoring methods used in speaker recognition with joint factor analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 4057–4060.

Hossein Zeinali received the B.Sc. degree in computer engineering from Shiraz University, Shiraz, Iran, in 2010 and the M.Sc. degree in artificial intelligence from Sharif University of Technology, Tehran, Iran, in 2012. He is currently working toward the Ph.D. degree in artificial intelligence at Sharif University of Technology. His research interests include speech processing, speaker recognition, and natural language processing.

Hossein Sameti was born in Tehran, Iran, in 1961. He received the Ph.D. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1994. In 1995, he joined the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, where he is an Associate Professor now. There, he has also served as the Department Chair. He founded Speech Processing Lab (SPL) in the department in 1998 and is the supervisor of the lab. SPL has developed Nevisa, the first Persian continuous speech recognition engine, which is a commercial product now. His current research interests include speech and language processing, automatic speech recognition, speech synthesis, speech enhancement, spoken dialogue systems, spoken language understanding, speaker identification and verification, and spoken term detection.

Lukáš Burget is an assistant professor at the Faculty of Information Technology, Brno University of Technology (FIT BUT) and Research Director of the BUT Speech@FIT group. He was a Visiting Researcher at OGI Portland, OR, USA and at SRI International, Menlo Park, USA. His scientific interests are in the field of speech data mining, concentrating on acoustic modeling for speech, speaker, and language recognition, including their software implementations. He was on numerous EU- and US-funded projects, was the PI of US-Air Force EOARD project and BUT's PI in IARPA BEST.