Available online at www.sciencedirect.com

ScienceDirect

Computer Speech & Language 46 (2017) 53–71

www.elsevier.com/locate/csl

Text-dependent speaker verification based on i-vectors, Neural Networks and Hidden Markov Models☆

Hossein Zeinali[a,b,*], Hossein Sameti[a], Lukáš Burget[b], Jan "Honza" Černocký[b]

[a] *Speech Processing Laboratory, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran*
[b] *Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic*

Received 10 October 2016; received in revised form 20 February 2017; accepted 25 April 2017
Available online 12 May 2017

Abstract

Inspired by the success of Deep Neural Networks (DNN) in text-independent speaker recognition, we have recently demonstrated that similar ideas can also be applied to the text-dependent speaker verification task. In this paper, we describe new advances with our state-of-the-art i-vector based approach to text-dependent speaker verification, which also makes use of different DNN techniques. In order to collect sufficient statistics for i-vector extraction, different frame alignment models are compared such as GMMs, phonemic HMMs or DNNs trained for senone classification. We also experiment with DNN based bottleneck features and their combinations with standard MFCC features. We experiment with few different DNN configurations and investigate the importance of training DNNs on 16 kHz speech. The results are reported on RSR2015 dataset, where training material is available for all possible enrollment and test phrases. Additionally, we report results also on more challenging RedDots dataset, where the system is built in truly phrase-independent way.

Keywords: Deep Neural Network; Text-dependent; Speaker verification; i-Vector; Frame alignment; Bottleneck features

1. Introduction

During the Deep last decade, *text-independent* speaker recognition technology has been largely improved in terms of both computational complexity and accuracy. Channel-compensation techniques, such as Joint Factor Analysis (JFA) (Kenny et al., 2008; 2007), evolved into the i-vector paradigm (Dehak et al., 2011), where each speech utterance is represented by a low-dimensional fixed-length vector. To verify a speaker identity, similarity of i-vectors can be measured as a simple cosine distance or by using a more elaborate Bayesian model such as Probabilistic Linear Discriminant Analysis (PLDA) (Prince and Elder, 2007; Kenny, 2010).

Recently, there has been an increased effort in applying these techniques also to the problem of *text-dependent* speaker verification, where not only the speaker of the test utterance but also the (typically very short) uttered phrase have to match with the enrollment utterance in order to get the utterance correctly accepted (see Table 1 for types of

☆ This paper has been recommended for acceptance by Roger Moore.

* Corresponding author at: Speech Processing Laboratory, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran.
E-mail addresses: zeinali@ce.sharif.edu, hsn.zeinali@gmail.com (H. Zeinali), sameti@sharif.edu (H. Sameti), burget@fit.vutbr.cz (L. Burget), cernocky@fit.vutbr.cz (J.H. Černocký).

http://dx.doi.org/10.1016/j.csl.2017.04.005
0885-2308/

Table 1
Trial types in text-dependent speaker verification (Larcher et al., 2014). We can report performance of the text-dependent systems based on the combination of Target-Correct trials with one of the other trial types (or all other types).

	Target speaker	Imposter speaker
Correct pass-phrase	Target-Correct	Imposter-Correct
Wrong pass-phrase	Target-Wrong	Imposter-Wrong

errors). A typical application is a voice-based access control. Unfortunately, the techniques used for *text-independent* speaker recognition were initially found 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) (Larcher et al., 2013; 2012) or NAP compensated GMM mean super-vector scored using a SVM classifier (Aronowitz, 2012; Novoselov et al., 2014). Only limited success was observed with i-vectors/PLDA (Larcher et al., 2014; Stafylakis et al., 2013) or with JFA, which mainly served as an i-vector-like feature extraction method (Kenny et al., 2014b; 2014c).

In Zeinali et al. (2015), we proposed a Hidden Markov Model (HMM) based i-vector approach for *text-prompted* speaker verification, where the phrases are composed of a limited predefined set of words. In this approach, an HMM is trained for each word. For each enrollment or test utterance, word specific HMMs are concatenated into a phrase specific HMM. This HMM is then used to collect sufficient statistics for i-vector extraction instead of the conventional GMM-UBM. This HMM based approach was further extended to *text-dependent* task in Zeinali et al. (2017), where the HMMs are trained for individual phonemes rather than words. Given the known transcriptions of enrollment and test utterances, the phrase-specific HMMs are constructed from the phoneme HMMs. Note 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 a better alignment of frames to Gaussian components as constrained by the HMM model. This approach was found to provide state-of-the-art performance on the RSR2015 dataset (Larcher et al., 2014). However, its drawback is that we need to know the phrase specific phone sequence for constructing the corresponding HMM.

More recently, techniques that make use of DNNs have been devised to improve *text-independent* speaker verification. In one of them, 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 (Lei et al., 2014; Garcia-Romero et al., 2014; Garcia-Romero and McCree, 2015; Dahl et al., 2012; Hinton et al., 2012; Kenny et al., 2014a). In this work, we experiment with the DNN-based alignment in the context of *text-dependent* speaker verification. We are mainly interested in comparing this method with the aforementioned i-vector method (Zeinali et al., 2017) relying on the HMM alignment. Note that, unlike in the HMM-based method, we do not need the phrase phonetic transcription in order to obtain the DNN alignment.

Another DNN-based approach, successful in *text-independent* speaker verification—as well as in other fields of speech processing (Grezl et al., 2009; Yaman et al., 2012; Matejka et al., 2014; Vesely et al., 2012; Matejka et al., 2016)—is using DNNs 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 (Richardson et al., 2015; Matejka et al., 2016). Such features are then used as the input to a usual i-vector based system. The good speaker recognition performance with such BN features is somewhat counter-intuitive as the DNN trained for phone classification should learn to suppress the “unimportant” speaker related information. However, it seems that a GMM-UBM trained on such BN features partitions the feature space into phone-like clusters. This seems to be important for the good speaker recognition performance just like in the case of the previously mentioned DNN approach (Lei et al., 2014), where the feature space partitioning is performed directly by the DNN outputs. This hypothesis is in agreement with the analysis in Matejka et al. (2016), where the best performance was obtained with standard i-vector system, where BN features were concatenated with standard MFCCs. While the BN features guaranteed good feature space partitioning, MFCCs contributed with the speaker information that may have been suppressed in BN feature extraction.

Although BN features can partition the feature space well, we still have to use MFCCs together with BN features to achieve the best performance. Another method of using BN features is BN Alignment (BNA) (Tian et al., 2015; Matejka et al., 2016): similarly to the DNN alignment described above, a GMM-UBM trained on BN features is used to align speech frames to Gaussian components, while another feature set is used to collect the sufficient statistics for i-vector extraction. This method will be explained in detail in Section 4.4.

For completeness (although not studied in this work), let us mention that DNNs have also been used to extract speaker identity vectors in a more direct way (compared to the DNN based i-vectors) (Variani et al., 2014; Heigold et al., 2016; Liu et al., 2015) or to classify i-vectors in a speaker recognition task (Ghahabi and Hernando, 2014).

In this paper, we verify that BN features—combined with MFCC features—provide an excellent performance also for *text-dependent* speaker verification. Although the BN features are already expected to provide good alignment, we show that further improvement can be obtained when combined with the HMM-based i-vector extraction. To our knowledge, this method provides the best performance obtained with a single i-vector based system on RSR2015 data. We investigate two scenarios: (1) all evaluation phrases are seen in the training data (i.e. RSR2015), (2) most of the evaluation phrases do not appear in the training data (i.e. RedDots). We report results for both scenarios as our previous experiments have shown that the performance of DNN based systems can differ from one scenario to another (Zeinali et al., 2016b).

This paper is an extension of our previous conference paper presented in Odyssey 2016 (Zeinali et al., 2016a). It provides more extensive presentation and analysis of the results and brings up the following issues not investigated in Zeinali et al. (2016a):

- Performance of DNNs trained on 16 kHz and 8 kHz data is compared in the text-dependent speaker verification task.
- Performances of different DNNs configurations (namely numbers of senones used and DNN targets) are compared in the text-dependent speaker verification task.
- Investigation into Bottleneck Alignment (BNA).
- Beside Imposter-Correct trials, results on Target-Wrong trials are also included as this trial type is very important in text-dependent speaker verification (see Table 1 for trial types).
- In addition to RSR2015, all results are reported also on RedDots (Zeinali et al., 2016b).

The rest of this article is organized as follows: in Section 2, we introduce i-vectors and the corresponding scoring methods. Bottleneck features and network topologies are described in Section 3. In Section 4, we show different frame alignment methods and in Section 5, the experimental setups and datasets are presented. Section 6 reports the results and finally, the conclusions of this study are given in Section 7.

