

Investigating the stacked phonetic bottleneck feature for speaker verification with short voice commands

Yichi Huang, Yuexian Zou*

ADSPLAB/Intelligent Lab, School of ECE, Peking
University
Shenzhen 518055, China

*{zouyx}@pkusz.edu.cn, yichihuang@pku.edu.cn

Yi Liu

IMSL Shenzhen Key Lab, PKU-HKUST Shenzhen Hong
Kong Institution

Abstract—Text-dependent speaker verification (SV) with short voice command (SV-SVC) has increasing demand in many applications. Different from conventional SV, SV-SVC usually uses short fixed voice commands for user-friendly purpose, which causes technical challenges compared with conventional text-dependent SV using fixed phrases (SV-FP). Research results show that the mainstream SV techniques are not able to provide good performance for SV-SVC tasks since they suffer from strongly lexical-overlapping and short utterance length problems. In this paper, we propose to fully explore the acoustic features and contextual information of the phonetic units to obtain better speaker-utterance related information representation for i-vector based SV-SVC systems. Specifically, instead of using MFCC only, the frame-based phonetic bottleneck (PBN) feature extracted from a phonetic bottleneck neural network (PBNN), the stacked phonetic bottleneck (SBN) feature, the cascaded feature of PBN and MFCC, the cascaded feature of SBN and MFCC (SBNF+MFCC) are extracted for developing i-vector based SV-SVC systems. Intensive experiments on the benchmark database RSR2015 have been conducted to evaluate the performance of our proposed i-vector SV-SVC systems. It is encouraged that the contextual information learnt from stacked PBNN does help and proposed i-vector SV-SVC system with (SBNF+MFCC) outperforms under experimental conditions.

Keywords—Text-dependent speaker verification, short voice command, phonetic bottleneck feature, i-vector

I. INTRODUCTION

Speaker verification (SV) is a binary classification task which aims to verifying a person's identity according to his/her voice. Speaker verification can be split into two categories: text-dependent and text-independent. In text-dependent speaker verification, the predefined speech phrases are used in enrollment and test phases. On contrary, in text-independent speaker verification, there are no limitation of speech phrases used. Research outcomes show that the text-dependent speaker verification techniques have advanced the text-independent speaker verification

techniques since additional phonetic information is available [1]. Moreover, it is noted that text-dependent speaker verification with fixed phrases, short voice commands and digits have increasing demands in practical speech applications such as keyword spotting, banking transactions, and voiceprint authentication systems. For text-dependent speaker verification, both person's identity and the content of spoken phrases are verified. Therefore, either the speaker identity or the phrase does not match will be considered as a non-target trial. As a result, for text-dependent speaker verification, three types of non-target trials will be considered, terms as *IC* (Impostor pronouncing, Correct lexical content), *TW* (Target speaker pronouncing, Wrong lexical content), and *IW* (Impostor pronouncing, Wrong lexical content), respectively.

So far, the state-of-the-art SV technique rooted in i-vector/PLDA framework [2-3] (denoted as i-vector SV in this paper). Latest progress of speaker verification has applied Deep Neural Networks (DNNs) to GMM-UBM model [4,5] or i-vector model [6]. Essentially, there are two approaches in implementing the DNN based speaker verification. One is to learn the features frame-by-frame for GMM-UBM SV system or i-vector SV system. Another approach is implemented in a new perspective where DNN is employed to estimate the frame alignment posteriors of the phonetic units, which is used to replace the GMM-UBM in i-vector model. For the first approach, the DNN is trained first in the frame-by-frame classification manner but with different conditions, which leads to the phonetic discriminative DNN based SV and speaker discriminative DNN based SV system, respectively. Specifically, for the former, the training pair is speech frame and phoneme (or triphone) and the DNN is trained to get the classification model between speech frames and their paired phonemes. For the latter, the training pair is the speech frame and speaker ID, then the DNN is trained for discriminating between speakers. At the end, the speaker-related information is stored in the hidden layers of the trained DNN model, which can be used as the feature vectors in some way [4]. Besides, the dimension of the hidden layer features is generally high and requires high computational cost. To address this problem, the dimension reduction techniques [4] or bottleneck layer approach are commonly

applied [7]. It is noted that i-vector SV with DNN features achieved the state-of-the-arts on public datasets for TD-SV using massive training data [6]. Part of reasons for this success may be that they fully explore the information of the phonetic units. However, we also noted that the predefined phrases can be split into phonetic units in sequences and the contextual information can be further investigated [8].

