Analysis and Optimization of Bottleneck Features for Speaker Recognition

Alicia Lozano-Diez[1], *Anna Silnova*[2], *Pavel Matějka*[2], *Ondřej Glembek*[2], *Oldřich Plchot*[2], *Jan Pešán*[2], *Lukáš Burget*[2], *Joaquin Gonzalez-Rodriguez*[1]

[1]ATVS-Biometric Recognition Group, Universidad Autónoma de Madrid, Madrid, Spain
[2]Brno University of Technology, Speech@FIT group and IT4I Centre of Excellence, Czech Republic
alicia.lozano@uam.es, {isilnova,matejkap,glembek,iplchot,ipesan,burget}@fit.vutbr.cz

Abstract

Recently, Deep Neural Network (DNN) based bottleneck features proved to be very effective in i-vector based speaker recognition. However, the bottleneck feature extraction is usually fully optimized for speech rather than speaker recognition task. In this paper, we explore whether DNNs suboptimal for speech recognition can provide better bottleneck features for speaker recognition. We experiment with different features optimized for speech or speaker recognition as input to the DNN. We also experiment with under-trained DNN, where the training was interrupted before the full convergence of the speech recognition objective. Moreover, we analyze the effect of normalizing the features at the input and/or at the output of bottleneck features extraction to see how it affects the final speaker recognition system performance. We evaluated the systems in the SRE'10, condition 5, female task. Results show that the best configuration of the DNN in terms of phone accuracy does not necessary imply better performance of the final speaker recognition system. Finally, we compare the performance of bottleneck features and the standard MFCC features in i-vector/PLDA speaker recognition system. The best bottleneck features yield up to 37% of relative improvement in terms of EER.

1. Introduction

The Speaker Recognition (speaker detection or speaker verification) task consists of determining whether a specified speaker is speaking in a given utterance. For several years, this task has been successfully addressed using the approach based on the i-vector/PLDA (Probabilistic Linear Discriminant Analysis) framework from a typical parameterization of the speech signal such as MFCCs [1, 2].

Recently, Deep Neural Networks (DNNs) have been introduced in the field of speech processing, providing systems that outperform the state-of-the-art approaches in speech recognition [3, 4], language identification [5] and, also, speaker recognition [6, 7, 8, 9].

In the field of speaker recognition, several approaches based on DNNs have been successfully applied, replacing parts of the i-vector/PLDA framework. Some approaches use a DNN to replace the UBM when computing the sufficient statistics or to compute posterior probabilities in an UBM-GMM scheme; others use a DNN, trained for the Automatic Speech Recognition (ASR) task, with a bottleneck layer as feature extractor. Both have shown impressive gains in performance with respect to the traditional approaches [6, 7, 8, 9].

In this paper, we consider the second approach, exploring whether DNNs trained for ASR but not fully optimized for this task could lead to better bottleneck features for speaker recognition. The hypothesis is that the more the DNN is optimized for ASR, the higher the capability of the network to suppress speaker information should be, which is not what is wanted when the DNN is used to extract bottleneck features to discriminate between speakers.

For this purpose, we compare the performance of bottleneck features extracted from a DNN trained with features optimized for ASR and with MFCCs, which are optimal for speaker recognition. We also study how feature normalization affects the performance of speaker recognition systems based on bottleneck features. In particular, we apply short-term mean and variance normalization (ST-MVN), typically used in speaker recognition [10], to the input of the DNN and/or to the input of the speaker recognition system (on top of the bottleneck features) [11]. Finally, we perform experiments with "under-trained" (UT) networks, i.e. DNNs that have not been fully optimized for the ASR task.

Our results show that a DNN with better performance on the ASR task (in terms of phone accuracy) does not necessarily provide better performing speaker recognition system. Therefore, the main contribution of this paper is the analysis of how suboptimal DNNs for ASR could lead to better bottleneck features for speaker recognition.

We evaluate the performance on the NIST SRE'10, condition 5, female task [12], and compare the results of the speaker recognition systems based on bottleneck features with a baseline i-vector/PLDA system based on MFCCs, showing large improvements in performance.

