

# Employing Phonetic Information in DNN Speaker Embeddings to Improve Speaker Recognition Performance

*Md Hafizur Rahman*<sup>1\*</sup>, *Ivan Himawan*<sup>2\*</sup>, *Mitchell Mclaren*<sup>3</sup>, *Clinton Fookes*<sup>4</sup>, *Sridha Sridharan*<sup>5</sup>

Speech and Audio Research Laboratory, Queensland University of Technology, Australia<sup>1,2,4,5</sup> Speech Technology and Research Laboratory, SRI International, California, USA<sup>3</sup>

{m20.rahman<sup>1</sup>, i.himawan<sup>2</sup>, c.fookes<sup>4</sup>, s.sridharan<sup>5</sup>}@qut.edu.au, mitchell.mclaren@sri.com<sup>3</sup>

## Abstract

The recent speaker embeddings framework has been shown to provide excellent performance on the task of text-independent speaker recognition. The framework is based on a deep neural network (DNN) trained to directly discriminate between speakers from traditional acoustic features such as Mel frequency cepstral coefficients. Prior studies on speaker recognition have found that phonetic information is valuable in the task of speaker identification, with systems being based on either bottleneck features (BFs) or tied-triphone state posteriors from a DNN trained for the task of speech recognition. In this paper, we analyze the role of phonetic BFs for DNN embeddings and explore methods to enhance the BFs further. Experimental results show that exploiting phonetic information encoded in BFs is very valuable for DNN speaker embeddings. Enriching the BFs using a cascaded DNN multi-task architecture is also shown to provide further improvements to the speaker embedding system.

**Index Terms**: speaker recognition, speaker embeddings network, x-vector, bottleneck features, stacked bottleneck, multitask learning

# 1. Introduction

In recent times, i-vector systems have been very successful as discriminative models to model the speaker and channel variability in the i-vector space [1]. A notable performance improvement has been observed after deep neural networks (DNNs) trained for automatic speech recognition (ASR) were integrated into the speaker verification task. One such method replaced the standard GMM modelling technique for acoustic speech modelling [2, 3, 4]. This technique relies on collecting sufficient Baum-Welch statistics corresponding to the traditional acoustic features from the DNNs and then models the speaker identity into a single low-dimensional i-vector space. An alternative technique incorporates phonetic information in the i-vector system by using BFs from the same DNN and concatenating it with traditional acoustic features [4]. A later study demonstrated that BFs were most valuable in the calculation of the zero-order statistics but tends to harm calibration of the system if it is included in the calculation of the first order statistics [5]

More recently, an end-to-end DNN system for speaker verification has attracted attention, to classify speakers by combining all the steps in traditional DNN/i-vector PLDA systems. These systems have shown very promising results for text-dependent as well as text-independent speaker verification tasks. Different speaker embedding systems have been proposed recently, including "d-vector" for speaker dependent tasks [6], and the "x-vector" system for speaker independent tasks [7, 8, 9]; the latter forms the fundamental system of this study on speaker embeddings. The main objective of these frameworks is to maximize same speaker probability, and to minimize between speaker and domain mismatch probability, thus embedding the speaker discriminant information during the DNN training. Yet to be explored is the role of phonetic information in the context of speaker embeddings, which forms the focus of this study.

This work aims to uncover the role that phonetic information plays in the speaker embeddings framework. This is accomplished by leveraging phonetically-rich BFs as input to the embeddings network. The BFs have been successfully applied in language and speaker identification tasks [4, 10, 11, 12]. The BFs are typically obtained from one of the internal layers of a DNN (with a small number of hidden units in comparison to the size of the other layers) and represent a nonlinear transformation of the input features into a low dimensional representation. As the DNN from which BFs are extracted is trained to be rich in phonetic content and is speaker-independent, one might assume there to be limited speaker information in the BFs. However, we demonstrate that the embeddings network cannot only exploit the speaker information from these features but also leverage the phonetic content to provide a more robust speaker embeddings space. In addition, we also investigate a multi-task learning setup to enhance the BFs for the speaker recognition task. For multi-task experiments, the auxiliary task is to discriminate between speakers in order to force the network to be 'speaker-aware' [13]. Multi-task learning has been very successful in automatic speech recognition and natural language processing tasks before [14, 15]. In this work, we show that enriching the BFs further with speaker discriminative information, improves the performance of DNN based speaker embedding systems as well. We found that a multi-task DNN architecture produces superior BFs compared to a DNN trained only to predict senone posteriors.