In principle, the SV with short voice commands (SV-SVC) is a special case of speaker verification task with two constraints: the shorter utterance and lexical overlapping. For example, in RSR2015 [1], for SV-SVC and SV with fixed phrases tasks, the speech duration ranges from 0.1s to 0.5s and ranges from 1s to 2s, respectively. Besides, from applications, we can see that the lexical content of different short voice commands strongly overlap, e.g., “Door open” and “Door close”. Thus, for SV with short voice commands, there is less phonetic variation available. Fortunately, some research work has shown that i-vector model is effective for retaining speaker and lexical content in the speech segments to some extent [9]. This is why i-vector model for SV with short voice commands is an appropriate choice. However, literature shows that the state-of-the-art i-vector SV with MFCC feature for SV with short voice commands tasks [1] still suffers from the short duration and the lexical similarity of the commands. Motivated by our previous DNN-based SV work [10] and the discussions above, in this paper, we propose to fully explore the contextual information of the phonetic units together with the MFCC features to improve the performance of SV with short voice commands technique under i-vector framework. Instead of using MFCC feature only, the frame-based phonetic bottleneck feature (PBN-feature) extracted from a phonetic bottleneck neural networks (PBNN), the stacked bottleneck feature (SBN-feature) [11], the cascaded feature of PBN and MFCC (PBNF+MFCC), the cascaded feature of SBN and MFCC (SBNF+MFCC) are extracted as speech feature vector for i-vector SV. To evaluate the impact of different feature vectors, intensive experiments are conducted on the benchmark database RSR2015. The experimental results are evaluated and compared under different conditions, such as TW , IC and IW , respectively. Some findings and discussions are given accordingly.

II. I-VECTOR FRAMEWORK

The i-vector approach [12] was proposed by Dehak and received widely investigation in the past decade. Essentially, the i-vector approach assumes that most relevant speaker information lives in a low-dimensional space called total variability space. Each utterance can be represented as a fixed-length vector called *i-vector* in this space. In principle, the utterance-dependent GMM supervector is represented as

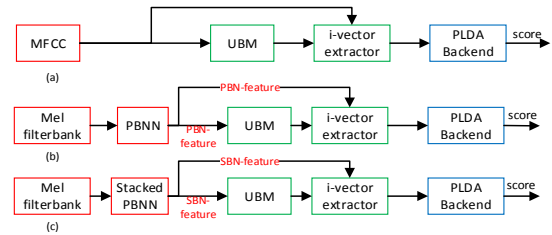


Figure 1. Block diagram: (a) baseline i-vector SV-SVC, (b) our proposed PBNF-i-vector SV-SVC, (c) our proposed SBNF-i-vector SV-SVC systems

