Odyssey 2016
21-24 Jun 2016, Bilbao, Spain

Deep Neural Networks and Hidden Markov Models in i-vector-based Text-Dependent Speaker Verification

Hossein Zeinali[1,2], Lukáš Burget[2], Hossein Sameti[1], Ondřej Glembek[2], Oldřich Plchot[2]

[1] Sharif University of Technology, Tehran, Iran, zeinali@ce.sharif.edu, sameti@sharif.edu
[2] Brno University of Technology, Czech Republic, {burget,glembek,iplchot}@fit.vutbr.cz

Abstract

Techniques making use of Deep Neural Networks (DNN) have recently been seen to bring large improvements in text-independent speaker recognition. In this paper, we verify that the DNN based methods result in excellent performances in the context of text-dependent speaker verification as well. We build our system on the previously introduced HMM based i-vector approach, where phone models are used to obtain frame level alignment in order to collect sufficient statistics for i-vector extraction. For comparison, we experiment with an alternative alignment obtained directly from the output of DNN trained for phone classification. We also experiment with DNN based bottleneck features and their combinations with standard cepstral features. Although the i-vector approach is generally considered not suitable for text-dependent speaker verification, we show that our HMM based approach combined with bottleneck features provides truly state-of-the-art performance on RSR2015 data.

1. Introduction

During the last decade, *text-independent* speaker recognition technology has been largely improved in terms of both computational complexity and accuracy. The newly introduced channel-compensation techniques, such as Joint Factor Analysis (JFA) [1,2], evolved in the i-vector paradigm [3], where each speech utterance is represented by a low-dimensional fixed-length vector. To verify 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) [4,5].

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 correctly accepted. 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) [6, 7] or NAP compensated GMM mean supervector scored using SVM classifier [8, 9]. Only limited success was observed in the experiments with i-vectors/PLDA [10, 11] or JFA, which mainly served as an i-vector-like feature extraction method [12, 13].

In [14], Hidden Markov Model (HMM) based i-vector approach was proposed for *text-prompted* speaker verification, where the phrases are composed from 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 the phrase specific HMM. This HMM is in turn used to collect sufficient statistics for i-vector extraction instead of the conventional GMM-UBM. This approach was further extended to *text-dependent* task in [15], where the HMM models were trained for individual phonemes rather than words. 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 the 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 RSR2015 data [10]. However, the drawback of this approach 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 significantly 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 [16–21]. 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 [15] relying on the HMM alignment. Note that, unlike in the HMM based method, we do not need the phrase phoneme 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 [22–26]—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 [26, 27]. Such features are then used as the input to the 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 [16], where the feature space partitioning is given directly by the DNN outputs. This hypothesis is in agreement with the analysis in [26], where the best performance was obtained with

doi: 10.21437/Odyssey.2016-4

standard i-vector system, where the input features were BN features concatenated with standard MFCCs. While the BN features guaranteed good feature space partitioning, MFCCs contributed with the speaker information that may have been already suppressed in the BN features.

In this paper, we verify that BN features—combined with MFCC features—provide 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.

For completeness (although not studied in this work), let us mention that DNNs have been also used to extract speakers identity vector in a more direct way (compared to the DNN based i-vectors) [28–30] or to classify i-vectors in speaker recognition task [31].

2. i-vector Based System

2.1. General i-vector Extraction

Although thoroughly described in literature, let us review the basics of i-vector extraction. The main principle is that the utterance-dependent Gaussian Mixture Model (GMM) supervector 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 supervector (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 $\boldsymbol{\phi}$ 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 $\mathcal{X}$ 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*: $\mathbf{n}_\mathcal{X} = [N_\mathcal{X}^{(1)}, \ldots, N_\mathcal{X}^{(C)}]'$ and $\mathbf{f}_\mathcal{X} = [\mathbf{f}_\mathcal{X}^{(1)'}, \ldots, \mathbf{f}_\mathcal{X}^{(C)'}]'$, where

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

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

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 using a separate model [16,26,32]. In this work, we compare the standard GMM-UBM frame alignment with HMM- and DNN-based approaches, described in the following sections. The i-vector is then expressed as

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

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

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

with c being 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 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 computed via the same alignment as used in Eq. (2) and (3).