This paper is organized as follows: Section 2 details some recent works on speaker embedding systems, Section 3 introduces the role of phonetic information for embedding system training and Section 4 describes the baseline system, where DNN embeddings architecture, i-vector baseline and PLDA backend scorer are described chronologically. The experimental methodology and corresponding results are described in Section 5 and Section 6, respectively. Finally, Section 7 concludes the paper.

#### 2. DNNs in speaker recognition

Recently, Variani *et al.* [6] proposed a DNN-based background modelling approach to directly model the speakers for text dependent task by computing the average of activations from the

<sup>\*</sup>Both authors contributed equally to this paper.

last hidden layer and referred this model as "d-vector". In this approach, the background DNNs are used as speaker specific feature extractors instead of speaker acoustic feature extractor like traditional DNN/i-vector framework. The DNNs are fed with 40-dimensional log filterbank energy features extracted from each feature frame, stacked together with its left and right context frames. The number of outputs of the DNN is equal to the total number of speakers in the development set. The d-vectors are extracted by accumulating output activations of the last hidden layer using standard feedforward propagation in the trained DNN, to represent the speaker model. In the evaluation phase, both target and test d-vectors are extracted and compared using the cosine distance to achieve the desired similarity score. Heigold et al. [16] proposed an end-to-end textdependent speaker verification framework to discriminate between the same and different speaker utterances. The DNNs used in this framework consist of several non-linear hidden layers to transform utterances into the d-vectors. This network uses a locally-connected layer and several fully connected layers using the rectified linear unit (ReLU) activation function [17] and fully connected linear output layer to produce speaker embeddings. The DNN parameters are optimized by minimizing the cross-entropy loss function, and the d-vectors are estimated by averaging the activation of the last hidden layer over all feature frames. Finally, the cosine similarity scoring (CSS) classifier is used to estimate the desired similarity score between the target and test d-vectors.

Li et al. [18] proposed another deep speaker embedding system by investigating two deep architecture including ResNet style [19] deep CNN and the Deep Speech 2 (DS2)-style architecture consisting of convolutional layers followed by GRU layers. In these architectures, instead of pooling layer [7], a temporal average layer is used to convert frame-level activations into speaker representation. They also used a triplet loss layer [20] to minimize the variability between same speaker embeddings and maximize the variability between different speaker embeddings. The triplet loss model uses three utterances, one utterance from the target speaker (anchor), another utterance from the same speaker as a positive example and another utterance from different speaker as negative example. The triplet model is trained in such a way that the cosine similarity between the anchor and the positive example is higher than the cosine similarity between the anchor and the negative example.

Snyder *et al.* [7, 8, 9] introduced another end-to-end approach that can handle variable length of the utterances for the text-independent task. Their proposed architecture uses a feed-forward DNN, which maps the stacked MFCCs fed as input to the network into a speaker embedding. The objective function is used to maximize the speaker probability for the embedding from speakers and minimize the probability for the embedding from different speakers. This network consists of five hidden layers, followed by a temporal pooling layer. This pooling layer estimates the average and standard deviation of the previous hidden layer and pass it to the last hidden layer. The output layer produces linear speaker embedding termed as "x-vectors".

# 3. Role of phonetic information

To the best of our knowledge, the role of phonetic information in the recent speaker embedding framework has not been explored for speaker recognition. It has, however, been leveraged in language embeddings in the related field of language recognition [10, 11]. In this study, we focus on exploiting phonetic information in the embeddings DNN by feeding phonetically



Figure 1: DNN architecture with stacked bottleneck features. The second DNN could either employ multi-task learning for training both primary and secondary tasks, or only the primary task in a single-task learning scenario. The bottleneck features from the second DNN is used as input to the DNN x-vector system.

rich features into the network as opposed to traditional MFCCs. The approaches to producing these phonetically rich features are described in this section.

#### **3.1. Bottleneck features**

Our system for extracting BFs consists of two distinct DNNs, where both DNNs contain bottleneck layer. Figure 1 shows the network architecture with stacked BFs used in this study. We use a DNN that is distributed as a part of the baseline system in the NIST LRE'17 evaluation [21] as the first ASR DNN for BFs extraction. This DNN is trained on speech data from combined switchboard (SWB1) (~319 hours) and Fisher corpora (~2000 hours) corpus with around 8700 senone targets. This model includes seven hidden layers including the bottleneck layer. Each hidden layer except the bottleneck layer uses the ReLU activations.