2. i-Vector based system

2.1. General i-vector extraction

Although thoroughly described in the literature, let us review the basics of i-vector extraction. The main principle is that the utterance-dependent Gaussian Mixture Model (GMM) super-vector of concatenated mean vectors $\mathbf{s}$ is modeled as

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

where $\mathbf{m} = [\boldsymbol{\mu}^{(1)'}, \ldots, \boldsymbol{\mu}^{(C)'}]'$ is the GMM-UBM mean super-vector (of C components), $\mathbf{T} = [\mathbf{T}^{(1)'}, \ldots, \mathbf{T}^{(C)'}]'$ is a low-rank matrix representing M bases spanning subspace with 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 ϕ is the Maximum a Posteriori (MAP) point estimate of the variable $\mathbf{w}$. It maps most of the relevant information from a variable-length observation (utterance) $\mathcal{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ to a fixed-dimensional vector, where $\mathbf{x}_t$ is a feature vector corresponding to t^{th} frame of the utterance. The closed-form solution for computing the i-vector can be estimated as a function of the *zero-* and *first-order statistics*: $\mathbf{n}_\mathcal{X} = [N_\mathcal{X}^{(1)}, \ldots, N_\mathcal{X}^{(C)}]'$ and $\mathbf{f}_\mathcal{X} = [\mathbf{f}_\mathcal{X}^{(1)'}, \ldots, \mathbf{f}_\mathcal{X}^{(C)'}]'$, where

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

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

where $\gamma_t^{(c)}$ is the posterior (or occupation) probability of frame $\mathbf{x}_t$ 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 these variables can be computed either using the GMM-UBM or using a separate model (Lei et al., 2014; Tian et al., 2015; Matejka et al., 2016). In this work, we compare the standard GMM-UBM frame alignment with BNA, HMM and DNN-based approaches, described in the following sections. The i-vector is then expressed as

$$\boldsymbol{\phi}_{\mathcal{X}} = \mathbf{L}_{\mathcal{X}}^{-1} \overline{\mathbf{T}}' \overline{\mathbf{f}}_{\mathcal{X}} \,, \tag{4}$$

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

$$\mathbf{L}_{\mathcal{X}} = \mathbf{I} + \sum_{c=1}^{C} N_{\mathcal{X}}^{(c)} \overline{\mathbf{T}}^{(c)'} \overline{\mathbf{T}}^{(c)} \,, \tag{5}$$

with the 'bar' symbols denoting normalized variables:

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

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

where $\boldsymbol{\Sigma}^{(c)-\frac{1}{2}}$ is the square root (or another 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 computed via the same alignment as used in Eqs. (2) and (3).

2.2. i-Vector normalization and scoring

We used several different i-vector normalizations. In our experiments on RSR2015, i-vectors are length-normalized (Garcia-Romero and Espy-Wilson, 2011), and further normalized using phrase-dependent regularized Within-Class Covariance Normalization (WCCN) (Hatch et al., 2006). In the case of standard WCCN, i-vectors are transformed using the linear transformation $\boldsymbol{\Sigma}_{wc}^{-1/2}$ in order to whiten the within-class covariance matrix $\boldsymbol{\Sigma}_{wc}$, which is estimated on training data. For the *text-dependent* task, we only found WCCN effective when applied in the phrase-dependent manner (i.e. for trials of a specific phrase, $\boldsymbol{\Sigma}_{wc}$ is estimated only on the training utterances of that phrase) (Zeinali et al., 2017). With RSR2015 dataset, however, this leaves us only very limited amount of data for estimating phrase specific matrices $\boldsymbol{\Sigma}_{wc}$. For this reason, we found it necessary to regularize $\boldsymbol{\Sigma}_{wc}$ by adding a small constant to the matrix diagonal (Zeinali et al., 2017; Friedman, 1989) (i.e. adding $\alpha\mathbf{I}$ to $\boldsymbol{\Sigma}_{wc}$ where $\mathbf{I}$ is the identity matrix and α is a small constant like 0.001). We called this method Regularized WCCN (RWCCN). Simple cosine distance scoring is then used in all RSR experiments followed by phrase-dependent s-norm score normalization (Kenny, 2010).

The RedDots evaluation data comes without any development set, which would contain recordings of the same phrases as used for enrollment and test. Therefore, we have to use training data from other datasets with mismatched phrases. In Zeinali et al. (2017), we have shown that such mismatch makes the channel compensation and score normalization techniques ineffective for the case of *text-dependent* speaker verification with very short enrollment and test utterances. Therefore, all the reported results for the RedDots dataset are based on simple cosine distance scoring without any score normalization.

3. Bottleneck features

Bottleneck neural network refers to a DNN with a specific topology, where one of the hidden layers has 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 (Karafiát et al., 2014). This architecture is based on a cascade of two such bottleneck DNNs. Several frames of the bottleneck layer output of the first network are *stacked* in time to define contextual input features for the second DNN (hence the term Stacked Bottleneck Features). The input features to the first stage DNN are log Mel-scale filter bank outputs (36 filters for 8 kHz data and 40 filters for 16 kHz) augmented with 3 fundamental frequency features (Karafiát et al., 2014) and normalized using conversation-side based mean subtraction. The first stage DNN has 4 hidden layers (each with 1500 sigmoid units except for the 3rd linear bottleneck layer with 80 neurons) and the final softmax layer trained for classification of senone targets. The bottleneck outputs from the first stage DNN are sampled at times $t-10$, $t-5$, t, $t+5$ and $t+10$ and stacked into single 400-dimensional feature vector ($5 \times 80 = 400$), where t is the index of the current frame. The resulting features are input to the second stage DNN, which has the same topology as the first stage. With this architecture, each output is effectively extracted from 30 frames (300 ms) of the input features in the context around the current frame. The outputs from the bottleneck layer of the second stage DNN are then taken as the final output features (i.e. the features to train the i-vector model on). In all our experiments, the extracted BN features are 80-dimensional. See Matejka et al. (2016) and Karafiát et al. (2014) for more details on the exact structure. We have used this architecture as it proved to be very effective in our previous *text-independent* speaker recognition experiments (Matejka et al., 2016). However, our more recent experiments indicate that similar results can be obtained with simpler single stage bottleneck neural networks (e.g. compare results in Matejka et al., 2016; Lozano-Diez et al., 2016).

The bottleneck DNNs are trained to discriminate between triphone tied-state targets. Using a pre-trained GMM/HMM ASR system, a decision tree based clustering is used to cluster triphone states to the desirable number of targets (DNN outputs also called senones) (Karafiát et al., 2014). The same ASR system is used to force-align the data for DNN training in order to obtain the target labels. We use several different DNNs in our experiment, two of them trained on Switchboard data (8 kHz, conversational telephone speech) and the others trained using LibriSpeech dataset (16 kHz, read speech).

For 8 kHz, the primary DNN for extracting BN features is trained to classify 8802 triphone tied states (senones). The second DNN with 1011 senones is primarily intended for DNN based alignment as described in Section 4.3. For the 16 kHz case, we trained 4 DNNs (different senones counts, 920, 3512, 6198 and 9418) and used them all for extracting BN features. The network with 920 senones was used for DNN alignment as well. Unless indicated otherwise, the primary BN features extracted from the largest network (i.e. with 9418 senones) trained on 16 kHz speech data are used in all our experiments.

4. Frame alignment methods

4.1. GMM-based

The simplest and conventional alignment method uses a GMM (i.e. UBM) to align frames to Gaussian components (Reynolds et al., 2000). This method is widely used in text-independent speaker verification and also has been used in the text-dependent task (Larcher et al., 2014; Stafylakis et al., 2013). The GMM training is totally unsupervised process, so it does not use any information about speakers and phrases. However, this method completely ignores the left-to-right temporal structure of phrases, which is important for the text-dependent speaker verification, especially to reduce the vulnerability to replay attacks. GMM alignment is used as the baseline in this paper.

4.2. HMM-based

In Zeinali et al. (2017), an HMM based method is proposed for *text-dependent* speaker verification, where a phoneme recognizer is first trained with 3-state mono-phone HMMs with the state distributions modeled using GMMs. The parameters of the recognizer (i.e. transition probabilities and state distribution mixture weights, mean vectors and diagonal covariance matrices) are trained in the conventional way using the embedded Baum–Welch training (Young et al., 1997). Let F be the total number of mono-phones (i.e. 39), $S = 3F$ be the number of all states, G the number of Gaussian components per state, and $C = SG$ the number of all individual Gaussians, and let (s, g) denote the Gaussian component g in state s. Then, a new phrase-specific HMM is constructed for each phrase by

concatenating the corresponding mono-phone HMMs.[1] The Viterbi algorithm is then used to obtain the alignment of the frames to the HMM states, and within each state s, GMM alignment $\gamma_t^{(s,g)}$ is computed for each frame t. We can now re-interpret the pair (s, g) as one out of C Gaussians and we can substitute $\gamma_t^{(c)}$ in Eqs. (2) and (3) by $\gamma_t^{(s,g)}$, so that the zero and first order statistics can be written as:

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

where:

$$N_{\mathcal{X}}^{(s,g)} = \sum_t \gamma_t^{(s,g)} \tag{8}$$

$$\mathbf{f}_{\mathcal{X}}^{(s,g)} = \sum_t \gamma_t^{(s,g)} \mathbf{x}_t , \tag{9}$$

Note that, in Eqs. (8) and (9), 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. Also, it is worth mentioning that after calculating the zero- and first-order statistics for the training set, a single (phrase-independent) i-vector extractor is trained.

4.3. DNN-based

In this approach, a DNN is trained to produce frame-by-frame posterior probabilities of senones (context-dependent phone states). It is assumed that such posterior probabilities can be interpreted as the probabilistic alignment of speech frames to UBM components. These posteriors can be then directly used for i-vector extraction in Eqs. (2) and (3). As described in Section 1, for the *text-independent* task, excellent results were previously reported for this approach, which can better represent different pronunciations of the same phoneme as produced by different speakers (Lei et al., 2014).

Compared to the HMM alignment, this method does not take into account the true transcription of the desired phrase. Instead, the phonetically-aware DNN provides the alignment. Therefore, it is to be expected that the DNN-based approach provides worse performance for rejecting Target-Wrong trials as compared to the HMM alignment.

