Odyssey 2018 The Speaker and Language Recognition Workshop
26-29 June 2018, Les Sables d'Olonne, France

Spoken Pass-Phrase Verification in the i-vector Space

Hossein Zeinali [1,2,3], *Lukáš Burget* [2], *Hossein Sameti* [1], *Jan "Honza" Černocký* [2]

[1] Sharif University of Technology, Tehran, Iran
[2] Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Czech Republic
[3] Sharif DeepMine Ltd., Tehran, Iran

hsn.zeinali@gmail.com, burget@fit.vutbr.cz, sameti@sharif.edu, cernocky@fit.vutbr.cz

Abstract

The task of spoken pass-phrase verification is to decide whether a test utterance contains the same phrase as given enrollment utterances. Beside other applications, pass-phrase verification can complement an independent speaker verification subsystem in text-dependent speaker verification. It can also be used for liveness detection by verifying that the user is able to correctly respond to a randomly prompted phrase. In this paper, we build on our previous work on i-vector based text-dependent speaker verification, where we have shown that i-vectors extracted using phrase specific Hidden Markov Models (HMMs) or using Deep Neural Network (DNN) based bottle-neck (BN) features help to reject utterances with wrong pass-phrases. We apply the same i-vector extraction techniques to the stand-alone task of speaker-independent spoken pass-phrase classification and verification. The experiments on RSR2015 and RedDots databases show that very simple scoring techniques (e.g. cosine distance scoring) applied to such i-vectors can provide results superior to those previously published on the same data.

1. Introduction

Utterance Verification (UV) is the task of confirming the content of a given utterance and answering the question of whether the user uttered the prompted pass-phrase or not. In this paper, we focus on spoken pass-phrase verification, where one or more spoken examples are given for the required pass-phrase. In other words, the task is to verify whether a test utterance (from, possibly, a previously unseen speaker) contains the same pass-phrase as a given enrollment utterance or a set of enrollment utterances. UV is a subtask of text-dependent Speaker Verification (SV), where the correctness of the uttered pass-phrase needs to be verified together with the speaker identity. Here, the UV component helps to prevent from replay attacks using a random utterance of the target speaker.

A single model is often used in text-dependent SV to jointly address both UV and SV tasks. For example, in our previous work on i-vector based text-dependent SV [1, 2, 3], HMMs constructed specifically for each pass-phrase were used to extract sufficient statistics in order to make the resulting i-vectors both speaker and phrase specific. Further text-dependent SV experiments have shown that it is enough to use the more conventional Universal Background Model - Gaussian Mixture Model (UBM-GMM) if the i-vectors are extracted from BN features [4, 5]. The frame-by-frame BN features are obtained from a DNN, which is trained to extract phonetic information from the acoustic context (300 ms) around the current frame. I-vectors extracted from such BN features contain lots of information about the phonetic content of the corresponding utterances and are very good for rejecting utterances with incorrect pass-phrases.

Although a single model can be used to jointly address both UV and SV tasks, there is still good reason to have stand-alone (speaker-independent) system for utterance verification or classification. In text-dependent SV, for example, replay attacks with pre-recorded correct pass-phrases are very difficult to reject. A possible way to tackle this problem is to use anti-spoofing techniques based on detecting typical distortions in recorded and replayed audio [6] or using audio fingerprinting [7] to detect a replay of an enrollment utterance. However, these techniques are often not very reliable. An alternative is to use a liveness detection using a separate UV subsystem as follows: in one step, a standard text-dependent SV is used to verify the speaker, while in the second step the user is prompted some random phrase, which he needs to pronounce to prove his responsiveness. Speaker identity can be verified from this second phrase in more text independent fashion. More importantly, the correctness of the phrase can be verified by the UV subsystem. The prompted random phrase can be in a textual form or can be represented by audio recording. The later case is of our main interest. Note that the UV techniques can be also applied to other problems than the text-dependent SV. An example can be re-scoring detections in keyword spotting or query-by-example system [8].