#### 3.2. Stacked bottleneck features

The stacked BFs (SBNF) are extracted from the second DNN, where we only use the switchboard (SWB1) corpora ( $\sim$ 319 hours) to train the second ASR DNN with the BFs extracted from the first DNN. These features are then used to train the GMM/HMM systems to provide the state alignments for training the second DNN. The feature-space maximum likelihood linear regression (fMLLR) for speaker adaptation is applied to the BN features prior to DNN training. The second DNN includes six hidden layers including the bottleneck layer (80 units) with 9784 senone targets.

#### 3.3. Multi-task learning of Stacked Bottleneck features

For multi-task learning experiments, the second DNN is trained to predict senone posteriors and speakers at the same time. The aim is to train a single DNN architecture to solve in parallel the primary task with additional, closely related tasks



Figure 2: DNN architecture of the speaker embeddings (x-vector) system.

that will improve the generalization of the model. This is achieved through a shared representation (i.e., weight sharing). In this study, the main task is the senone posteriors estimation (i.e., ASR) and the secondary task focuses on recognizing/classifying speakers. It is hypothesized that the network is able to gain contextual information about the speakers and improve the speaker-dependent representations which are beneficial for speech recognition task [12, 13]. The features extracted from the bottleneck layer of this DNN, termed as multi-task stacked bottleneck features (MT SBNF), are later fed into the embedded system training to enhance the system performance.

#### 4. System description

#### 4.1. Speaker embeddings network

In this work, we use a feedforward end-to-end DNN architecture that embeds the speaker information directly into the DNN architecture to classify speakers from the training data proposed by Snyder *et al.* [7, 8, 9]. Figure 2 shows the structure of this end-to-end DNN architecture. The first 5 layers perform on frame level with small temporal contexts for the training, followed by a statistics pooling layer, which aggregates all frames for utterance level representation and computes mean and standard deviation. The statistics pooling layer is followed by another two hidden layers operating on the utterance level and finally the output softmax layer. However, the last two layers are removed, and speaker embedding features are extracted from the layer 6.

#### 4.2. I-vector systems

Speaker embeddings networks have been shown to outperform the prior state-of-the-art system in a number of scenarios [8, 9]. In this work, we also present a comparison of an i-vector pipeline to our embeddings pipeline which supports the conclusions of these prior studies. The traditional i-vector based speaker verification system was the state-of-the-art technology for a very long period of time. In the i-vector system, instead of modelling speaker and channel variability separately, a low dimensional total-variability space is used to represent both speaker and channel models together. In this paper, we

used three i-vector frameworks as baseline systems. In the first system, an UBM with 2048 components is used to map the 13 dimensional feature-warped MFCCs with  $\Delta$  and  $\Delta\Delta$ coefficients into higher dimensional feature space. Later 500dimensional i-vector extractor is used to reduce its dimension into a low-dimensional subspace defined by the matrix T. In the second framework, an ASR DNN is used instead of UBM for collecting sufficient statistics for i-vector extractor training. A multi-splice TDNN is trained with six hidden layers and splicing configuration. The hidden layers use a p-norm activation function (where p = 2). The input layer takes 39-dimensional MFCC features with five-frame temporal context, and cepstral mean subtraction (CMS) performed over a window of six seconds. Each hidden layer has 350 nodes, the output dimension is 3500, and a softmax output layer computes posteriors for 5,346 senone targets. The force-alignment is applied between state-level transcripts and corresponding speech signals to generate HMM state-alignment labels for DNN training. Finally, in the third framework, 80-dimensional phonetic BFs are extracted from the ASR DNN to train the i-vector extractor [4]. The hidden and softmax layers are removed in this framework, as they are not required for BN feature extraction.

#### 4.3. PLDA backend scoring

In this paper, we use a PLDA backend scorer for both i-vector and embedding systems to compute scores between target and test data. Prior to the PLDA scoring, the dimension of the data (including both i-vectors and embedding) is reduced to 150 using LDA sub-space transformation. Later, length normalization is applied prior to GPLDA modelling. Two hypotheses are tested: whether both sessions are from the same speaker and share the same speaker factor, or that they are from the different speaker and have different speaker factors. The scoring between the target and test data is calculated using the batch likelihood ratio [22]. For a given target sample  $\mathbf{w}_{target}$  and test sample  $\mathbf{w}_{test}$ , the batch likelihood ratio can be calculated as follows,