Note that the output of this system has to be used for computing the normalization UBM parameters in (6) and (7). In our experiments, the topology of this network is identical to the one used for BN feature extraction except for the number of output nodes (see Section 3). Note also that the DNN is usually trained on a separate set of speech features (log Mel filter bank outputs in our case), which is different from the features used in (3) for collecting the sufficient statistics (MFCC, etc.).

4.4. BN alignment (BNA)

In this approach, a GMM is trained on BN features. For the i-vector extraction, however, this GMM is only used to obtain the alignment of frames to UBM components (i.e. to calculate the posteriors $\gamma_t^{(c)}$). Just like in the DNN-based approach, different set of features is used to collect the sufficient statistics (3). Similarly to the DNN-based approach, a consistent BN-based alignment has to be also used to compute the normalization parameters (6) and (7). When BN features are used for both the alignment and the sufficient statistics collection, then there is no difference between BNA and the standard GMM-UBM approach.

BNA was first proposed in Tian et al. (2015) and afterward analyzed in Matejka et al. (2016). A GMM trained using BN features seems to partition the feature space into phone-like clusters and leads to an alignment similar to the DNN-based one. Again, this method does not take into account the true phrase transcription, which can be detrimental for rejecting Target-Wrong trials.

[1] We assume that (phonetic) transcriptions of the enrollment phrases are known, which is the case for our evaluation data.

5. Experimental setup

5.1. Data

For the sake of performing a comprehensive analysis, we did our experiments in two different scenarios. In the first one, all enrollment and test phrases are seen also in the data for system training. For this scenario, the RSR2015 dataset (Larcher et al., 2014) is used. In the second one, there is a mismatch in phrases between the training and evaluation data. Here, Part-01 of the RedDots dataset (Lee et al., 2015) is used for the evaluation. This dataset does not come with any training set. Therefore, we have to use a different dataset for the training and, as a result, (most of) the evaluation phrases do not appear in the training data.

The RSR2015 dataset comprises recordings from 300 speakers (157 males and 143 females), each in 9 distinct sessions. The data is divided into three disjoint speaker subsets: *background, development* and *evaluation*. It is further divided into three parts based on different lexical constraints. Since the focus of this paper is text-dependent speaker verification, we only use RSR2015 Part-1, where the enrollment and test phrases are the same. In this part, each speaker uttered 30 different TIMIT phrases in 9 sessions. For each phrase, three repetitions from different sessions were used to enroll a single i-vector as a speaker model and other phrases were used for testing based on RSR2015 trial definition.

In all experiments on RSR2015, the *background set* was used for UBM (both the GMM based and the monophone phoneme recognizer for the HMM-based alignment described in Section 4.2) and i-vector extractor training. All results are reported on the *evaluation sets*. The *development set* is not used at all. The training was done in a gender-independent manner. We used all speakers from the background set for gender-independent RWCCN and gender-dependent score normalization. Based on our experimental results, we decided to use phrase-dependent RWCCN and score normalization in all experiments. Note that we use exactly the same training and test sets as Kenny et al. (2014b).[2] Therefore, our results should be directly comparable with the best results reported in Table 6 in Kenny et al. (2014b). We also use the same HTK-based MFCC features as in Kenny et al. (2014b). However, we use our own voice activity detection (VAD) different from the one in Kenny et al. (2014b).

Note that, in some studies, authors prefer to report results on the more challenging *development set*. We have also found this set more difficult. However, the results and conclusions drawn from the experiments on the *development set* are very consistent with those reported here on the *evaluation set*, which we have chosen for the sake of comparison with Kenny et al. (2014b).

The current snapshot of RedDots dataset contains 62 speakers (49 males and 13 females). 41 speakers are the target ones (35 males and 6 females) and the others are considered as unseen imposters. RedDots consists of four parts. In this paper, we used only Part-01 with the official evaluation protocol. In this part, each speaker uttered 10 common phrases. RedDots was used for evaluation and both RSR2015 (Part-1 of all sets including development set) and LibriSpeech were used as training data. We only report results for male trials and omit the unreliable results on the very limited number of female trials. For RedDots system, we used gender-dependent UBM and i-vector extractor. No channel compensation or score normalization was used for the reasons explained in Section 2.2. UBM and i-vector extractor were trained on a subset of freely available LibriSpeech data (i.e. Train-Clean-100) (Panayotov et al., 2015) with 251 speakers and about 100 h of speech. In this dataset, each speaker reads several books and each recording was split to short segments ranging from one to several sentences. For each segment, there is a word-level transcription.

When training DNNs on 8 kHz speech, the Switchboard-1 training data (Phase-1 Release 2) is used as described in Section 3. From this dataset, about 255 h of speech were used for DNNs training. When training DNNs on 16 kHz speech, we use two parts of LibriSpeech called Train-Clean-100 and Train-Clean-360 with about 460 h of speech. About 416 h are used for DNN training and the rest is used for cross-validation.

A summary of the contents and specifications of RSR2015, LibriSpeech and RedDots data sets is shown in Table 2 (Zeinali et al., 2017).

[2] We thank the authors for sharing their enrollment and trial lists.

Table 2
Datasets, parts and numbers of speakers (Larcher et al., 2014; Panayotov et al., 2015; Lee et al., 2015).

Dataset	Subset	# Males	# Females
RSR2015	Background	50	47
	Development	50	47
	Evaluation	57	49
LibriSpeech	Train-Clean-100	126	125
	Train-Clean-360	482	439
RedDots	Part-01	49	13

5.2. Features

As the baseline speech features for our experiments, 60-dimensional MFCCs are extracted from 16 kHz signal using HTK (Young et al., 1997) with a standard configuration: 25 ms Hamming windowed frames with 15 ms overlap. Unlike in text-independent systems, non-speech frames cannot be simply dropped as VAD errors would harm the Viterbi alignment. Therefore, we used a silence HMM to model the non-speech regions at the beginning and the 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.[3] Finally, cepstral mean and variance normalization is applied to the trimmed utterances.

Besides the cepstral features, several versions of 80-dimensional DNN based bottleneck features (one 8 kHz and four 16 kHz BN features as described in Section 3) are used in our experiments. Note that 8 kHz features are extracted from data down-sampled to 8 kHz.

5.3. Systems

All reported results are obtained with i-vector based systems. Based on two evaluation datasets, we used two different system configurations. For RSR2015, the 400-dimensional i-vectors are length-normalized (Garcia-Romero and Espy-Wilson, 2011), and further normalized using phrase-dependent RWCCN as described in Section 2.2. Cosine distance is then used to obtain speaker verification scores, which are further normalized using phrase-dependent s-norm. For RedDots, we used 600-dimensional i-vectors extracted from a gender-dependent system. The scoring was done using cosine distance.

Results are reported for individual i-vector based systems, which differ in the input features (MFCC, BN or their combination), in the sampling rate of DNNs training data and in the method for aligning speech frames to the Gaussian components as described in Section 2. The four possible alignment models are: (1) GMM with 1024 components (i.e. the standard i-vector approach), (2) HMM with 3 states and 8 Gaussian components for each of 39 mono-phones (resulting in total of 936 Gaussian components), (3) DNN with 1011 or 920 outputs (corresponding to 1011 or 920 Gaussian components in the i-vector extraction model for 8 kHz and 16 kHz DNNs, respectively) and (4) BNA extracted from DNN with about 8000 outputs. The numbers of Gaussian components in GMM and HMM based systems and the number of DNN outputs (target senones) were selected so that the resulting i-vector extractors have roughly the same number of parameters (size of total variability matrix **T**).

6. Results

We only report results for Imposter-Correct and Target-Wrong trials because the error rate on Imposter-Wrong trials for all methods is close to zero. For each method, the results are reported in terms of Equal Error Rate (EER) and Normalized Detection Cost Function as defined for NIST SRE08 ($NDCF^{min}_{old}$) and NIST SRE10 ($NDCF^{min}_{new}$). In

[3] Only 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.

all DET curves, the square and star markers correspond to $NDCF^{min}_{old}$ and $NDCF^{min}_{new}$ operating points, respectively. In each section of tables, the best result is highlighted.

6.1. Comparison of GMM, HMM, DNN and BN alignments

In Tables 3 and 4, we analyze the performance of the four different alignment techniques for i-vector extraction (see Section 2) on RSR2015. The DET curves for a few systems selected from Table 3 are also shown in Figs. 1 and 2. In addition, Table 5 and Fig. 3 show these analyses for RedDots dataset. In these experiments, DNN alignments were calculated using a DNN with 920 senone targets and BN features were extracted by a bottleneck DNN with 9418 senone targets. All DNNs in these experiments were trained on 16 kHz speech.

We start our analyses on RSR2015 dataset and the most difficult Imposter-Correct condition (i.e. every non-target trial comes from an imposter speaker uttering the correct phrase). The first section of Table 3 shows results with MFCC features. The first line corresponds to the standard i-vector extraction model with GMM alignment as used in *text-independent* speaker verification. From the second line, we can see that the HMM-based alignment significantly improves the performance, which is in line with the results from Zeinali et al. (2017), where this method was proposed and analyzed. DNN based alignment performs better than HMM, even though it does not rely on the phrase transcription. Note that the nature of the DNN based alignment is rather different from (and perhaps complementary to) the HMM one: Instead of relying on the transcription, DNN makes the decision locally based only on the acoustic

Table 3
Comparison of different features and alignment methods on Imposter-Correct trials of the RSR2015 dataset. Note that all features are extracted from 16 kHz speech signal.