2.2. HMMs in Text-Dependent i-vector Systems

In [15], an HMM structure is proposed for *text-dependent* speaker verification, where phoneme recognizer is first trained with 3-state, GMM-based, mono-phone HMMs. Let F be the total number of mono-phones, $S = 3F$ be the number of all states, G be the number of Gaussian components per state, and $C = SG$ be the number of all individual Gaussians, and let (s, g) denote a Gaussian component g 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. The Viterbi algorithm is then used to obtain the alignment of the frames to the HMM states, and within each state s, the GMM alignment $\gamma_t^{(s,g)}$ for each frame t is computed. We can now re-interpret the pair (s, g) as one out of C Gaussians and we can substitute $\gamma_t^{(c)}$ in Eq. (2) and (3) by $\gamma_t^{(s,g)}$. Note that, due to the typically short duration of the phrases, not all phonemes are used in the phrase-specific HMM, therefore the alignment of frames to the Gaussian components can often be sparse.

It is worth mentioning that after calculating the zero- and first-order statistics for the training set, a single (phrase-independent) i-vector extractor was trained.

2.3. Frame Alignment Using DNNs

In this approach, it is assumed that the output of a DNN produces true posteriors (e.g. softmax function at the output). These posteriors can be then directly used for i-vector extraction in Eq. (2) and (3). As described in the Introduction section, it has been found that a DNN trained for phoneme classification produces excellent results.

Let us note, that the output of this system has to be used for computing the base UBM normalization parameters in (6) and in (7). In our experiments, the topology of this network is identical to the one used for BN feature extraction except for the number of the output nodes, as described in the following section.

2.4. i-vector Normalization and Scoring

In our experiments i-vectors are length-normalized [33], and further normalized using phrase dependent regularized Within-class Covariance Normalization (WCCN) [34]. In the case of standard WCCN, i-vectors are transformed using 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 *text-dependent* task, we only found WCCN effective when applied in phrase dependent manner (i.e. for trials of a specific phrase, $\boldsymbol{\Sigma}_{wc}$ is estimated only on the training utterances of that phrase) [15]. With RSR2015 dataset, however, this leaves us only very limited amount of data for estimating phrase specific

matrices $\mathbf{\Sigma}_{wc}$. For this reason, we found it necessary to regularize $\mathbf{\Sigma}_{wc}$ by adding a small constant to the matrix diagonal [15, 35].

Simple cosine distance scoring is used in all our experiments followed by phrase dependent s-norm score normalization [5].

3. Bottleneck Features

Bottleneck neural network refers to 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 [36], which proved to be very effective in our previous *text-independent* speaker recognition experiments [26]. This architecture is based on a cascade of two such BN DNNs. The BN output of the first network is *stacked* in time, defining context-dependent input features for the second DNN (hence the term Stacked Bottleneck Features). The input features to the first stage DNN are 24 log Mel-scale filter bank outputs augmented with 13 fundamental frequency features [36] and normalized using conversation-side based mean subtraction. The outputs from the BN 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). 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. See [26, 36] for more details on the exact structure of this architecture. In all our experiments the extracted BN features are 80-dimensional.

The BN DNNs are trained to discriminate between triphone tied-state targets. Using a pre-trained GMM/HMM ASR system, we can cluster triphone states to the desirable number of targets (DNN outputs) [36]. The same ASR system is used to force-align the data for DNN training in order to obtain the target labels. We use two different DNNs in our experiment, both trained on Switchboard data (8 kHz, conversational telephone speech). The primary DNN for extracting BN features is trained to classify 8000 triphone state target. The second DNN with 1011 targets is primarily intended for DNN based alignment as described in Section 2.3. However, since this second DNN also uses the BN architecture, we also examine the BN features extracted using this network.