$$\ln \frac{P(\mathbf{w}_{target}, \mathbf{w}_{test} \mid H_1)}{P(\mathbf{w}_{target} \mid H_0)P(\mathbf{w}_{test} \mid H_0)}$$
(1)

where  $H_1$ : The speakers are same,  $H_0$ : The speakers are different

#### 5. Experimental methodology

The training dataset is derived from both NIST and SWB datasets containing 3,769 speakers, in a total of 5,4450 sessions collected from NIST-2004, 2005, 2006, 2008 SRE and Switchboard I, II phase I, II, III corpora. This training dataset is used for both i-vector and DNN speaker embeddings training. In order to increase the data diversity for DNN speaker embeddings training, we use data augmentation recipe proposed by Snyder *et. al* [9]. This data augmentation is performed via adding different levels of noise collected from MUSAN dataset including babble, music, and noise; and reverberation via convolution with simulated room impulse responses (RIR) [23]. We evaluate our proposed systems on NIST 2010 extended core-core conditions. The performances are measured using equal error rate (EER).

For speaker modelling, 13-dimensional feature-warped MFCCs with appended delta coefficients are extracted from raw speech signal using 25 ms frames with 10 ms frameshift. Later, we apply VAD to remove the silence frames from the MFCC feature stream. An UBM of 2048-mixtures is trained and used

Table 1: Performance comparison of baseline speaker recognition systems, evaluated on NIST-2010 extended *core-core* condition

Approach	NIST-2010 core-core EER
UBM/i-vector	3.05%
DNN/i-vector	2.54%
BNF i-vector	1.98%
x-vector system	2.83%
x-vector system (aug)	1.67%

for Baum-Welch (BW) statistics calculation for total-variability space training and i-vector extraction. Later, 500-dimensional i-vector extractor reduces the dimension of the GMM supervector into a low-dimensional subspace defined by the matrix T. For enhanced BN feature extraction we use two distinct ASR DNNs as described in Section 3.2. Both of the ASR DNN models are trained using Kaldi on a senone set with nearly 8700 targets. Both DNNs are trained with 80 hidden units using the renorm nonlinearity. The first ASR DNN is trained with almost 2300 hours of speech data with proper transcripts collected from both Fisher English and SWB I corpus and the second ASR DNN is trained with only SWB I data which is around 319 hours of speech data. The BN features extracted from the first DNN are used to train the secondary DNN with a bottleneck layer. Later, the DNN embeddings are trained with the BN features extracted from the second DNN. Prior to the PLDA modelling, LDA subspace reduces the dimension of the data, where LDA subspace is trained by selecting most significant 150 eigenvectors from 500 eigenvectors based on highest eigenvalues. In this paper, we use the Kaldi toolkit [24] to train our ASR DNN and UBM based i-vector, as well as "x-vector" system, following the recipe proposed by Snyder et al. [24, 9].

# 6. Results and discussion

Table 1 presents the performance of different baseline speaker recognition systems described in Section 4. Experimental results clearly show that DNN/i-vector system and subsequently BFs based i-vector systems perform significantly better than UBM/i-vector system. Clearly, phonetic BFs play a vital role in speaker modelling leading to superior speaker recognition performance. Although, the deep embedded system shows slightly better performance than UBM/i-vector system, i-vector systems trained with BFs performs 30% better than deep embedded system. However, augmenting data diversity in the deep embedded DNN training significantly improve the speaker embeddings system performance by 41%. These results confirm that augmenting the embedded system training is less sensitive to noise and domain variations compared to the UBM/DNN i-vector systems [9, 25]. As the best performing system, we use this augmented DNN speaker embeddings system for the rest of the experiments in this paper.

Now, Figure 3 shows the performance of the augmented DNN speaker embeddings system trained with phonetically enriched BFs. First, we investigated the effectiveness of the BFs for x-vector system training, and the experimental results clearly indicate that employing BFs gains at least 19.8% compared to the DNN speaker embeddings trained with traditional MFCC features. We also trained the x-vector system using BFs extracted from the ASR DNN trained with the stacked BFs, and this system enhances the speaker recognition performance by



Figure 3: Performance comparison of the augmented speaker embeddings system trained with different phonetic bottleneck features, evaluated on NIST 2010 extended core-core condition.

23.4% over MFCC x-vector system. We also investigated the performance of the multi-task training to enrich the BFs for x-vector system training. Experimental results show that multi-task training attains the best performance by achieving 8.6% performance gain over the single-target stacked BFs system. Clearly, stacked BN system performs significantly better than the other systems. However, multi-task training can boost this performance slightly by adding speaker discriminant information into the model during ASR DNN training.