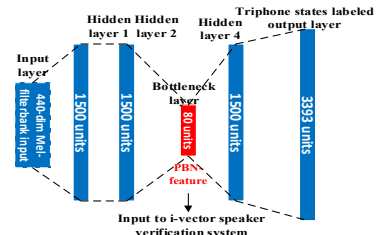
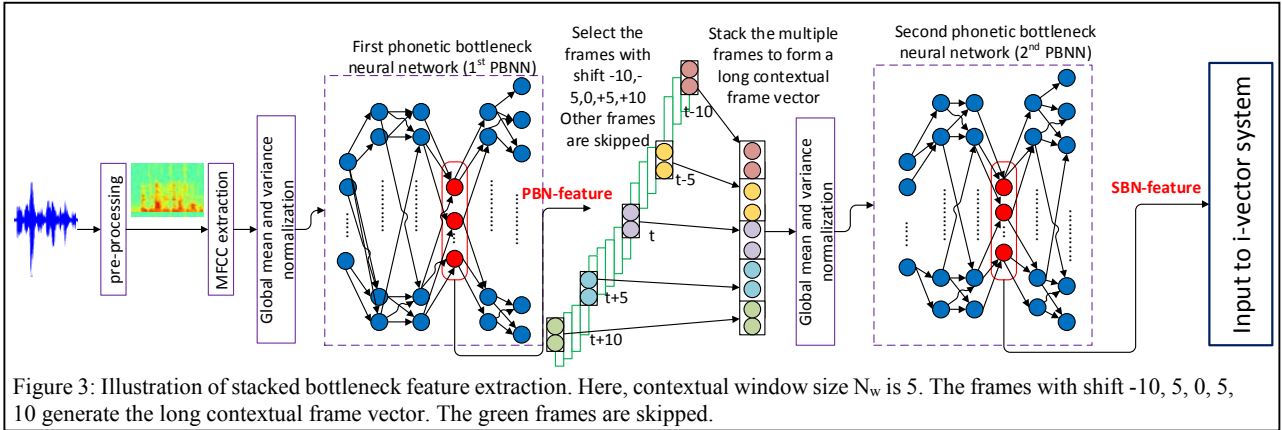


Figure 2 Illustration of PBNN used in this study

$$M = m + Tw \quad (1)$$

where m is mean supervector of Universal Background Model (UBM), T is a low-rank matrix representing speaker and session variability, and the latent variable w is called i-vector.

The i-vector SV consists of three key stages: the collection of sufficient statistics (SS), the extraction of i-vectors and a PLDA backend. The collection of SS is a process where a sequence of feature vectors (e.g. MFCC) are represented by the Baum-Welch statistics obtained with respect to UBM. These high-dimensional statistics are converted into a single low-dimensional feature vector called i-vector. After i-vectors are extracted, a PLDA model is used to produce verification scores. In conventional i-vector SV system, the short-time spectral features (e.g. MFCC) are most widely used. Besides, we found that MFCCs also have been employed in i-vector SV with short voice commands system [1]. In this study, under i-vector SV framework, we propose to extract novel features for better representing speaker and content discriminative information, which is a possible good way to improve the performance of SV with short voice commands systems. The initiatives lie in two aspects: fully making use of the available text content with its sequence information and taking advantage of i-vector SV.



III. PROPOSED SV-SVC SYSTEM

A. Overview of the proposed SV-SVC system

The block diagram of our proposed PBNF-i-vector SV-SVC and SBNF-i-vector SV-SVC systems is illustrated in Figure 1. For comparison, the baseline i-vector SV with short voice commands system is also given. Obviously, these SV-SVC systems all consist of three parts: front-end feature extraction (modules in red), speaker modeling (modules in green) and back-end scoring (modules in blue). The difference of three systems lies in the feature extraction part. As discussed in Section 2, in this study, our focus is to extract more effective features on characterizing the short voice command and speaker discriminative information. As shown in the red modules in Figure 1, we proposed to fully make use of phonetic bottleneck features, stacked bottleneck features and spectral-based features including MFCC for better representing speech and speaker information. The details are described in following subsections.

B. The phonetic bottleneck feature

The phonetic bottleneck feature (PBN-feature) [7,13-14] refers to the feature extracted from a DNN trained for phone classification with a bottleneck layer, where the term bottleneck implies one of the hidden layers is designed to have relatively small number of hidden units (denoted as N_{bn} in this paper) compared to others. Preliminary research [13] conducted on speaker identification (SID) shows that SID with PBN-features outperforms that with MFCCs under UBM/i-vector framework. The experimental results in [14] verified the hypothesis that the PBN-features provide information to the UBM during unsupervised clustering, which enables the UBM align better with phonetic units compared to that purely based on acoustic features. Essentially, PBN-feature is a low-dimension feature vector formed by the outputs of a bottleneck layer, which is a possible better representation since phone discriminative

information learnt by a PBNN has been included. An example of PBNN is shown in Figure 2. To our knowledge, the effectiveness of PBN-features for i-vector SV with short voice commands task is still not clear. One of our research motivation in this study is to investigate the effectiveness of PBN-features for i-vector SV with short voice commands through intensive experiments.