4. Experimental Setup

4.1. Data

We report our results on RSR2015 data set Part I [10]. This dataset comprises recordings from 157 male and 143 female speakers, each pronouncing 30 different phrases in 9 distinct sessions. For each speaker and each phrase, three sessions are used for enrollment, while the remaining are used for testing. This data is further divided into three disjoint speaker subset: *background*, *development* and *evaluation set*. In our experiments, only the *background set* is used for training. It is used to train gender independent UBM and i-vector extractor and phrase dependent regularized WCCN [15, 34]. It is also used to extract scores for phrase dependent s-norm. All results are reported for the *evaluation sets*. The *development set* is not used at all in this work. Note that we use exactly the same setup (training data and trial set for evaluation) as used in [12]. Therefore, our results should be directly comparable with the best results reported in Table 6 in that paper.

The Switchboard data is used for training of DNNs as described in Section 3.

4.2. Features

We have experimented with few different configuration for the standard cepstral features. For the experiments reported in this paper, we have selected 39-dimensional PLP features and 60-dimensional MFCC features, which perform slightly better for females and males, respectively. Moreover, combination of these two feature sets performs particularly well in fusion. Both PLP and MFCC are extracted from 16 kHz signal using HTK [37] with a similar configuration: 25 ms hamming windowed frames with 15 ms overlap. For each utterance, the features are normalized using cepstral mean and variance normalization after dropping the (initial and final) silence frames.

Beside the cepstral features, two versions of 80-dimensional DNN based bottleneck features are used in our experiments as described in Section 3. Note that these features are extracted from RSR data down-sampled to 8 kHz as, at the time of running these experiments, we only had available BN DNN trained on 8 kHz conversational telephone Switchboard data. Therefore, for comparison, we also report results with 8 kHz version of MFCC features.

4.3. Systems

All reported results are obtained with i-vector based systems. The 400-dimensional i-vectors are length-normalized [33], and further normalized using phrase dependent regularized WCCN as described in section 2.4. Cosine distance is then used to obtain speaker verification scores, which are further normalized using phrase dependent s-norm.

Results are reported for individual i-vector based systems, which differ in the input features (MFCC, PLP, BN or their combination) and in the method for aligning speech frames to the Gaussian components as described in Section 2. The three 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 number of 936 Gaussian components) and 3) DNN with 1011 outputs (corresponding to 1011 Gaussian components in the i-vector extraction model). All three alignment methods therefore result in i-vector model of a similar size.

5. Results

5.1. GMM, HMM and DNN Alignment Comparison

In Table 1, we analyze the effect of the three different alignment techniques for i-vector extraction from Section 2. The DET curves for few systems selected from Table 1 are also shown in Figure 1 and Figure 2 for male and female trials, respectively.

The first section of the table 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 [15], where this method is proposed and analyzed. DNN based alignment performs comparably to HMM, even though it does not rely on the exact phrase transcription. Note that the nature of the DNN based

Table 1: *Comparison of different features and alignment methods in terms of Equal Error Rate (EER) and Normalized Detection Cost Function as defined for NIST SRE08 (*$\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{old}}$*) and NIST SRE10 (*$\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{new}}$*). Note that MFCC features are extracted from 16kHz speech signal, while BN features are extracted only from 8kHz speech signal.*

		Male			**Female**		
Features	Alignment	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	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	0.36	0.0203	0.1286	0.39	0.0218	0.1441
BN	GMM	0.59	0.0325	0.1564	0.40	0.0201	0.1066
	HMM	0.48	0.0242	0.1446	0.33	0.0151	0.0845
	DNN	0.77	0.0428	0.2026	0.59	0.0296	0.1416
MFCC+BN	GMM	0.31	0.0176	0.0955	0.28	0.0144	0.0898
	HMM	**0.30**	**0.0148**	**0.0927**	**0.27**	**0.0134**	**0.0809**
	DNN	0.43	0.0236	0.1410	0.45	0.0255	0.1291