These results confirm what other studies have found that incorporating phonetic BN features provide consistent improvements in speaker recognition task using the i-vector system and the x-vector system. We observe that system performance improvements can be obtained when these features are enriched further via BFs stacking. In our experiments, the same portion of data used for training the first DNN is used as input for training the second DNN (i.e., SWB1). This suggests that BN features extracted from the ASR DNN can be enriched by incorporating speaker discriminant information during ASR DNN training, which eventually improves the DNN embeddings system for speaker recognition.

## 7. Conclusions

This paper explored the possibilities of improving the speaker embeddings system performance by employing phoneticallyrich bottleneck features for DNN embeddings network training. Experimental results confirmed that phonetically-rich BFs obtained from the ASR DNN provide more robust speaker models compared to the traditional feature-based speaker embeddings. We also investigated a cascaded DNN architecture BF system to enrich the BFs in order to improve the overall system performance. Moreover, this system performance can be further enhanced by enriching the BFs with the speaker discriminant information during multi-task ASR DNN training where the speaker targets were used as an auxiliary task. Experimental results showed that BFs extracted from multi-task DNN provide superior system performance compared to the other systems. Our proposed phonetically-rich embeddings networks performance has been validated with NIST 2010 SRE. Future work will include optimizing the speaker embedding network together with with the ASR DNN to share more phonetically discriminant information directly into the embeddings network.

#### 8. References

- N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans actions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.
- [2] D. Snyder, D. Garcia-Romero, and D. Povey, "Time delay deep neural network-based universal background models for speaker recognition," in *IEEE Workshop on Automatic Speech Recognition* and Understanding (ASRU). IEEE, 2015, pp. 92–97.
- [3] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Study of senonebased deep neural network approaches for spoken language recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 1, pp. 105–116, 2016.
- [4] M. McLaren, Y. Lei, and L. Ferrer, "Advances in deep neural network approaches to speaker recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2015, pp. 4814–4818.
- [5] M. McLaren, D. Castan, L. Ferrer, and A. Lawson, "On the issue of calibration in dnn-based speaker recognition systems." in *INTERSPEECH*, 2016, pp. 1825–1829.
- [6] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint textdependent speaker verification," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 4052–4056.
- [7] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT)*, 2016 IEEE. IEEE, 2016, pp. 165–170.
- [8] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech* 2017, pp. 999–1003, 2017.
- [9] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," *ICASSP, Calgary*, 2018.
- [10] P. Matejka, L. Zhang, T. Ng, H. S. Mallidi, O. Glembek, J. Ma, and B. Zhang, "Neural network bottleneck features for language identification," *Proc. Speaker Odyssey: The Speaker and Language Recognition Workshop*, pp. 299–304, 2014.
- [11] P. Matějka, O. Glembek, O. Novotný, O. Plchot, F. Grézl, L. Burget, and J. H. Cernocký, "Analysis of dnn approaches to speaker identification," in *IEEE International Conference on Acoustics*, *Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5100– 5104.
- [12] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.
- [13] G. Pironkov, S. Dupont, and T. Dutoit, "Speaker-aware long shortterm memory multi-task learning for speech recognition," in *Signal Processing Conference (EUSIPCO), 2016 24th European.* IEEE, 2016, pp. 1911–1915.
- [14] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 6965–6969.
- [15] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.
- [16] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. IEEE, 2016, pp. 5115–5119.
- [17] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international* conference on machine learning (ICML-10), 2010, pp. 807–814.

- [18] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," arXiv preprint arXiv:1705.02304, 2017.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [20] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.
- [21] "NIST 2017 language recognition evaluation plan," Tech. Rep., 2017.
- [22] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in Proc. Speaker Odyssey: The Speaker and Language Recognition Workshop, 2010.
- [23] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Acoustics, Speech and Signal Processing* (*ICASSP*), 2017 IEEE International Conference on. IEEE, 2017, pp. 5220–5224.
- [24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop* on automatic speech recognition and understanding. IEEE Signal Processing Society, 2011.
- [25] M. McLaren, D. Castan, M. K. Nandwana, L. Ferrer, and E. Yilmaz, "How to train your speaker embeddings extractor," in *Proc. Speaker Odyssey: The Speaker and Language Recognition Workshop*, 2018.