C. The stacked phonetic bottleneck feature

It is noted that PBN-features yield a compact representation of phonetic related information for each frame independently and long-term context is still ignored. Therefore, in order to fully explore the contextual information in consecutive frames, we consider to extract the stacked bottleneck features (SBN-feature) as well which has been successfully used in language identification [11] and speech synthesis [15]. Studies [15] showed that the SBN-feature is able to provide a wide context around the current frame by stacking the PBN-features of multiple consecutive frames. In principle, SBN-feature is a cascaded PBN-feature extracted from a cascaded PBNN (two PBNNs are used in this study) and SBN-feature is frame-by-frame dependent and contains additional contextual information compared to PBN-feature.

The process of SBN-feature extraction is shown in Figure 3. It can be seen that from Figure 3, there are two PBNNs are cascaded (can be more than two PBNNs) and the SBN-feature is obtained at the outputs of the second PBNN bottleneck layer. More specifically, the first PBNN is designed to generate a stacked PBN-feature by taking shifted multi-frames as input. The frame shift size can be chosen and is indicated by N_w . The second PBNN just takes the stacked PBN-feature as input and to generate the SBN-feature. It is noted that, since the SBN-feature is generated by using multi-frames, SBN-feature is expected to hold much more contextual information than PBN-feature at the price of increasing computational cost. Luckily, since the PBNNs are used, the dimension of the

Table 1. Equal Error Rate (EER(%)) and minDCF₀₈ Performance Comparison (the female evaluation set of RSR2015 partII for three types of non-target trials).

Systems	Feature (subscript:dimensionality)	TW		IW		IC	
		EER(%)	minDCF ₀₈	EER(%)	minDCF ₀₈	EER(%)	minDCF ₀₈
i-vector	MFCC ₆₀	1.749	1.0481	0.295	0.1368	4.560	1.9690
PBNF-i-vector	PBN ₈₀	1.51	0.9819	0.261	0.1191	4.917	2.1391
SBNF-i-vector	SBN ₈₀	0.886	0.6107	0.182	0.1026	7.312	3.2779
(PBNF+MFCC)-i-vector	PBN ₈₀ +MFCC ₆₀	0.704	0.4598	0.0625	0.0372	3.418	1.5659
(SBNF+MFCC)-i-vector	SBN ₈₀ +MFCC ₆₀	0.352	0.2140	0.0341	0.0196	3.384	1.5630

PBN-features is low and the price paying for obtaining SBN-feature with several frames are affordable.

IV. PERFORMANCE EVALUATION

In this subsection, we evaluate and compare the performance of five SV with short voice commands systems developed under i-vector framework which include the baseline i-vector, PBNF-i-vector, SBNF-i-vector, (PBNF+MFCC)-i-vector and (SBNF+MFCC)-i-vector speaker verification system with short voice command. A benchmark dataset RSR2015 [1] is used.