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 context; the alignment units are tied triphone states rather than Gaussian components in mono-phone states. Also, the DNN is discriminatively trained on large amount of speech data and using different features, while HMMs are trained only on the small amount of RSR2015 *background data*. On the other hand, the HMM based method leads to much more compact representation as there is just single model (and features) used for both the alignment and the rest of the i-vector extraction. Furthermore, we only report results on our evaluation set where all the target and non-target trials share the same phrase in both the enrollment and the test utterance. In other words, this setup assumes that both the target speakers and imposters always know and pronounce the correct phrase. However, HMM alignment would offer much better performance if the evaluation trials included also wrong phrase trials due to checking the correctness of the phrase using the HMM structure and the Viterbi alignment.

The second section of Table 1 reports 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. Most likely, this can be attributed to the better phone-like feature space clustering obtained with GMM trained on BN features. The good performance of the BN features is striking, especially when we realize that the BN features perform better than MFCC with the GMM alignment, even though the BN features are extracted only from 8 kHz speech signal. In fact, the performance with similar MFCC features extracted from 8 kHz signal is very poor (1.84% and 2.19% EER for males and females, respectively). The BN features still significantly benefit from the HMM based alignment, although not as much as MFCC features. Interestingly, BN features fail to perform well in combination with the DNN based alignment.

In the third section of Table 1, results are shown for concatenated MFCC+BN features (60+80=140 dimensions). In [26], superior performance was reported for *text-independent* speaker recognition with i-vector based system trained on such features. Here, we verify that concatenated MFCC+BN features provide excellent performance also for the *text-dependent* task.

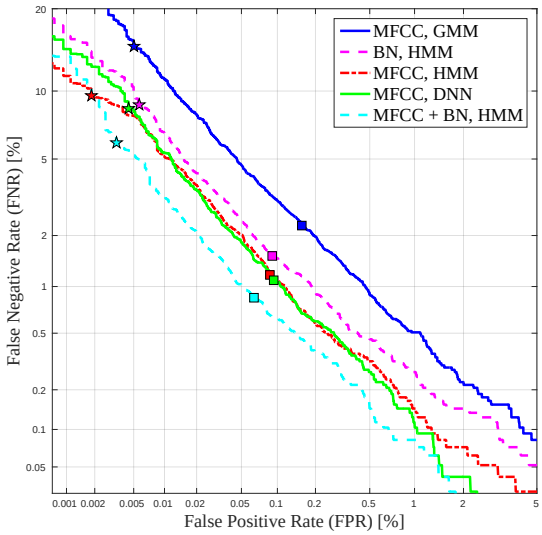

Figure 1: *DET curves for different methods of extracting posterior probabilities for male trials. The square and star markers corresponds to* $\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{old}}$ *and* $\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{new}}$ *operating points, respectively.*

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 guaranties 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.

5.2. Fusion Results

Table 2 shows results for different strategies of combining features and systems. The DET curves for few selected systems are also shown in Figure 3 and Figure 4 for male and female tri-

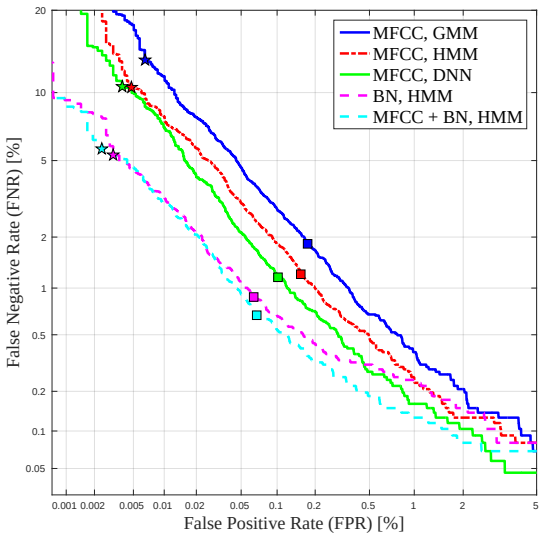

Figure 2: *DET curves for different methods of extracting posterior probabilities for female trials. The square and star markers corresponds to* $\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{old}}$ *and* $\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{new}}$ *operating points, respectively.*