In this work, we experiment with the aforementioned i-vector based text-dependent SV techniques. However, we apply these techniques to the stand-alone task of (speaker-independent) spoken pass-phrase verification or classification. We show that the i-vectors extracted in the described way contain predominantly information about the lexical content of the utterance and are therefore excellent representations for this task. We also show that our solution based on the simple i-vector representation outperforms the previously proposed and often more computationally complex methods, which serve as our baseline [9].

2. Baseline Utterance Verification Methods

The effort on UV described in the literature is quite limited. In [9], four systems are described, which constitute a good example of the standard techniques for UV. We use these systems and the corresponding results as our baseline. In some of our experiments with i-vector based UV, the same setup as in [9] is used to make the results directly comparable. Here, we provide the only brief description of these baseline systems. For more detailed description, we kindly refer the reader to [9].

The system denoted as **UV1** uses Mel-Frequency Cepstral Coefficients (MFCCs) with their first and second order

10.21437/Odyssey.2018-52

derivatives and a GMM-UBM with 512 Gaussian components trained on TIMIT data. The utterance models are adapted from the GMM-UBM using the standard relevance maximum-a-posteriori (MAP) adaptation [10] and the log-likelihood ratio between the utterance and the UBM serves as the UV score. Note that this technique only models the distribution of acoustic features in the training utterances, but does not try to model the temporal structure of the uttered phrases.

The system denoted as **UV2** uses 5-state HMM with the left-to-right topology to model the temporal structure of utterances. Each state is modeled using a GMM, which is MAP adapted in a similar manner and from the same GMM-UBM as in the case of the system UV1. Viterbi alignment of frames to HMM states is used to train phrase specific models on training utterances and to evaluate the log-likelihood ratio score for the test utterances.

The **UV3** system uses perhaps the most conventional approach to spoken utterance verification: dynamic time warping (DTW) [11] is used to frame-align utterances and to calculate the distances between the utterances. Euclidean distance between MFCC feature vectors is used as the frame-to-frame distortion. Note that the DTW based UV could be further improved by using more sophisticated frame-to-frame distortions [12] or by calibrating the resulting DTW scores to make them proper UV log-likelihood ratios [13]. These improvements are, however, not considered in this work.

UV4 makes use of a DNN based automatic speech recognition (ASR) system trained on TIMIT data using Kaldi [14] toolkit. Each test utterance is forced-aligned to the known reference transcript of a given pass-phrase and the acoustic score (pseudo log likelihood) for this alignment is used as the UV score. Note that this system performs UV using the pass-phrase given as text, unlike the other methods described in this paper, which rely on spoken pass-phrase.

3. i-vector Based Utterance Verification

In this work, we use i-vectors as fixed length low-dimensional representations of speech utterances. First, i-vectors were proposed for the task of text-independent speaker recognition [15], but soon became popular for other tasks of utterance level classification or verification such as language, gender, signature, age or emotion recognition [16, 17, 18, 19]. In the probabilistic model for i-vector extraction, a low-dimensional latent variable is used to representing utterance specific GMM. I-vector is the MAP point estimated of the latent variable adapting the corresponding GMM to a given speech utterance. For more details on the i-vector model, we kindly refer the reader to other sources [15, 5]. Here, we only recall that the i-vector can be inferred from sufficient statistics, which are collected from the speech utterance. To collect the sufficient statistics, we need an alignment of speech frames to i-vector model Gaussian components. This alignment is traditionally obtained using an underlying UBM-GMM.

3.1. HMM based frame alignment methods

In our previous works on text-dependent SV [1, 2] and also text-prompted SV [3], i-vectors were extracted using HMM based alignment. For this purpose, phoneme recognizer is first trained, where mono-phone 3-state HMMs are used with state distributions modeled using GMMs. Given the known transcriptions of enrollment and test utterances, the phrase specific HMMs are constructed from the mono-phone HMMs. The Viterbi algorithm is then used to obtain the alignment of the frames to the HMM states in order to collect the sufficient statistics. 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. More details on this i-vector extraction method can be found in [1, 5].