A. Database and Experiment settings

Regarding to the RSR2015 dataset, the details are as follows: RSR2015 are uttered by 300 speakers (157 male/143 female) under 9 different sessions. RSR2015 part I contains 30 phonetically-balanced sentences. RSR2015 part II contains 30 short commands while RSR2015 part III has 13 random digit trials. In this study, to increase the training dataset, a new dev dataset (100 male and 94 female speakers) is created by merging the bkg and dev dataset of RSR2015. The new bkg dataset combines all 73 utterances of 194 speakers. A gender-independent UBM model with 512 components is trained on the new bkg dataset, which is initially trained for 4 iterations of EM using a diagonal covariance matrix and then for an additional 4 iterations with a full-covariance matrix. A 400-dim i-vector model is trained on the new bkg dataset for 5 iterations of EM. A gender-dependent PLDA is trained on dev dataset of RSR2015 part II. The MFCC-feature in this study refers to the 60-dimension feature vector extracted from 25 ms of speech frames with 10 ms sliding window, consisting of 19 MFCCs, log-energy and the first and second derivatives MFCCs. An energy-based VAD is used to eliminate non-speech frames. Evaluation is performed using the female evaluation set of RSR2015 (49 female) defined in [1]. We followed the protocol and performance measure in terms of Equal Error Rate (EER) and the minimum detection cost function (minDCF₀₈) defined in [1]. Our i-vector SV-SVC systems have been implemented using the toolkit Kaldi [16].

B. Experiments and Results

The training of the PBNN

As discussed in Section 3.2, to obtain PBN-feature, a PBNN is built and trained. This PBNN is trained by using 100-hours clean speech data from Librispeech [17]. In our study, a 6-layer PBNN is configured as shown in Figure 2. The input of the PBNN is 440-dim feature vector, consisting of 11 frames (5 frame on each side of current frame) where 40 log mel-filterbank coefficients are extracted from each frame. For the BN layer, it is located at the third hidden layer with 80 neurons. For other three hidden layers, each has 1500 neurons. The output layer has 3393 units corresponding to triphone states. The initialization of the PBNN, the fine-tuning, and the learning rate are set by following the settings used in [18].

Feature extraction

- 1) Extraction of PBN-feature: With the trained PBNN, the mel-filterbank features of enrollment and testing set are passed through the PBNN. The output of bottleneck layer is extracted as the phonetic bottleneck feature (PBN-feature).
- 2) The cascade feature of PBN-feature and MFCC feature (PBNF+MFCC): we form a new 140-dim feature vector by appending 60-dim MFCC feature to the 80-dim PBN-feature, which is termed as (PBNF+MFCC) in this context.
- 3) Extraction of SBN-feature: Two PBNNs are stacked as shown in Figure 3. The configuration for the first PBNN is given in section 3.2. The configuration for the second PBNN is same as first PBNN except for the input feature. The bottleneck outputs from the first PBNN are sampled at times $t-5$, t and $t+5$ where the t is the index of the current frame. The contextual window size N_w is set to 3 with fixed step size 5 (since 5 frames are shifted in the first PBNN). By stacking the selected frames, the input feature to the second PBNN is $80 \times 5 = 400$. The 80-dim bottleneck output from the second PBNN is termed as SBN-feature.
- 4) The cascade feature of SBN-feature and MFCC feature (SBNF+MFCC): Similarly, instead of using PBN-feature, we form a new 140-dim feature vector by appending 60-dim MFCC feature to the 80-dim SBN-feature (SBNF+MFCC).