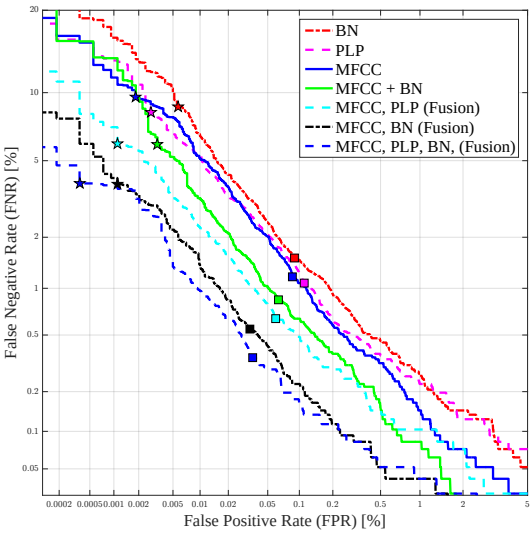

Figure 3: *DET curves for BN features with two best configurations from MFCC and PLP for male trials. The square and star markers correspond to* $\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{old}}$ *and* $\mathrm{NDCF}^{\mathrm{min}}_{\mathrm{new}}$ *operating points, respectively.*

als, respectively. Since the HMM based alignment turned out to generally provide the best performance in the previous experiments, all the following results are reported for this alignment method. Note that Table 2 repeats parts of the results from Table 1 to facilitate the comparison.

The first section of Table 2 shows results for the individual systems based on different features. Newly added are results for PLP features, which perform better compared to MFCCs for female trials, but slightly worse for male trials.

Our primary DNN for extracting BN features is trained to classify 8000 target triphone tied states. On the other hand, the alignment DNN from the previous experiments is trained only with 1011 target triphone states in order to keep the size of the corresponding i-vector extractor model reasonable and comparable to the GMM and HMM based models. Although unnecessary, we use BN topology also for the alignment DNN. Therefore, for comparison, we show also the result with the 80-dimensional BN1011 features extracted using the DNN with 1011 targets, which was primarily intended for the alignment. As can be seen from the results, the BN features from the fine-grained DNN with 8000 outputs significantly outperforms the BN1011 features.

The second section of Table 2 shows results for 140-dimensional MFCC+BN and 119-dimensional PLP+BN concatenated features. Both concatenated features performs comparably. In both cases, the concatenated features perform significantly better than the individual constituent features.

The third section of Table 2 contains results for score level fusion of systems with individual MFCC, PLP or BN features. In this work, we use the most trivial fusion, where the scores from the individual systems are simply averaged (given the equal weight). Interestingly, the score level fusion is very effective and, in contrary to our experience from *text independent* task, it brings larger improvements than concatenation of cepstral and BN features (i.e. fusion at the feature level). The likely reason for such behavior is the small amount of RSR2015 training data, which might not be sufficient to train the larger model based on the higher-dimensional concatenated features. Even the fusion of the two system based on MFCC and PLP cepstral features performs comparably or better than the systems with MFCC+BN and PLP+BN features. Nevertheless, score level fusion of cepstral features with BN provides superior performance. Clearly the best results reported in this work are obtained with three-fold score level fusion of MFCC, PLP and BN based system.

6. Conclusions

This work verified that the successful DNN based approaches to *text-independent* speaker recognition are very effective for the RSR2015 *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 [15]. In the case of the baseline system, the statistics are to be collected using a 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 DNN based alignment, where no transcription is necessary.