Features	Alignment	Male			Female		
		EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
MFCC	GMM	0.67	0.0382	0.1983	0.62	0.0355	0.1991
	HMM	0.37	0.0204	0.1142	0.49	0.0275	0.1533
	DNN	0.32	0.0174	0.0985	0.37	0.0203	0.1282
	BNA	0.32	0.0177	0.0697	0.27	0.0134	0.0730
BN	GMM (BNA)	0.42	0.0234	0.1319	0.27	0.0136	0.0837
	HMM	0.37	0.0206	0.1263	0.30	0.0136	0.0806
	DNN	0.69	0.0336	0.1792	0.54	0.0240	0.1311
MFCC+BN	GMM	0.22	0.0132	**0.0790**	0.18	0.0091	0.0477
	HMM	**0.20**	0.0128	0.0901	0.25	0.0111	0.0655
	DNN	0.41	0.0222	0.1404	0.42	0.0224	0.1211
	BNA	0.21	**0.0127**	0.0888	0.22	0.0097	0.0540
MFCC	Rel. MAP/GMM	0.40	0.0199	0.1061	**0.15**	**0.0081**	**0.0354**
MFCC+BN	Rel. MAP/GMM	0.31	0.0161	0.0998	0.17	0.0091	0.0405

Table 4
Comparison of different features and alignment methods on Target-Wrong trials of the RSR2015 dataset.

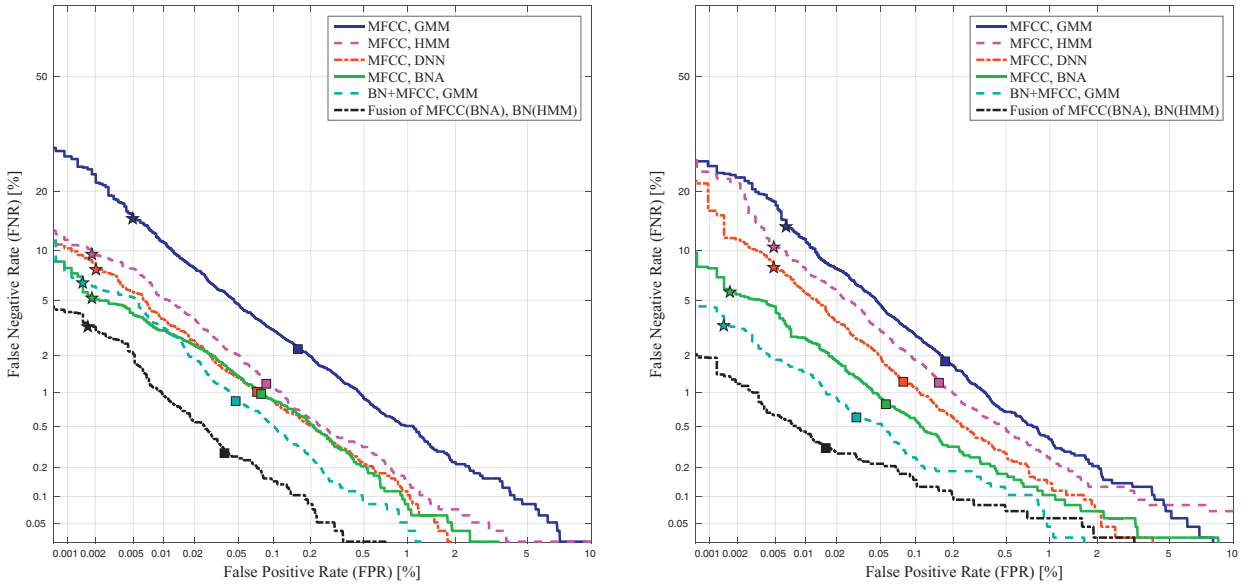
Features	Alignment	Male			Female		
		EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
MFCC	GMM	2.06	0.1301	0.6649	0.72	0.0468	0.3088
	HMM	0.32	0.0179	0.1024	0.14	0.0058	0.0415
	DNN	0.87	0.0584	0.4453	0.36	0.0221	0.1502
	BNA	0.17	0.0101	0.0428	0.08	0.0034	0.0194
BN	GMM (BNA)	0.04	0.0029	0.0143	0.05	0.0017	0.0158
	HMM	0.09	0.0054	0.0359	0.07	0.0026	0.0111
	DNN	0.22	0.0125	0.0574	0.16	0.0077	0.0260
MFCC+BN	GMM	0.04	0.0025	0.0157	**0.02**	0.0013	0.0135
	HMM	0.05	0.0028	0.0338	0.06	0.0016	0.0067
	DNN	0.31	0.0195	0.1289	0.22	0.0139	0.0946
	BNA	**0.02**	**0.0017**	**0.0074**	0.03	**0.0012**	**0.0027**
MFCC	Rel. MAP/GMM	0.62	0.0361	0.1788	0.13	0.0058	0.0330
MFCC+BN	Rel. MAP/GMM	0.29	0.0102	0.0322	0.09	0.0043	0.0227

Table 5
Comparison of different features and alignment methods on RedDots dataset for both Imposter-Correct and Target-Wrong trials.

Features	Alignment	Imposter-Correct			Target-Wrong		
		EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
MFCC	GMM	2.07	0.0899	0.3105	3.76	0.1762	0.4275
	HMM	1.88	**0.0809**	**0.2271**	1.11	0.0338	0.0509
	DNN	**1.64**	0.0820	0.3098	1.76	0.0806	0.1843
	BNA	2.31	0.0938	0.2750	2.50	0.0989	0.3179
BN	GMM (BNA)	5.15	0.2500	0.6790	0.37	0.0109	0.0164
	HMM	5.18	0.2388	0.6752	0.28	**0.0054**	0.0074
	DNN	4.81	0.2364	0.6635	**0.25**	0.0063	0.0111
MFCC+BN	GMM	3.46	0.1446	0.5368	0.56	0.0189	0.0673
	HMM	3.40	0.1354	0.4305	0.40	0.0059	**0.0065**
	DNN	2.99	0.1342	0.4298	0.43	0.0127	0.0281
	BNA	3.58	0.1659	0.5566	0.49	0.0165	0.0284
MFCC	Rel. MAP/GMM	1.98	0.0848	0.2879	4.01	0.1733	0.4960
MFCC+BN	Rel. MAP/GMM	2.59	0.1295	0.4423	0.46	0.0155	0.0549

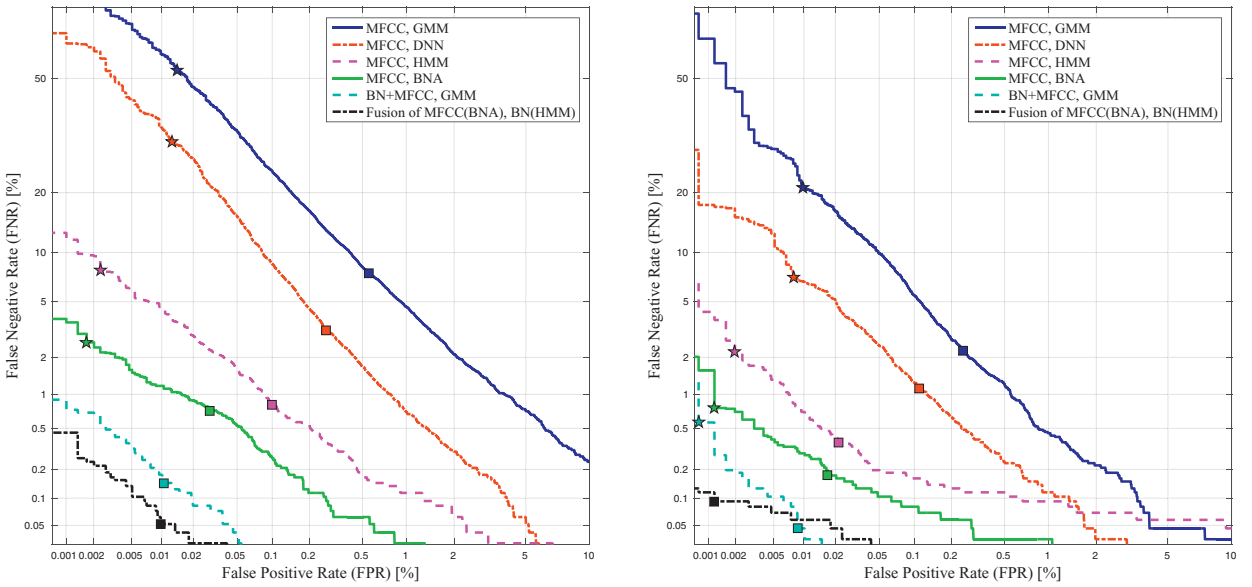
context; the alignment units are tied triphone states (senones) rather than Gaussian components in mono-phone states. Also, the DNN is discriminatively trained on a large amount of speech data and using different features, while HMMs are trained only on the small amount of RSR2015 *background set*. On the other hand, the HMM-based method leads to much more compact representation as there is just a single model (and features) used for both the alignment and the rest of the i-vector extraction. From the last row of this section, it is clear that the BN Alignment (BNA) performs comparably to the DNN one for males and is much better for females. Again, unlike with HMMs, BNA does not rely on the phrase transcription.