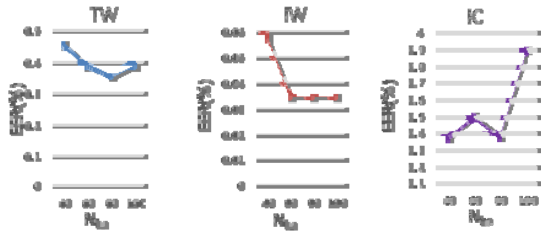


Figure 4: EER (%) of (SBNF+MFCC)-ivector versus N_{bn}

Experiment 1: Performance comparisons

First of all, this experiment is conducted to evaluate the performance of the five i-vector SV-SVC systems. To make it clearly, let's make the following definitions for different SV with short voice commands systems evaluated. The "i-vector" refers to the baseline i-vector SV with short voice commands system by using 60-dim MFCC feature as input of the system. The PBNF-ivector, SBNF-ivector, (PBNF+MFCC)-ivector and (SBNF+MFCC)-ivector indicate four i-vector SV with short voice commands systems using PBN-feature, SBN-feature, (PBNF+MFCC) feature and (SBNF+MFCC) feature, respectively. Experimental settings are given in Section 4.1. The experimental results are given in Table 1. From Table 1, we can see that, for *TW* and *IW* trails, PBN-feature and SBN-feature do helps to improve the performance of SV with short voice commands systems and (SBNF+MFCC)-ivector outperforms other four systems. These results confirm that the adverse impact of content-mismatch can be eliminated by PBN-feature or SBN-feature which provides phonetic discriminative capability. However, for *IC* trials, SBNF-ivector works worst among all. To our understanding, PBN-feature or SBN-feature are not necessarily speaker discriminative and there is a possibility that the speaker-related information is lost to some extent in the training of PBNs. Therefore, it is not surprising that the performance of SBNF-ivector is inferior to that of i-vector and PBNF-ivector. It is encouraged to see that (SBNF+MFCC)-ivector obtains the best result for *IC* trail. Overall, (SBNF+MFCC)-ivector performs best under the experimental conditions. Based on the results in Table 1, we may conclude that MFCC and SBN-feature have good complementary property for SV with short voice commands task.

Experiment 2: (SBNF+MFCC)-ivector performance versus number of bottleneck neurons

As shown in Table 1, (SBNF+MFCC)-ivector achieves the best results under experimental conditions. In this experiment we investigate the impact of N_{bn} on the performance of the (SBNF+MFCC)-ivector. Experimental conditions are the same as those in Experiment 1 except N_{bn}

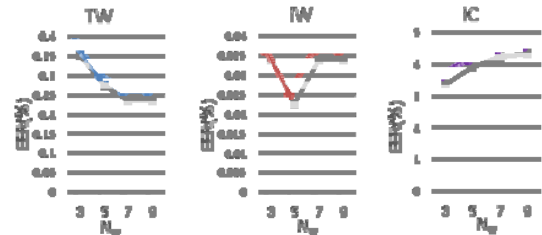


Figure 5: EER (%) of (SBNF+MFCC)-ivector versus N_w

varying from 40 to 100 at the interval of 20. Results are given in Figure 4. From Figure 4, it is clear to see that the system with 80 bottleneck neurons gives the best performance for *TW* and *IW* trials, and gives second best performance for *IC* trial. As the N_{bn} varies from 40 to 100, the standard deviation of EER for *TW*, *IW* and *IC* are 0.037, 0.0098 and 0.29 respectively. This finding shows that N_{bn} has larger impact on *IC* trials but small impact on *TW* trial and much small impact on *IW* trials.

Experiment 3 :Effects of different contextual window size

This experiment evaluates the impact of N_w on the performance of the (SBNF+MFCC)-ivector for three different trails. Here, N_{bn} is 80 for two PBNs and other experimental settings are the same as those in Experiment 1 except N_w varies from 3 to 9 in the step of 2. Figure 5 plots EER results versus different N_w . From Figure 5, we can see that the EER drops as N_w increases for *TW* trial, which indicates SBN-features does help to improve the performance of SV-SVC due to its content mismatch detection ability. However, for *IC* trial, EER increases as N_w increases. This finding suggests that the SBN-features from more frames may lead to loss of the speaker discriminative information. Besides, for *IW* trial, the best result is obtained when N_w equals 5. As the N_w varies from 3 to 9, the standard deviation of EER for *TW*, *IW* and *IC* are 0.046, 0.0049 and 0.37 respectively. This result reflects that N_w has larger impact on *IC* trials, but small impact on *TW* trial and much small impact on *IW* trials.