For text-dependent SV, it was shown [1, 2] that this alignment extraction strategy produces more phrase specific i-vectors, which are especially effective for rejecting utterance with wrong pass-phrases. For the same reason, this technique is also suitable for utterance verification task as demonstrated in our experiments. One the drawback of this approach is that we need to know the phrase specific phone sequence for constructing the corresponding HMM.

3.2. Bottleneck features

MFCCs were conventionally used as the speech features for i-vector extraction. More recently, however, significant improvements were obtained for both text-dependent [4, 5] and text-independent [20, 21, 22] verification task from using BN features or concatenated MFCC+BN features. Note that BN features were previously successfully used also in other areas of speech processing [23, 24, 25].

BN features are frame-by-frame extracted using a bottleneck DNN, which is typically trained for phone classification. Bottleneck DNN is a neural network 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 output of the bottleneck layer where the relevant information is compressed into a low dimensional vector. In this work, we use more elaborate architecture for BN features called Stacked Bottleneck Features [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. The input features to the first stage DNN are 36 log Mel-scale filter bank outputs augmented with 3 fundamental frequency features [26] 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 at least 30 frames (300 ms) of the input features in the context around the current frame. Therefore, each BN feature vector contains important information about the phonetic context around the current frame, which is further propagated to the i-vector extracted from these features. This makes BN feature based i-vectors very phrase specific even when extracted using the conventional UBM-GMM model (i.e. there is no need for the HMM based alignment), which was previously demonstrated in text-dependent SV experiments [4, 5].

3.3. Scoring methods

In our experiments, we consider both the task of close-set pass-phrase classification and open-set pass-phrase verification. To classify or compare i-vectors, we use only two very simple techniques, namely Linear Gaussian Classifier (LGC) and cosine similarity scoring.

3.3.1. Linear Gaussian Classifier (LGC)

For each class (pass-phrase) $i = 1 \dots K$, LGC assumes Gaussian distribution of i-vectors $\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$. Each class is modeled by its own mean vector $\boldsymbol{\mu}_i$. All the classes, however, share the same average within-class covariance matrix $\boldsymbol{\Sigma}$, which is typically estimated as

$$\boldsymbol{\mu}_i = \frac{1}{N_i} \sum_{n=1}^{N_i} \mathbf{w}_i^n \tag{1}$$

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{i=1}^{K} \sum_{n=1}^{N_i} (\mathbf{w}_i^n - \boldsymbol{\mu}_i)(\mathbf{w}_i^n - \boldsymbol{\mu}_i)^T , \tag{2}$$

where N_i is the number of training samples (i-vectors) for phrase i and $\mathbf{w}_i^n$ is the n^{th} training sample of phrase i. Once the model is trained on the training (or enrollment) utterances, evaluation data can be classified by simply selecting the class with the highest posterior probability:

$$P\,(i|\mathbf{w}) = \frac{\mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}) P(i)}{\sum_{k=1}^{K} \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) P(k)} , \tag{3}$$

were we assume equal priors $P(i)$ for all classes. To be consistent with results from [9], we also report the performance in terms of Equal Error Rate (EER) for LGC, where the posterior probabilities serve as the verification score for the corresponding classes. In this case, however, we cannot talk about open-set verification as the score from the close-set of K phrases depends on each other through the normalization in the posterior probability calculation.

3.3.2. Cosine Similarity Scoring

Cosine similarity scores are also used in our experiments to perform classification and verification of i-vectors. In this case, the enrolled pass-phrase models are obtained as a simple average of training (or enrollment) i-vectors. Note that there is no need to estimate any covariance matrix for this scoring method, which makes it more robust for the cases where only few training examples are available. To perform classification of a test utterance, we can select the class with the highest cosine similarity score. For the detection task (i.e. to evaluated EER), we simply use the cosine similarity score as the verification scores. Note that in this case, verification scores for individual pass-phrases are completely independent of each other and the obtained EER can be correctly interpreted as open-set pass-phrase verification performance.

Again, to be consistent with results from [9], we alternatively normalize the cosine similarity scores using the so-called Max-Norm method. In this case, for each test utterance, the maximum of cosine scores over all other the $K - 1$ phrases is subtracted from the original cosine scores. The same normalization is also used for some of the results from [9], which are also presented for comparison in Table 2. Although the normalization (seemingly) improves the classification and verification results, we no more deal with the open-set verification problem just like in the case of LGC.