It is worth mentioning that for the HMM-based alignment, each phrase must be uttered correctly. If a phoneme is pronounced incorrectly, it affects the alignment accuracy of other phonemes during the Viterbi forced alignment. We know that there are various mispronunciations in both RSR2015 and RedDots as both were collected mostly



(a) Male (b) Female

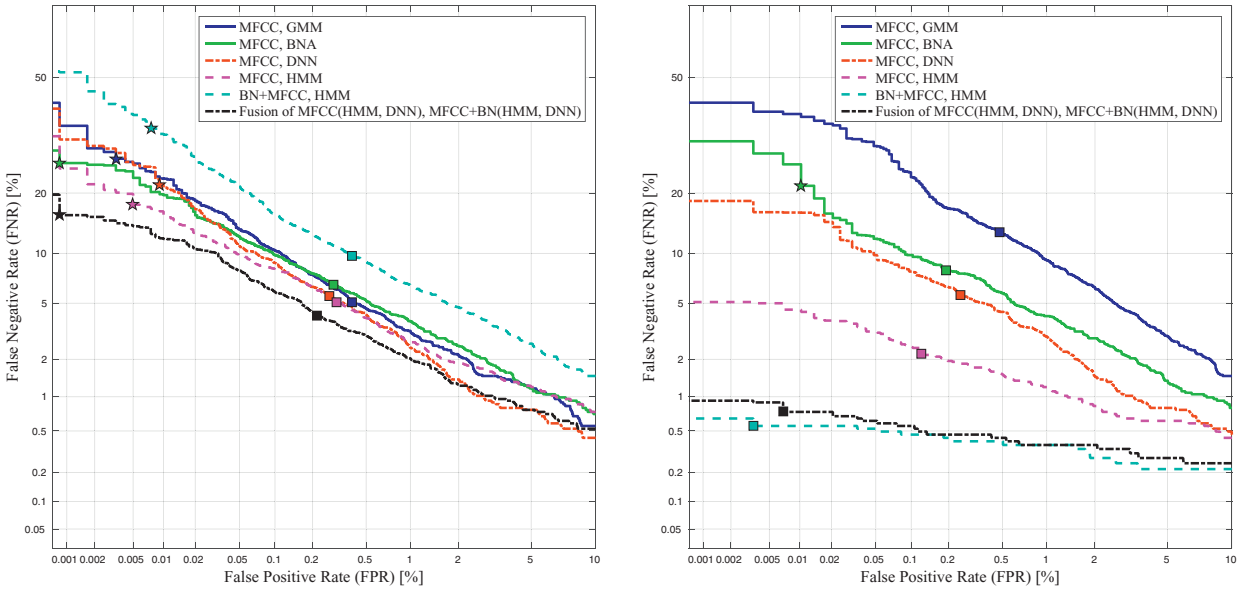
Fig. 1. DET curves for different methods of extracting posterior probabilities for Imposter-Correct trials of RSR2015 from Table 3. The fusion system was selected from Table 10.



(a) Male (b) Female

Fig. 2. DET curves for different methods of extracting posterior probabilities for Imposter-Correct trials of RSR2015 from Table 3. The fusion system was selected from Table 10.

from non-native English speakers. In addition, there are other human errors such as late starts or early terminations of recordings, which generate cropped utterances. These errors harm HMM-based alignment more than the other alignment methods, which we have observed when manually inspecting large portion of the errors made by the HMM-based system.



(a) Male (b) Female

Fig. 3. DET curves for different methods of extracting posterior probabilities for RedDots from Table 5. The fusion system was selected from Table 11.

The second section of Table 3 reports the results obtained with the BN features. We can see that BN features perform very well even in the standard i-vector setting with the GMM alignment (i.e. with the BN Alignment, which is implicit in the case of BN features). Most likely, this can be attributed to the better phone-like feature space clustering obtained with the GMM trained on BN features. For HMM-based systems, we see that replacing MFCCs with BN features helps especially on female trials. However, the improvement is not as significant as in the case of simple GMM-based alignment. Interestingly, BN features fail to perform well in combination with the DNN based alignment.

In the third section of Table 3, results are shown for concatenated MFCC+BN features ($60 + 80 = 140$ dimensions). In Matejka et al. (2016), superior performance was reported for i-vector based *text-independent* speaker recognition with such more informative, higher dimensional features regardless of the alignment method used. Here, we verify that concatenated MFCC+BN features provide excellent performance also for the *text-dependent* task on RSR2015 dataset. This time, however, only small improvement is obtained from the HMM-based alignment compared to the GMM-based one. It seems that the presence of the BN features already guarantees an appropriate feature space partitioning and alignment even with the GMM model. Again, the DNN based alignment seems to fail in the presence of the BN features. The performance of BNA (i.e. where the GMM for alignment is trained only on BN features rather than on MFCC+BN features) is also comparable to HMM- and GMM-based alignments.

For comparison, the last two rows in Table 3 show results for the simple relevance MAP GMM-UBM system (Reynolds et al., 2000; Larcher et al., 2013; 2012) trained using MFCC or MFCC+BN features. For females, this simple system with MFCC features actually provides the best results. Still, the best overall performance is obtained with MFCC+BN features and the i-vector based systems. It is perhaps even more important that the i-vector systems lead to more compact representation of utterances and therefore are more practical compared to the simple relevance MAP systems.

Now, we turn our attention to RSR2015 Target-Wrong condition, where all non-target trials come from target speakers uttering a wrong phrase. The first section of Table 4 presents again results with MFCC features. We can see that the HMM-based alignment performs much better compared to the GMM-based one. HMM makes use of enrollment phrase transcription, so for wrong trials, frames are aligned to wrong states and Gaussian components. Extracted i-vectors from such alignments are different from enrollment i-vectors and can be easily rejected. DNN alignment performs much better compared to GMM-based one. However, it is still significantly worse than HMM alignment as it does not take into account the enrollment phrase transcription. Perhaps somewhat surprisingly, the best performance with MFCC features for Target-Wrong condition is obtained with BN Alignment. We found that BN features, even if used only for alignment, produce very phrase-specific i-vectors, which are very good for the discrimination between phrases and therefore for rejecting wrong phrase trials.

In the second section of Table 4, we can see that even better performance in rejecting wrong phrase trials is obtained when BN features are used for collecting sufficient statistics (i.e. not only for the alignment). In this case, the GMM alignment performs the best. By inspecting the verification results, we observe that the worse performance of HMM alignment is caused mainly by the aforementioned problems with phrase mispronunciations and utterance cropping. Again, the combination of BN features and DNN alignment leads to the worst performance.

The best results were again obtained with MFCC+BN features, as can be seen in the third section of Table 4. Any of the alignment methods (except for DNN) provides excellent results. In almost all cases, BN Alignment performs the best.

For comparison, the last two lines of the table show again the results for the simple relevance MAP GMM-UBM systems. Although reasonable results can be obtained with such systems, their performance still stays far behind the systems from the previous section (i.e. i-vector and MFCC+BN based).

In Table 5, results for the male part of RedDots dataset are reported for both Imposter-Correct and Target-Wrong conditions. First, we focus on the Imposter-Correct results presented in the left side of the table. In contrast with the results obtained on RSR2015 data, any use of BN features always leads to suboptimal performance on RedDots. As can be seen in the left part of the first section of Table 5, the best results are obtained with MFCC features and HMM alignment. A comparable performance is also obtained with DNN alignment.

In order to explain the suboptimal performance obtained with the BN features on RedDots dataset, we need to take into account the mismatch between the RedDots evaluation data and the training data used. In our RSR2015 experiments, the data for training HMM-UBM and i-vector extractor contains the same phrases as the enrollment

and test utterances. However, this is not the case for most of the phrases in RedDots. Therefore, we suspected that the BN features might be sensitive to such mismatch between training and test data. To prove this hypothesis, we concentrated on three out of the ten phrases from Part-01 of RedDots that are common with RSR2015 data. Since the RSR2015 data are also used for RedDots system training, the i-vector extractor is also effectively trained on the data from these three matched phrases. In Fig. 4, we plot score distributions estimated on correct phrase trials for two separate sets of phrases: *Seen* set contains the three phrases seen during training, and *Unseen* set contains the remaining seven phrases. For MFCC+BN features, the target and non-target score distributions of the *Seen* set are farther from each other than for the *Unseen* set. Therefore, better performance of the BN features is to be expected when test phrases are included in the training data.

The RedDots results for the Target-Wrong conditions can be found in the right part of Table 5. For MFCC features, the best performance is obtained with HMM alignment. However, just like in the case of the RSR2015 Target-Wrong condition, a superior performance is obtained with the BN and MFCC+BN features (with any alignment method), which performed poorly for Imposter-Correct condition. As mentioned before very phrase specific i-vectors are obtained with BN features, which makes it easy to reject wrong phrase trials.

The last two lines of Table 5 show again results for the simple relevance MAP GMM-UBM systems. For the Imposter-Correct condition, the performance of the MFCC based system is comparable to the best i-vector system from the first section of the table. However, as already pointed out, the i-vector based systems are more practical because of their compact representation of utterances.

In summary, the best features for RSR2015 dataset are the concatenated MFCC+BN and among four alignment methods, BNA performs the best. Although the performance of the HMM-based method is slightly worse, we believe it can be improved by properly dealing with the mispronounced utterances. For RedDots dataset, MFCC features with HMM or DNN alignments perform well on Imposter-Correct condition, while BN features have the best performance on Target-Wrong condition. In order to take advantage for both conditions, we fuse systems that makes use of these different features (see Section 6.4).