2. Bottleneck Features for Speaker Recognition

The structure of the speaker recognition system based on bottleneck features used in this paper can be split into two different parts. Firstly, a DNN is trained using some input features in order to discriminate between phonetic states. In our case, the architecture of the DNN consists of an input layer followed by four hidden layers, and a final softmax output layer. One of the hidden layers is designed to be relatively small with respect to the others, which is known as the bottleneck layer. The aim of this layer is to compress the information obtained by the network and be able to represent the information learnt by the previous layers. An example of this structure is shown in Figure 1.

Secondly, the trained DNN is used to extract a new frame-by-frame representation of the input signal by propagating the original features through the DNN and taking the activations of the bottleneck layer. These new feature vectors are used to train a GMM-UBM, from which sufficient statistics are collected and used to train the Total Variability matrix [1]. Finally, the cor-

responding i-vectors are extracted, and compared using PLDA model [13, 2] to obtain speaker verification scores.

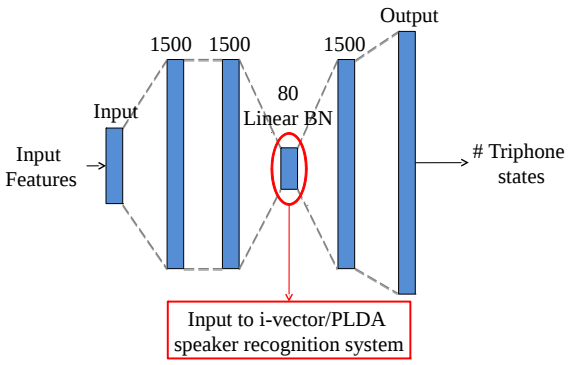

Figure 1: Representation of DNN architecture used in the experiments of this work.

3. Feature Extraction and Normalization

3.1. Input Features

In this work, we used two different sets of input features to feed the DNN: one is optimized for ASR (we will refer to them as "ASR features") meanwhile the other is optimized for speaker recognition (referred to as "MFCC features").

Thus, the experiments tagged as "ASR feat." are those that use the first set of input features optimized for ASR [4]. These feature vectors are composed of 24 Mel-filter bank log outputs concatenated with 13 fundamental frequency (F0) features, resulting in a 37-dimensional vector as described in detail in [14]. Furthermore, utterance mean subtraction is applied on the whole feature vector, which is what we used as default for the ASR task [14].

For the rest of the experiments, tagged as "MFCC", we trained the DNN with the traditional MFCC parameterization used successfully in speaker recognition, either adding the derivatives or not (Δ and $\Delta\Delta$). We used 24 Mel-filter banks to compute these MFCC vectors of 20 coefficients, including c0.

3.2. Short-term Mean and Variance Normalization

The aim of the feature normalization techniques is to compensate the mismatch existent between feature vectors due to environmental effects.

In this work, we consider the normalization strategy known as "short-term mean and variance normalization" (ST-MVN), which was shown to be a simple and fast method to successfully normalize speech segments for the speaker recognition task [10]. This ST-MVN consists of normalizing the mean and variance in a symmetric sliding window as follows:

$$\bar{F}_{i,j} = \frac{F_{i,j} - \mu_{i,j}}{\sigma_{i,j}} \quad (1)$$

where F corresponds to the feature matrix; i and j are the indexes of the frame and the coefficient of the feature vector, respectively; and $\mu_{i,j}$ and $\sigma_{i,j}$ are the mean and standard deviation within the corresponding window. Typically, the window is 3 seconds long (i.e. 150 frames to the left and 150 frames to the right).

This normalization, when applied to cepstral features such as MFCC, is what we also call "floating window cepstral mean and variance normalization" or "short-term cepstral mean and variance normalization" (ST-CMVN).

4. Experimental Framework

4.1. Datasets and Performance Metrics

We use two different datasets in order to train the two parts of the system: the DNN and the i-vector/PLDA system.

We train the DNN using the Fisher English Part 1 and Part 2 datasets. The dataset is composed of approximately 1700 hours of speech. We use 90% of the data for training and the remaining 10% for validation (speakers in these two sets are disjoint). In order to evaluate the performance of the DNN for the task of phoneme classification, we use the frame-by-frame tied-state classification accuracy, which will be referred to as "phone accuracy" for simplicity.