3.3.3. Motivation for simple classifiers

We have used t-SNE [27] to reduce 400-dimensional i-vectors extracted using UBM-GMM from MFCC+BN features into 2-dimensional space. The i-vectors were taken from all male speakers from the RSR2015 test set. Figure 1 shows the plot of the resulting vectors for 30 phrases of the RSR2015 database. Each point in the plot corresponds to one i-vector and is colored according to the phrase label. One can see that i-vectors from different phrases form nicely separated clusters in the t-SNE space. Moreover, all classes have roughly Gaussian distribution with the same within-class covariance matrix. Although, in general, t-SNE provides nonlinear transformation of the original space, the nicely separated clusters and simple distributions make us believe that pass-phrase verification should be an easy task in this i-vector space and simple scoring technique should be sufficient.

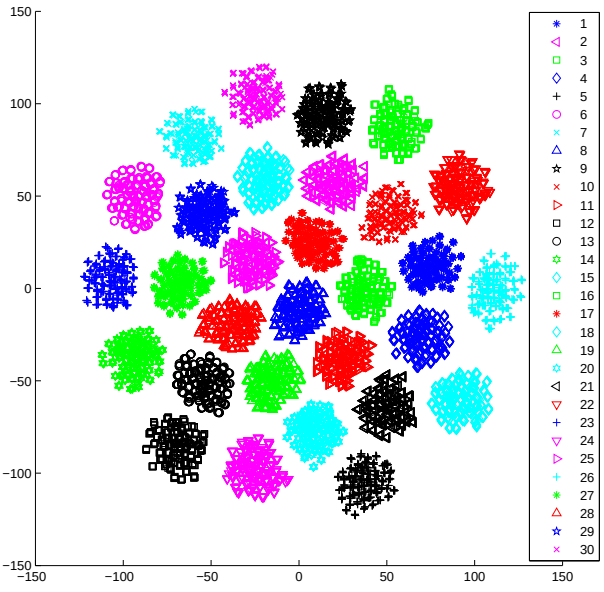

Figure 1: Male i-vectors from the RSR2015 database reduced to 2-dimensional space using t-SNE. There are 30 well separated clusters corresponding to different phrases.

4. Experimental setup

We report results on Part1 of the RSR2015 database [28] as well as Part1 of the RedDots database [29]. RSR2015 comprises recordings from 30 different phrases. The same phrases appear in three disjoint subsets of speakers *background*, *development* and *evaluation set*. Each speakers repeats each phrase 9 times. The male utterances from the *background set* (50 speakers) are used for training the classifiers. The results are reported on male part of the *evaluation set* (57 speakers). The *development set* is not used in our experiments.

Part1 of RedDots contains 49 male speakers, each pronouncing several times 10 common pass-phrases. For the results reported in Table 2, the evaluation setup defined at UEF university was adopted to make our results directly comparable with those previously reported in [9]. In this setup, all utterances from 9 different speakers are used to train the 10 pass-phrase models (In total, 1485 utterances are used for training, roughly 148 utterances per pass-phrase) and 30 other speakers are selected for the evaluation set. Note that 10 out of the 49 male speaker were not used at all in our experiments. For the results reported in Table 3 analyzing the performance for reduced amount of training data, several subsets of the 9 training speakers are used and evaluation set remains unchanged.

Table 1: Performance of the i-vector based methods on RSR2015 .

	Classification Error [%]	EER [%]
LGC	0.0	0.000
Cosine	0.0	0.007
Cosine Max-Norm	0.0	0.000

A UBM-GMM with 1024 components, alignment HMMs with 3 states and 8 components in each state and a 600-dimensional i-vector extractor are trained using LibriSpeech database [30] and the *background set* of RSR database. BN feature extractor was also trained on LibriSpeech [5].