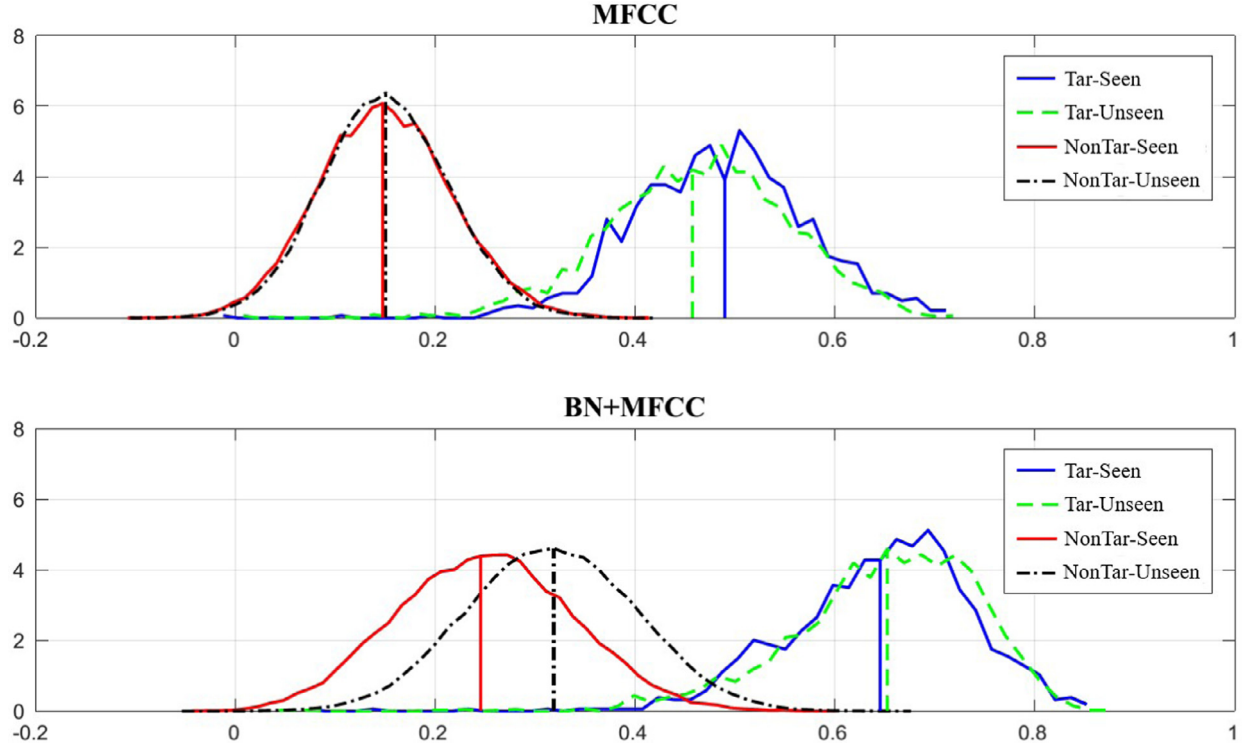


Fig. 4. Score distributions of two different phrase sets for correct male trials for MFCC and MFCC+BN. The vertical lines show the means of normal distributions fitted to scores (Zeinali et al., 2016b).

6.2. Comparison of 8 kHz and 16 kHz networks

In Zeinali et al. (2016a), we reported results with DNNs trained on Switchboard-1 database (8 kHz telephone conversational speech) as those were the only DNNs we had available at the time of writing the Odyssey paper. In this section, we compare the results obtained with DNNs trained on either 8 kHz or 16 kHz speech. The 16 kHz DNNs are trained on LibriSpeech database (high quality read speech). For a fair comparison, the 8 kHz DNNs are trained on the same dataset downsampled to 8 kHz. We also include results obtained with 8 kHz DNNs trained on the Switchboard-1 database. The results for the three datasets allow us to assess whether it is more important to have DNNs trained on the wide-band 16 kHz or DNNs trained more robustly on a telephone data from the real environments.

For comparison, the first lines in Tables 6–8 show results obtained with 8 kHz MFCC features and the conventional GMM based i-vector system. Note, however, that 16 kHz MFCCs are used in all other cases even if the 8 kHz DNN is used for the alignment or to extract the BN features.

Table 6 shows results on RSR2015 dataset for Imposter-Correct condition. Comparing the DNNs trained on the different 8 kHz datasets (Switchboard vs. downsampled LibriSpeech), we see consistently better performance with LibriSpeech, which matches well the conditions of RSR2015 data (i.e. contains very clean, non-spontaneous speech). We observe further consistent improvement (especially for BN features and female trials) when training DNNs on 16 kHz LibriSpeech.

Table 7 shows similar results for RSR2015 Target-Wrong condition. We again see that it is beneficial to train DNNs on the clean LibriSpeech data. However, the improvement from using the 16 kHz speech compared to the downsampled 8 kHz speech is rather small. For Target-Wrong condition, we need to discriminate between phrases (not speaker voices) and it seems that 8 kHz data is sufficient for this task.

On the other hand, for the RedDots experiments reported in Table 8, we do not see any benefit from training DNNs on the clean 16 kHz LibriSpeech. On the contrary, the 8 kHz DNNs trained on Switchboard often deliver better performance. The most likely reason for this behavior is the nature of the training and evaluation datasets: RSR2015 was recorded in controlled environment and has a limited noise, while RedDots was collected in real environments and therefore it is more noisy and challenging. Switchboard database contains telephone conversational speech with different channels and noises. Therefore, DNNs trained on 8 kHz Switchboard data are more robust.

In all other experiments reported in this paper, DNNs trained on 16 kHz speech are used as they provide superior performance on RSR2015 data and comparable performance on RedDots data. As mentioned before, in Zeinali et al. (2016a), we reported results with DNNs trained only on Switchboard as those were the only DNNs we had available at the time of writing the paper. We recommend the reader to compare the results from Section 6.1 with similar results reported in Zeinali et al. (2016a), which were obtained with the 8 kHz DNN systems. The new conclusions drawn here from the 16 kHz DNNs results are sometimes quite different from the ones reported in Zeinali et al. (2016a).

Table 6
Comparison of 8 and 16 kHz DNNs on Imposter-Correct trials of the RSR2015 dataset. The second column shows the main features used for verification and the third one shows the alignment method. MFCC features are always extracted from 16 kHz speech except for the first line of the table, where 8 kHz speech is used.

DNN training	Features	Align.	Male			Female		
			EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
–	MFCC-8 kHz	GMM	1.46	0.0822	0.3485	2.10	0.0952	0.3410
	MFCC-16 kHz	GMM	0.67	0.0382	0.1983	0.62	0.0355	0.1991
8 kHz Switch board-1	MFCC	DNN	0.36	0.0203	0.1286	0.39	0.0218	0.1441
	MFCC	BNA	0.39	0.0192	0.0855	0.30	0.0180	0.0937
	BN	GMM	0.59	0.0325	0.1564	0.40	0.0201	0.1066
	MFCC+BN	GMM	0.31	0.0176	0.0955	0.28	0.0144	0.0898
8 kHz Libri Speech	MFCC	DNN	0.31	0.0190	0.1086	0.35	0.0203	0.1595
	MFCC	BNA	0.31	0.0178	0.0857	0.33	0.0156	0.0852
	BN	GMM	0.53	0.0283	0.1530	0.31	0.0161	0.0966
	MFCC+BN	GMM	0.23	0.0149	0.0910	0.25	0.0123	0.0721
16 kHz Libri Speech	MFCC	DNN	0.32	0.0174	0.0985	0.37	0.0203	0.1282
	MFCC	BNA	0.32	0.0177	**0.0697**	0.27	0.0134	0.0730
	BN	GMM	0.42	0.0234	0.1319	0.27	0.0136	0.0837
	MFCC+BN	GMM	**0.22**	**0.0132**	0.0790	**0.18**	**0.0091**	**0.0477**

Table 7
Comparison of 8 and 16 kHz DNNs on Target-Wrong trials of the RSR2015 dataset. The second column shows the main features used for verification and the third one shows the alignment method. MFCC features are always extracted from 16 kHz speech except for the first line of the table, where 8 kHz speech is used.

DNN training	Features	Align.	Male			Female		
			EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
–	MFCC-8 kHz	GMM	2.94	0.1861	0.8111	1.81	0.0965	0.4832
	MFCC-16 kHz	GMM	2.06	0.1301	0.6649	0.72	0.0468	0.3088
8 kHz Switch board-1	MFCC	DNN	1.13	0.0806	0.5709	0.42	0.0284	0.2133
	MFCC	BNA	0.21	0.0106	0.0503	0.08	0.0044	0.0249
	BN	GMM	0.12	0.0073	0.0357	0.09	0.0046	0.0278
	MFCC+BN	GMM	0.08	0.0054	0.0330	0.07	0.0025	0.0236
8 kHz Libri Speech	MFCC	DNN	0.95	0.0620	0.4495	0.39	0.0249	0.1805
	MFCC	BNA	0.16	0.0095	0.0463	0.12	0.0039	0.0195
	BN	GMM	0.09	0.0060	0.0302	0.07	0.0026	0.0110
	MFCC+BN	GMM	0.05	0.0026	0.0155	0.03	0.0012	0.0030
16 kHz Libri Speech	MFCC	DNN	0.87	0.0584	0.4453	0.36	0.0221	0.1502
	MFCC	BNA	0.17	0.0101	0.0428	0.08	0.0034	0.0194
	BN	GMM	**0.04**	0.0029	**0.0143**	0.05	0.0017	0.0158
	MFCC+BN	GMM	**0.04**	**0.0025**	0.0157	**0.02**	**0.0013**	**0.0135**

Table 8
Comparison of 8 and 16 kHz DNNs on the male part of the RedDots dataset. The second column shows the main features used for verification and the third one shows the alignment method. MFCC features are always extracted from 16 kHz speech except for the first line of the table, where 8 kHz speech is used.