Furthermore, excellent performance was obtained with DNN based bottleneck features especially when concatenated with the standard cepstral features. Our experiments support the hypothesis that a GMM trained on the bottleneck results in a superior partitioning of the feature space into phone like clusters: The standard i-vector approach based GMM-UBM provides performance similar to the phone transcription supervised HMM based method. Note however that the HMM based alignment can still significantly help in rejecting wrong-phrase utter-

Table 2: *Results for different features, concatenated features and score fusions with HMM based systems. Note that MFCC and PLP cepstral features are extracted from 16kHz speech signal, while BN features are extracted only from 8kHz speech signal.*

	Male			**Female**		
Features	EER [%]	$\text{NDCF}^{\text{min}}_{\text{old}}$	$\text{NDCF}^{\text{min}}_{\text{new}}$	EER [%]	$\text{NDCF}^{\text{min}}_{\text{old}}$	$\text{NDCF}^{\text{min}}_{\text{new}}$
MFCC	0.37	0.0204	0.1142	0.49	0.0275	0.1533
PLP	0.41	0.0217	0.1103	0.42	0.0207	0.1029
BN	0.48	0.0242	0.1446	0.33	0.0151	0.0845
BN1011	0.58	0.0308	0.1780	0.44	0.0193	0.1060
MFCC+BN	0.30	0.0148	0.0927	0.27	0.0134	0.0809
PLP+BN	0.27	0.0149	0.1019	0.27	0.0124	0.0627
MFCC, PLP fusion	0.25	0.0123	0.0712	0.27	0.0139	0.0721
MFCC, BN fusion	0.15	0.0088	0.0493	**0.16**	0.0078	0.0315
PLP, BN fusion	0.18	0.0096	0.0637	0.17	0.0073	0.0326
MFCC, PLP, BN fusion	**0.13**	**0.0070**	**0.0424**	**0.16**	**0.0058**	**0.0299**

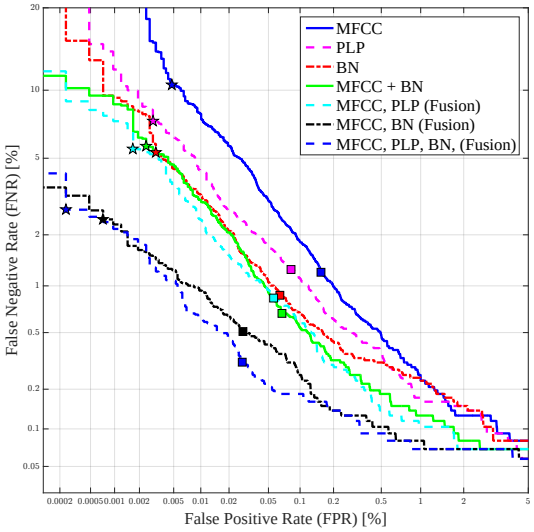

Figure 4: *DET curves for BN features with two best configurations from MFCC and PLP for female trials. The square and star markers correspond to* $\text{NDCF}^{\text{min}}_{\text{old}}$ *and* $\text{NDCF}^{\text{min}}_{\text{new}}$ *operating points, respectively.*

ances, which are not present in our evaluation data.

The best results reported in this paper were obtained with a simple score level fusion of the HMM based i-vector systems, each trained on different cepstral or bottleneck features.

7. References

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

[2] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1435–1447, 2007.

[3] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[4] Simon JD Prince and James H Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[5] Patrick Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey-The Speaker and Language Recognition Workshop*, 2010, p. 14.

[6] Anthony Larcher, Kong Aik Lee, Bin Ma, and et al., "Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7673–7677.

[7] Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li, "The RSR2015: Database for text-dependent speaker verification using multiple pass-phrases," in *Interspeech*, 2012.

[8] Hagai Aronowitz, "Text dependent speaker verification using a small development set," in *Odyssey-The Speaker and Language Recognition Workshop*, 2012.