In order to evaluate EERs, the verification scores for the individual pass-phrases are simply polled. We understand that it is questionable to use such pooled EER in the case of close-set problem, where the verification scores are normalized using the scores from the competing hypothesis (i.e. our results with LGC and Cosine Max-Norm). Nevertheless, we also include such results in order to allow for comparison with the baselines from [9]. For the RSR2015 database, each evaluation utterance forms one target trial and 29 non-target trials corresponding to the remaining pass-phrases. Similarly, one target and 9 non-target trials are formed for each evaluation utterance from RedDots.

5. Results

5.1. RSR2015 Results

First, we report results on male utterances from RSR2015 *evaluation set*. This is an example of a scenario where plenty of training examples are available for each of 30 pass-phrases ($50 \times 9 = 450$ training utterances per phrase). Further, we deal here with the ideal condition, where UBM and i-vector extractor are trained on training data of the same phrases. This leads to nearly faultless recognition performance for any of the scoring technique as presented in Table 1. In this case, we have chosen i-vectors extracted using UBM-GMM from MFCC+BN features, which was the configuration previously providing excellent performance in text-dependent SV task [4, 5].

5.2. RedDots Results

Table 2 shows results for more challenging RedDots database. The phrase models are still trained on relatively many examples. As mentioned in section 4 there are about 148 examples from 9 different speakers for each pass-phrase. But the data are recorded under more challenging conditions and the UBM and i-vector extractor are trained on data of mismatched phrases. Note also that, in the case of LGC, within-class covariance matrix (i.e. Eq. (2)) was estimated on RSR2015 on data of different phrases (i.e. in a phrase independent fashion). Only the class means were estimated on RedDots data.

The first section of Table 2 shows results obtained with the baseline systems, which were described in Section 2. These results are borrowed from Table 5 of [9] and are directly comparable with our result from the second section of Table 2. The results show that the proposed i-vectors (again UBM-GMM and BN features are used) easily outperform even the fusion of the previously published baseline methods from [9]. We have manually inspected the utterances where the i-vector based systems made an error, and we have observed that those were most severely corrupted utterances (i.e. mispronunciation, only silence, etc). Note also the very good performance of the Cosine similarity with no normalization, which is the result for the true open-set pass-phrase verification task.

Table 2: Comparison of the i-vector bases methods with the baseline methods from [9] on RedDots data.

Method	No-Norm EER [%]	Max-Norm EER [%]	Classification Error [%]
UV1	9.31	2.08	–
UV2	5.54	1.11	–
UV3	24.81	7.80	–
UV4	16.60	4.56	–
Fused (UV1 ... UV4)	6.13	1.43	–
LGC	0.11	–	0.25
Cosine	0.61	0.10	0.25

Table 3: Comparison of features, alignment methods and different amount of training examples on RedDots. Three training i-vectors are used per speaker. The results are EERs [%]

		Number of Speakers				
Method	Feature / Align	**1**	**2**	**3**	**5**	**9**
LGC	MFCC / GMM	61.01	7.78	3.70	2.71	1.45
	MFCC / HMM	9.60	1.55	1.16	1.15	0.85
	MFCC+BN / GMM	39.11	1.10	0.21	0.15	0.14
Cosine	MFCC / GMM	24.54	16.7	12.9	10.1	7.17
	MFCC / HMM	19.19	9.58	7.18	4.87	3.02
	MFCC+BN / GMM	7.53	2.00	1.35	0.95	0.55
Cosine Max-Norm	MFCC / GMM	15.51	8.18	5.67	3.62	2.01
	MFCC / HMM	9.79	3.51	2.36	1.16	0.50
	MFCC+BN / GMM	2.52	0.35	0.30	0.20	0.10

From the results in Table 2, we can see that both LGC and Cosine distance perform similarly. This is understandable realizing the close relation between the two scoring methods: LGC with identity within-class covariance matrix applied to length normalized i-vectors[1] would produce class likelihood proportional to Cosine distance. In reality, the within-class covariance matrix will not be far from identity as the i-vector extractor is trained to produce standard normal distributed i-vectors. Moreover, the Max-Norm applied to Cosine distance scores can be seen as an approximation to the softmax normalization embedded in equation 3.