DNN training	Features	Align.	Imposter-Correct			Target-Wrong		
			EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
–	MFCC-8 kHz	GMM	2.75	0.1317	0.4326	3.77	0.1832	0.3917
	MFCC-16 kHz	GMM	2.07	0.0899	0.3105	3.76	0.1762	0.4275
8 kHz Switch board-1	MFCC	DNN	1.67	**0.0765**	0.2786	1.54	0.0693	0.1978
	MFCC	BNA	2.41	0.0993	**0.2672**	2.35	0.0870	0.1775
	BN	GMM	4.91	0.2255	0.6681	0.50	0.0173	0.0562
	MFCC+BN	GMM	3.05	0.1385	0.5002	0.56	0.0226	0.0515
8 kHz Libri Speech	MFCC	DNN	1.73	0.0817	0.3287	1.76	0.0760	0.2099
	MFCC	BNA	2.47	0.0925	0.2790	2.47	0.0843	0.2398
	BN	GMM	5.59	0.2552	0.7215	0.46	0.0111	0.0157
	MFCC+BN	GMM	3.33	0.1565	0.5060	0.59	0.0232	0.0525
16 kHz Libri Speech	MFCC	DNN	**1.64**	0.0820	0.3098	1.76	0.0806	0.1843
	MFCC	BNA	2.31	0.0938	0.2750	2.50	0.0989	0.3179
	BN	GMM	5.15	0.2500	0.6790	**0.37**	**0.0109**	**0.0164**
	MFCC+BN	GMM	3.46	0.1446	0.5368	0.56	0.0189	0.0673

6.3. Influence of the number of senones

Table 9 reports the results of different BN features extracted from different DNNs with varying numbers of senones. In this part, we just report results for the male part of RSR2015 and RedDots using MFCC+BN features and HMM-based method. From this table, it is clear that all DNNs perform similarly, especially for RSR2015. It seems that the larger networks performs slightly better, but not significantly. These results are consistent with Tian et al. (2015), where performances of different DNNs differed only slightly in text-independent speaker verification.

6.4. Fusion results

Table 10 shows the results for different strategies of combining features and systems for RSR2015 dataset. The DET curves for the best systems are shown in Figs. 1 and 2. Note that Table 10 repeats some of the results from

Table 9
Comparison of performance based on number of senones for the male part of both RSR2015 and RedDots datasets. MFCC+BN features and HMM-based alignment were used for this experiment.

Senones	Non-target trial type	RSR2015			RedDots		
		EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
920	Imposter-Correct	0.26	0.0151	0.1021	**2.96**	**0.1277**	**0.3882**
	Target-Wrong	0.08	0.0053	0.0227	0.40	0.0066	0.0114
3512	Imposter-Correct	0.23	0.0133	0.1091	3.58	0.1397	0.4024
	Target-Wrong	0.06	0.0030	**0.0159**	0.37	0.0066	0.0083
6198	Imposter-Correct	0.21	0.0138	**0.0896**	3.21	0.1389	0.4455
	Target-Wrong	**0.04**	0.0033	0.0185	**0.34**	**0.0059**	0.0142
9418	Imposter-Correct	**0.20**	**0.0128**	0.0901	3.40	0.1354	0.4305
	Target-Wrong	0.05	**0.0028**	0.0338	0.40	**0.0059**	**0.0065**

Table 10
RSR2015 results for feature and score domain fusion using different strategies. The methods in parenthesis show score domain fusion of different alignment method using the same feature.

NT-TT	Feature(Alignments)	Male			Female		
		EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
Imp-Corr	MFCC+BN(HMM)	0.20	0.0128	0.0901	0.25	0.0111	0.0655
	MFCC(HMM), BN(HMM)	**0.13**	0.0070	**0.0482**	0.15	0.0050	0.0274
	MFCC(BNA), BN(HMM)	**0.13**	**0.0065**	0.0497	**0.13**	**0.0047**	**0.0253**
	MFCC(BNA, HMM)	0.23	0.0132	0.0658	0.22	0.0104	0.0729
	MFCC+BN(GMM, HMM, BNA)	0.16	0.0095	0.0665	0.17	0.0077	0.0374
Tar-Wrg	MFCC+BN(HMM)	0.05	0.0028	0.0338	0.06	0.0016	0.0067
	MFCC(HMM), BN(HMM)	0.05	0.0022	0.0188	0.06	0.0013	0.0021
	MFCC(BNA), BN(HMM)	0.03	0.0015	0.0073	0.03	0.0010	0.0013
	MFCC(BNA, HMM)	0.11	0.0043	0.0143	0.09	0.0018	0.0037
	MFCC+BN(GMM, HMM, BNA)	**0.01**	**0.0009**	**0.0030**	**0.02**	**0.0005**	**0.0008**

Tables 3 and 4 to facilitate the comparison. Table 11 reports fusion results for RedDots dataset. Note that for each dataset, fusion strategies were selected based on the results on that dataset.

The first section of Table 10 contains Imposter-Correct condition results for score level fusion of systems with individual MFCC, BN or MFCC+BN features and different alignment methods as well as feature-level fusion of MFCC and BN features (i.e. MFCC+BN). In this work, we use the most trivial fusion: the scores from the individual systems are simply averaged (with equal weight). Interestingly, the score level fusion is very effective and, contrary to our experience from the *text-independent* task, it brings larger improvements than the concatenation of cepstral and BN features (i.e. MFCC+BN). In Zeinali et al. (2016a), we concluded that the small amount of RSR2015 training data might not be sufficient to train the larger model based on the higher-dimensional concatenated features. However, the score level fusion turns out to be more effective also for RedDots (see later), where we use much larger

Table 11
RedDots results for feature and score domain fusion using different strategies. The methods in parenthesis show score domain fusion of different alignment method using the same feature.

Feature(Alignments)	Imposter-Correct			Target-Wrong		
	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$	EER [%]	$NDCF^{min}_{old}$	$NDCF^{min}_{new}$
MFCC+BN(HMM)	3.40	0.1354	0.4305	0.40	**0.0059**	**0.0065**
MFCC(HMM), BN(HMM)	2.22	0.0988	0.3080	**0.31**	0.0065	0.0074
MFCC(HMM, DNN)	**1.33**	**0.0595**	0.1966	0.86	0.0340	0.0722
MFCC(HMM), MFCC+BN(HMM)	2.04	0.0775	0.2450	0.43	0.0081	0.0093
MFCC(DNN), BN(GMM)	2.11	0.0937	0.2746	0.49	0.0118	0.0127
MFCC(DNN), MFCC+BN(DNN)	1.69	0.0784	0.2697	0.60	0.0202	0.0309
MFCC+BN(HMM, DNN)	2.74	0.1172	0.3567	**0.31**	0.0066	0.0068
MFCC(HMM, DNN), MFCC+BN(HMM, DNN)	1.56	0.0625	**0.1663**	0.43	0.0078	0.0083

amount of training data. Clearly, the best results reported for RSR2015 Imposter-Correct condition are obtained with a two-fold score level fusion of MFCC and BN based systems with two different alignment methods (i.e. BNA and HMM).

The second section of Table 10 reports results of the same fused systems on Target-Wrong trials of RSR2015 dataset. Most of the conclusions drawn above are true here except that the best performance is for a three-fold fusion of different alignment methods using MFCC+BN features.

Table 11 shows the results of fusion strategies on RedDots dataset. As mentioned before, all the DNN related methods (i.e. DNN alignment and BN features) behave differently on RedDots. So, we selected HMM and DNN as the best alignment methods for fusion. The fusion of systems based on MFCC features and HMM and DNN alignments performs the best for Imposter-Correct trials. Again, the score-domain fusion of MFCC and BN features performs better than feature-domain fusion (compare the third and the first row). For Target-Wrong trials, all systems that used BN features and HMM alignment perform very well.

7. Conclusions

This work verified that the successful DNN-based approaches to *text-independent* speaker recognition are very effective for Imposter-Correct trials of the *text-dependent* task as well. Our baseline system is based on the previously proposed phrase-independent i-vector approach, where HMM-based phone recognizer serves as UBM for collecting sufficient statistics (Zeinali et al., 2017). In the case of the baseline system, the statistics are to be collected using forced-alignment based on the correct phrase transcription in order to obtain good performance for the *text-dependent* task. On the other hand, similar or better verification performance is obtained with a DNN based alignment, where no transcription is necessary. For Target-Wrong trials, HMM outperforms the DNN approach, due to using the exact phrase transcription.

Furthermore, excellent performance was obtained with DNN based bottleneck features, especially when concatenated with the standard cepstral features for RSR2015 dataset. Our experiments support the hypothesis that a GMM trained on bottleneck features results in a superior partitioning of the feature space into the phone-like clusters: the standard i-vector based GMM-UBM provides performance similar to the phone transcription supervised HMM-based method. Unfortunately, these features perform well just for RSR2015 dataset and their performance on RedDots dataset is not so good for Imposter-Correct trials, while they perform very well for Target-Wrong trials of this dataset. It seems that in the close phrase-set task (i.e. when test phrases are present in the training data) BN features work better than in the open phrase-set task. Based on our experiments on RedDots dataset, in the presence of BN features, the part of i-vector system that most affected by the open-set training is the i-vector extractor, while the training data does not have much influence on the UBM. BNA is another alignment method that was investigated in this paper. This method also performs very well on RSR2015 dataset, however, for RedDots its performance is worse than other methods.