V. CONCLUSIONS

Text-dependent speaker verification with short voice commands faces technical challenges due to two practical constraints of lexical overlapping and short utterance length. Inspired by the success of bottleneck features in text-dependent speaker verification using fixed phrases (e.g. part I of RSR2015 database), this paper makes an effort to utilize the information provided by the phonetic bottleneck features (PBN-feature) as well as the stacked PBN-feature (SBN-feature) together with MFCC features for i-vector SV with short voice commands systems. Intensive experiments have been conducted to evaluate and compare the performance of our proposed SV with short voice

commands systems for *TW*, *IW* and *IC* non-target trials. On RSR2015 dataset, with a small training dataset, we observe that PBN-features and SBN-features provide content discriminative information and help to improve the performance of SV with short voice commands systems. The cascade feature (SBNF+MFCC) outperforms all other i-vector SV with short voice commands systems since it benefits from the content discriminative and speaker discriminative information. Performances of different stacked phonetic DNNs configurations (namely number of hidden units in bottleneck layer and size of contextual windows) are compared in the (SBNF+MFCC) ivector SV with short voice commands. From our findings, 80 bottleneck neurons would be a good choice. Besides, wider contextual window does not necessarily decrease EER under *IW* and *IC* trials. Future work we will investigate the impact of noise on the EER of (SBNF+MFCC)-ivector for SV with short voice commands tasks.

ACKNOWLEDGEMENT

This work is partially supported by Shenzhen Science & Technology Fundamental Research Programs (No. JCYJ20170306165153653 & JCYJ20150331165212372)

REFERENCES

- [1] 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.
- [2] Jiang Y, Lee K, Tang Z, et al. "PLDA Modeling in I-Vector and Supervector Space for Speaker Verification," in *Proc. InterSpeech*, 2012.
- [3] Matejka, Pavel, et al. "Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification." in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on. IEEE*, 2011, pp. 4828-4831.
- [4] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, "Deep features for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1-13, 2015.
- [5] Alam M J, Kenny P, Gupta V. "Tandem Features for Text-dependent Speaker Verification on the RedDots Corpus," in *Proc. InterSpeech*, 2016, pp. 420-424.
- [6] S. Dey, S. Madikeri, M. Ferras, and P. Motlicek, "deep neural network based posteriors for text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on. IEEE*, March 2016.
- [7] Lozano-Diez, Alicia, et al. "Analysis and optimization of bottleneck features for speaker recognition." *Odyssey 2016: The Speaker and Language Recognition Workshop*. 2016.
- [8] Dey, Subhadeep, et al. "Exploiting sequence information for text-dependent speaker verification,". *No. EPFL-REPORT-225954. Idiap*, 2017.
- [9] Larcher A, Bousquet P, Lee K A, "I-vectors in the context of phonetically-constrained short utterances for speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on. IEEE*, 2012, pp. 4773-4776.
- [10] Wang D S, Zou Y X, Liu J H, Y. C. Huang. "A Robust DBN-vector based Speaker Verification System under Channel Mismatch Conditions," *the 21th IEEE International Conference on Digital Signal Processing (DSP)*, Beijing, China, October 16-18, 2016.
- [11] Matejka P, Zhang L, Ng T, et al. "Neural network bottleneck features for language identification," in *Odyssey 2014: The Speaker and Language Recognition Workshop*. 2014, pp. 299-304.
- [12] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 788-798, 2010.
- [13] M. McLaren, Y. Lei, and L. Ferrer, "Advances in deep neural network approaches to speaker recognition," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on. IEEE*, 2015.
- [14] McLaren M, Ferrer L, Lawson A. "Exploring the role of phonetic bottleneck features for speaker and language recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE*, 2016: 5575-5579.
- [15] Wu Z, Valentinibotinhao C, Watts O, et al. "Deep neural networks employing Multi-Task Learning and stacked bottleneck features for speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on. IEEE*, 2015, pp. 4460-4464.
- [16] Povey D, Ghoshal A, Boulianne G, et al. "The Kaldi speech recognition toolkit," in *Proceedings of the Automatic Speech Recognition and Understanding (ASRU) Workshop*, 2011
- [17] Panayotov V, Chen G, Povey D, et al. "Librispeech: An ASR corpus based on public domain audio books," *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on. IEEE*, 2015, pp : 5206-5210.
- [18] Rath S P, Povey D, Vesely K, et al. "Improved feature processing for deep neural networks," in *Proc. InterSpeech*, 2013, pp. 109-113.