5.3. Features, Alignments and Amount of Training Data

Table 3 compares results obtained with the different proposed i-vector extraction variants: UBM-GMM vs. HMM alignment, MFCC vs. MFCC+BN features. For LGC, within-class covariance matrix was again estimated on RSR20105 data. The results show the degradation of the performance with the decreasing

[1] However, note that we do not apply the length normalization in the case of LGC scoring in our experiments.

number of training examples (and training speakers). Here, we use only three training examples per speakers and the columns of the table correspond to the number of speakers considered for the training. With only MFCC features, HMM alignment performs better than UBM-GMM in almost all cases. This is due to the HMM ability to model the temporal structure of individual phrases, which has been previously shown to be very effective for rejecting the wrong phrase trials [4, 1]. The best performance is achieved with MFCC+BN with UBM-GMM alignment. In this case, the information about the temporal structure of phrases is encoded directly in the BN features, which are extracted from a considerably large context window (i.e. more than 300 ms). This allows us to obtain the superior performance even with the simpler UBM-GMM alignment.

Again, excellent results can be obtained with MFCC+BN features and the simple Cosine similarity scoring without any normalization, considering that this corresponds to open-set verification task. In this case, acceptable performance is achieved just with 3 samples from 5 speakers (still outperforming all the baseline systems from Table 2), which might lead to very useful and practical applications.

The simplicity of the i-vector based scoring methods and the relatively low number of parameters that need to be estimated on the training data of matching pass-phrases makes our approach suitable also for the cases with very limited amount of training examples. As can be seen from the results, acceptable performance can be obtained even with only 2 training speakers.

In the case of only single enrollment speaker, the results for LGC based scoring seems to be quite unstable as compared to Cosine distance (e.g. note the surprisingly high 39.11% EER for MFCC+BN / GMM). Our further analysis revealed that this was due to the insufficient data used for the estimation of the LGC within-class covariance matrix. As mentioned above, the covariance matrix is pre-estimated on the RSR2015 data of mismatched pass-phrases. Estimating the covariance matrix on more (still mismatched) data helped alleviated this problem.

6. Conclusions

In this paper, we proposed simple but effective i-vector based spoken pass-phrase verification methods and evaluated them on two standard databases: RSR2015 and RedDots. Experimental results have shown the effectiveness of the methods, which achieved almost zero error rate on both databases and significantly outperformed previously published result.

The main reason for the excellent performance of these methods is the suitability of i-vectors for utterance verification. I-vector extracted from short duration utterance contains predominantly information about the phonetic content of the utterance. Therefore, such i-vectors naturally form phrase specific cluster in the i-vector space without any need for channel compensation and score normalization [1, 4], which are otherwise necessary for tasks like speaker verification.

The advantages of the proposed methods are simplicity, speed, very low overhead and excellent performance. Another interesting property is suitability of these methods for low resource scenarios, which is allowed by their good performance with little amount of training data.

Although the proposed methods have achieved near zero error rate on both databases, we can hardly say that the pass-phrase verification is a solved problem. Much larger databases with plenty of phrases will be necessary to reliably evaluate the verification methods and also to analyze the possible performance degradations due to the phrase similarity. This is an open topic for future works.

7. Acknowledgment

The work was supported by Czech Ministry of Education, Youth and Sports from Project No. CZ.02.2.69/0.0/0.0/16_027/0008371 and the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602" and also partially supported by Sharif DeepMine Ltd. company in Iran.

8. References

[1] Hossein Zeinali, Hossein Sameti, and Lukas Burget, "HMM-based phrase-independent i-vector extractor for text-dependent speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, pp. 1421–1435, 2017.

[2] Hossein Zeinali, Hossein Sameti, Lukas Burget, Jan Cernocky, Nooshin Maghsoodi, and Pavel Matejka, "i-vector/HMM based text-dependent speaker verification system for RedDots challenge," in *InterSpeech*, 2016, pp. 440–444.

[3] 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), IEEE International Conference on*, 2015, pp. 4839–4843.