Our experimental results show that the 8 kHz DNNs work similar to 16 kHz DNNs on text-dependent speaker verification. The 16 kHz DNNs perform better on RSR2015 while their performance is a bit worse for RedDots dataset. The performance gap between the 8 kHz BN and MFCC features is much higher than for the 16 kHz version. As a future work, we plan more experiments on 16 kHz DNNs for improving their performance or finding a reason for this behavior.

Based on the experimental results, it seems that we need more efforts on the BN-based method, especially for solving its weakness in the open phrase-set scenario. Although the DNN-based method in some cases outperformed the HMM-based one, we believe that the HMM method reflects the very nature of the text-dependent task and we should be able to improve its performance. Experimenting with triphone models to improve the context modeling will be the first natural step of our future work.

Acknowledgment

The work was partially supported by Iranian Ministry of Science, by Czech Ministry of Interior Project no. VI20152020025 “DRAPAK”, European Union’s Horizon 2020 Project no. 645523 BISON and by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project “IT4Innovations excellence in science – LQ1602”.

References

Aronowitz, H., 2012. Text dependent speaker verification using a small development set. In: Proceedings of Odyssey – The Speaker and Language Recognition Workshop, pp. 312–316.

Dahl, G.E., Yu, D., Deng, L., Acero, A., 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. Audio Speech Lang. Process. 20 (1), 30–42.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011. Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. 19 (4), 788–798.

Friedman, J.H., 1989. Regularized discriminant analysis. J. Am. Stat. Assoc. 84 (405), 165–175.

Garcia-Romero, D., Espy-Wilson, C.Y., 2011. Analysis of i-vector length normalization in speaker recognition systems. In: Proceedings of Interspeech, pp. 249–252.

Garcia-Romero, D., McCree, A., 2015. Insights into deep neural networks for speaker recognition. In: Proceedings of Interspeech, pp. 1141–1145.

Garcia-Romero, D., Zhang, X., McCree, A., Povey, D., 2014. Improving speaker recognition performance in the domain adaptation challenge using deep neural networks. In: Proceedings of Spoken Language Technology Workshop (SLT). IEEE, pp. 378–383.

Ghahabi, O., Hernando, J., 2014. Deep belief networks for i-vector based speaker recognition. In: Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 1700–1704.

Grezl, F., Karafiát, M., Burget, L., 2009. Investigation into bottle-neck features for meeting speech recognition. In: Proceedings of Interspeech, pp. 2947–2950.

Hatch, A.O., Kajarekar, S.S., Stolcke, A., 2006. Within-class covariance normalization for SVM-based speaker recognition. In: Proceedings of Interspeech. (paper 1874)

Heigold, G., Moreno, I., Bengio, S., Shazeer, N., 2016. End-to-end text-dependent speaker verification. In: Proceedings of 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5115–5119.

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.-R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al., 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process. Mag. 29 (6), 82–97.

Karafiát, M., Grézl, F., Veselý, K., Hannemann, M., Szőke, I., Černocký, J., 2014. BUT 2014 Babel system: analysis of adaptation in NN based systems. In: Proceedings of Interspeech, pp. 3002–3006.

Kenny, P., 2010. Bayesian speaker verification with heavy-tailed priors. In: Proceedings of Odyssey. (paper 14)

Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P., 2007. Joint factor analysis versus eigenchannels in speaker recognition. IEEE Trans. Audio Speech Lang. Process. 15 (4), 1435–1447.

Kenny, P., Gupta, V., Stafylakis, T., Ouellet, P., Alam, J., 2014a. Deep neural networks for extracting Baum–Welch statistics for speaker recognition. In: Proceedings of Odyssey – The Speaker and Language Recognition Workshop, pp. 293–298.

Kenny, P., Ouellet, P., Dehak, N., Gupta, V., Dumouchel, P., 2008. A study of interspeaker variability in speaker verification. IEEE Trans. Audio Speech Lang. Process. 16 (5), 980–988.

Kenny, P., Stafylakis, T., Alam, J., Ouellet, P., Kockmann, M., 2014b. Joint factor analysis for text-dependent speaker verification. In: Proceedings of Odyssey – The Speaker and Language Recognition Workshop, pp. 200–207.

Kenny, P., Stafylakis, T., Ouellet, P., Alam, M.J., 2014c. JFA-based front ends for speaker recognition. In: Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 1705–1709.

Larcher, A., Lee, K.A., Ma, B., et al., 2013. Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances. In: Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 7673–7677.

Larcher, A., Lee, K.A., Ma, B., Li, H., 2012. RSR2015: database for text-dependent speaker verification using multiple pass-phrases. In: Proceedings of Interspeech, pp. 1580–1583.

Larcher, A., Lee, K.A., Ma, B., Li, H., 2014. Text-dependent speaker verification: classifiers, databases and RSR2015. Speech Commun. 60, 56–77.

Lee, K.A., Larcher, A., Wang, G., Kenny, P., Brümmer, N., van Leeuwen, D., Aronowitz, H., Kockmann, M., Vaquero, C., Ma, B., et al., 2015. The RedDots data collection for speaker recognition. In: Proceedings of Interspeech, pp. 2996–3000.

Lei, Y., Scheffer, N., Ferrer, L., McLaren, M., 2014. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In: Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 1695–1699.

Liu, Y., Qian, Y., Chen, N., Fu, T., Zhang, Y., Yu, K., 2015. Deep feature for text-dependent speaker verification. Speech Commun. 73, 1–13.

Lozano-Diez, A., Silnova, A., Matejka, P., Glembek, O., Plchot, O., Pešán, J., Burget, L., Gonzalez-Rodriguez, J., 2016. Analysis and optimization of bottleneck features for speaker recognition. In: Proceedings of Odyssey 2016, pp. 21–24.

Matejka, P., Glembek, O., Novotny, O., Plchot, O., Grezl, F., Burget, L., Cernocky, J., 2016. Analysis of DNN approaches to speaker identification. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5100–5104.

Matejka, P., Zhang, L., Ng, T., Mallidi, H., Glembek, O., Ma, J., Zhang, B., 2014. Neural network bottleneck features for language identification. In: Proceedings of Odyssey – The Speaker and Language Recognition Workshop, pp. 299–304.

Novoselov, S., Pekhovsky, T., Shulipa, A., Sholokhov, A., 2014. Text-dependent GMM-JFA system for password based speaker verification. In: Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 729–737.

Panayotov, V., Chen, G., Povey, D., Khudanpur, S., 2015. Librispeech: an ASR corpus based on public domain audio books. In: Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 5206–5210.

Prince, S.J., Elder, J.H., 2007. Probabilistic linear discriminant analysis for inferences about identity. In: Proceedings of IEEE 11th International Conference on Computer Vision (ICCV 2007). IEEE, pp. 1–8.

Reynolds, D.A., Quatieri, T.F., Dunn, R.B., 2000. Speaker verification using adapted Gaussian mixture models. Digital Signal Process. 10 (1), 19–41.

Richardson, F., Reynolds, D.A., Dehak, N., 2015. A unified deep neural network for speaker and language recognition. In: Proceedings of Interspeech, pp. 1146–1150.

Stafylakis, T., Kenny, P., Ouellet, P., Perez, J., Kockmann, M., Dumouchel, P., 2013. Text-dependent speaker recognition using PLDA with uncertainty propagation. In: Proceedings of Interspeech, pp. 3684–3688.

Tian, Y., Cai, M., He, L., Liu, J., 2015. Investigation of bottleneck features and multilingual deep neural networks for speaker verification. In: Proceedings of Interspeech, pp. 1151–1155.

Variani, E., Lei, X., McDermott, E., Lopez Moreno, I., Gonzalez-Dominguez, J., 2014. Deep neural networks for small footprint text-dependent speaker verification. In: Proceedings of 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 4052–4056.

Vesely, K., Karafiát, M., Grezl, F., Janda, M., Egorova, E., 2012. The language-independent bottleneck features. In: Proceedings of 2012 IEEE Spoken Language Technology Workshop (SLT). IEEE, pp. 336–341.

Yaman, S., Pelecanos, J., Sarikaya, R., 2012. Bottleneck features for speaker recognition. In: Proceedings of Odyssey – The Speaker and Language Recognition Workshop, 12, pp. 105–108.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., et al., 1997. The HTK Book. 2. Entropic Cambridge Research Laboratory, Cambridge.

Zeinali, H., Burget, L., Sameti, H., Glembek, O., Plchot, O., 2016a. Deep neural networks and hidden Markov models in i-vector-based text-dependent speaker verification. In: Proceedings of Odyssey – The Speaker and Language Recognition Workshop, pp. 24–30.

Zeinali, H., Kalantari, E., Sameti, H., Hadian, H., 2015. Telephony text-prompted speaker verification using i-vector representation. In: Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 4839–4843.

Zeinali, H., Sameti, H., Burget, L., 2017. HMM-based phrase-independent i-vector extractor for text-dependent speaker verification. IEEE/ACM Trans. Audio Speech Lang. Process. PP (99). 1–1

Zeinali, H., Sameti, H., Burget, L., Cernocky, J., Maghsoodi, N., Matejka, P., 2016b. i-vector/HMM based text-dependent speaker verification system for RedDots challenge. In: Proceedings of Interspeech 2016, pp. 440–444.