[9] Sergey Novoselov, Timur Pekhovsky, Andrei Shulipa, and Alexey Sholokhov, "Text-dependent GMM-JFA system for password based speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 729–737.

[10] Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li, "Text-dependent speaker verification: Classifiers,

databases and RSR2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

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

[12] Patrick Kenny, Themos Stafylakis, J Alam, Pierre Ouellet, and Marcel Kockmann, "Joint factor analysis for text-dependent speaker verification," *Odyssey-The Speaker and Language Recognition Workshop*, 2014.

[13] Patrick Kenny, Themos Stafylakis, Pierre Ouellet, and Mohammad Jahangir Alam, "JFA-based front ends for speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1705–1709.

[14] Hossein Zeinali, Elaheh Kalantari, Hossein Sameti, and Hossein Hadian, "Telephony text-prompted speaker verification using i-vector representation," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4839–4843.

[15] Hossein Zeinali, Hossein Sameti, and Lukas Burget, "HMM-based phrase-independent i-vector extractor for text-dependent speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, in.press.

[16] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Moray McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[17] Garcia-Romero D., Zhang X., McCree A., and Povey D., "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *SLT*, 2014.

[18] Daniel Garcia-Romero and Alan McCree, "Insights into deep neural networks for speaker recognition," in *Interspeech*, 2015.

[19] George E Dahl, Dong Yu, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[20] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[21] Patrick Kenny, Vishwa Gupta, Themos Stafylakis, P Ouellet, and J Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Odyssey-The Speaker and Language Recognition Workshop*, 2014.

[22] Frantisek Grezl, Martin Karafiát, and Lukas Burget, "Investigation into bottle-neck features for meeting speech recognition," in *INTERSPEECH*, 2009, pp. 2947–2950.

[23] Sibel Yaman, Jason Pelecanos, and Ruhi Sarikaya, "Bottleneck features for speaker recognition," in *Odyssey-The Speaker and Language Recognition Workshop*, 2012, vol. 12, pp. 105–108.

[24] Pavel Matejka, Le Zhang, Tim Ng, HS Mallidi, Ondrej Glembek, Jeff Ma, and Bing Zhang, "Neural network bottleneck features for language identification," *Odyssey-The Speaker and Language Recognition Workshop*, pp. 299–304, 2014.

[25] Karel Vesely, Martin Karafiát, Frantisek Grezl, Marcel Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 336–341.

[26] Pavel Matejka, Ondrej Glembek, Ondrej Novotny, Oldrich Plchot, Frantisek Grezl, Lukas Burget, and Jan Cernocky, "Analysis of DNN approaches to speaker identification," in *ICASSP*, 2016.

[27] Fred Richardson, Douglas A. Reynolds, and Najim Dehak, "A unified deep neural network for speaker and language recognition," in *Interspeech*, 2015.

[28] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Jorge Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[29] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-End text-dependent speaker verification," *arXiv preprint arXiv:1509.08062*, 2015.

[30] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[31] Omid Ghahabi and Juan Hernando, "Deep belief networks for i-vector based speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1700–1704.

[32] Yao Tian, Meng Cai, Liang He, and Jia Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification," in *Interspeech*, 2015.

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

[34] Andrew O Hatch, Sachin S Kajarekar, and Andreas Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Interspeech*, 2006.

[35] Jerome H Friedman, "Regularized discriminant analysis," *Journal of the American statistical association*, vol. 84, no. 405, pp. 165–175, 1989.

[36] Martin Karafiát, František Grézl, Karel Veselý, Mirko Hannemann, Igor Szőke, and Jan Černocký, "BUT 2014 Babel system: Analysis of adaptation in NN based systems," in *Interspeech*, 2014, pp. 3002–3006.

[37] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al., *The HTK book*, vol. 2, Entropic Cambridge Research Laboratory Cambridge, 1997.