[4] Hossein Zeinali, Lukas Burget, Hossein Sameti, Ondrej Glembek, and Oldrich Plchot, "Deep neural networks and hidden Markov models in i-vector-based text-dependent speaker verification," in *Odyssey-The Speaker and Language Recognition Workshop*, 2016, pp. 24–30.

[5] Hossein Zeinali, Hossein Sameti, Lukáš Burget, and Jan Černockỳ, "Text-dependent speaker verification based on i-vectors, deep neural networks and hidden Markov models," *Computer Speech & Language*, vol. 46, pp. 53–71, 2017.

[6] Tomi Kinnunen, Md Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," .

[7] Zhizheng Wu, Sheng Gao, Eng Siong Cling, and Haizhou Li, "A study on replay attack and anti-spoofing for text-dependent speaker verification," in *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*. IEEE, 2014, pp. 1–5.

[8] Haisheng Dai, Xiaoyan Zhu, Yupin Luo, and Shiyuan Yang, *An Utterance Verification Algorithm in Keyword Spotting System*, pp. 555–561, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[9] Tomi Kinnunen, Md Sahidullah, Ivan Kukanov, Héctor Delgado, Massimiliano Todisco, Achintya Kumar Sarkar, Nicolai Bæk Thomsen, Ville Hautamäki, Nicholas WD Evans, and Zheng-Hua Tan, "Utterance verification for text-dependent speaker recognition: A comparative assessment using the reddots corpus," in *InterSpeech*, 2016, pp. 430–434.

[10] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[11] Lawrence R Rabiner and Biing-Hwang Juang, *Fundamentals of speech recognition*, PTR Prentice Hall, 1993.

[12] Gilles Boulianne, "Language-independent voice passphrase verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4490–4494.

[13] Qian Liu, Zhang-qin Huang, Yi-bin Hou, and Rui Chen, "Utterance verification on DTW based speech recognition using likelihood," in *Computer Application and System Modeling (ICCASM), 2010 International Conference on*. IEEE, 2010, vol. 2, pp. V2–427.

[14] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, K. Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý, "The kaldi speech recognition toolkit," in *Proceedings of ASRU 2011*. 2011, pp. 1–4, IEEE Signal Processing Society.

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

[16] David Martınez, Oldrich Plchot, Lukás Burget, Ondrej Glembek, and Pavel Matejka, "Language recognition in ivectors space," *InterSpeech*, pp. 861–864, 2011.

[17] Hossein Zeinali and Bagher BabaAli, "On the usage of i-vector representation for online handwritten signature verification," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*. IEEE, 2017, vol. 1, pp. 1243–1248.

[18] Mohamad Hasan Bahari, ML McLaren, DA van Leeuwen, et al., "Age estimation from telephone speech using i-vectors," 2012.

[19] Rui Xia and Yang Liu, "Using i-vector space model for emotion recognition," in *InterSpeech*, 2012.

[20] Alicia Lozano-Diez, Anna Silnova, Pavel Matejka, Ondrej Glembek, Oldrich Plchot, Jan Pešán, Lukáš Burget, and Joaquin Gonzalez-Rodriguez, "Analysis and optimization of bottleneck features for speaker recognition," in *Odyssey-The Speaker and Language Recognition Workshop*, 2016, pp. 21–24.

[21] Yao Tian, Meng Cai, Liang He, and Jia Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification," in *InterSpeech*, 2015, pp. 1151–1155.

[22] Pavel Matejka, Ondrej Glembek, Ondrej Novotny, Oldrich Plchot, Frantisek Grezl, Lukas Burget, and Jan Cernocky, "Analysis of DNN approaches to speaker identification," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2016, pp. 5100–5104.

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

[24] 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.

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

[26] 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.

[27] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[28] 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.

[29] Kong Aik Lee, Anthony Larcher, Guangsen Wang, Patrick Kenny, Niko Brümmer, David van Leeuwen, Hagai Aronowitz, Marcel Kockmann, Carlos Vaquero, Bin Ma, et al., "The RedDots data collection for speaker recognition," in *InterSpeech*, 2015, pp. 2996–3000.

[30] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2015, pp. 5206–5210.