The i-vector/PLDA speaker recognition system is developed using the female portion of the PRISM [15] training dataset, discarding any noise or reverberation data. This set comprises Fisher 1 and 2, Switchboard phase 2 and 3 and Switchboard cellphone phases 1 and 2, along with a set of Mixer data sets. A total number of 9670 speakers is used to train the PLDA models.

Finally, the speaker recognition systems are evaluated on female test data from the NIST SRE'10, condition 5 (telephone condition, normal vocal effort conversational telephone speech in enrollment and test) [12], which includes a total of 236781 trials (3704 targets and 233077 non-targets). The recognition performance is evaluated in terms of the equal error rate (EER, in %) and the minimum detection cost functions (DCF^{min}) as defined in the NIST Speaker Recognition Evaluations 2008 (DCF_{08}^{min}) and 2010 (DCF_{10}^{min}) [16, 12].

4.2. I-vector PLDA Baseline System

The speaker recognition system used as the reference in this work follows the scheme based on i-vectors and PLDA modeling [1, 2], which has been a state-of-the-art approach for the speaker recognition task.

As features for this baseline system, we use a 60-dimensional input vector for each frame, which corresponds to the MFCCs+Δ+$\Delta\Delta$ parameterization. To compute these input vectors, we use the same configuration as described in Section 3.1. Finally, they are normalized according to the ST-MVN described in Section 3.2, using a sliding window of 3 seconds.

With those features, we train a GMM-UBM, collect the sufficient statistics and train the i-vector extractor (total variability matrix), using the data described in Section 4.1. The UBM consists of 512 Gaussian components, and the obtained i-vectors are 400-dimensional vectors. Dimensionality of i-vectors is re-

	EER (%)	DCF_{08}^{min}	DCF_{10}^{min}
Baseline	2.68	0.133	0.517

Table 1: Performance of speaker recognition system based on MFCCs, UBM of 512 Gaussian components, 400-dimensional i-vectors, evaluated on the NIST SRE'10, condition 5, female task.

Features	Input Norm.	Phone Acc.(%)	Raw bottlenecks EER(%)	Raw bottlenecks DCF_{08}^{min}	Raw bottlenecks DCF_{10}^{min}	Normalized bottlenecks (MVN) EER(%)	Normalized bottlenecks (MVN) DCF_{08}^{min}	Normalized bottlenecks (MVN) DCF_{10}^{min}
ASR feat. (EOT)	Utt. CMN	*	2.31	0.105	0.374	2.10	0.093	0.359
ASR feat.	Utt. CMN	**49.8**	2.51	0.100	0.360	2.32	0.103	0.348
$MFCC_{\Delta+\Delta\Delta}$	ST-CMVN	49.6	1.99	0.085	0.328	2.02	0.089	0.337
$MFCC_{20dim}$	ST-CMVN	45.57	**1.67**	**0.079**	**0.312**	**1.91**	**0.088**	**0.325**
$MFCC_{\Delta+\Delta\Delta}$ (UT)	ST-CMVN	47.3	1.72	0.081	0.332	2.06	0.090	0.319
$MFCC_{20dim}$ (UT)	ST-CMVN	42.8	1.68	**0.075**	0.334	**1.78**	**0.083**	**0.317**

Table 2: Performance of speaker recognition systems based on bottleneck features on the NIST SRE'10, condition 5, female task, with an UBM of 512 Gaussian components and 400-dimensional i-vectors. *For this case, the classification accuracy was 49.4%, but for more difficult task of classifying 9824 triphone states compared to 2423 states used for other experiments.

duced to 250 using LDA. Such i-vectors are then transformed by global mean and variance normalization, followed by length-normalization [1, 17].

Finally, the comparison of i-vectors is done via PLDA [2], a generative model that models i-vector distributions allowing for direct evaluation of the desired log-likelihood ratio verification score.

The results of this baseline system can be seen in Table 1. It should be noticed that this system is a scaled down system to allow for fast turnaround of the experiments, but conclusions hold for a large-scale system: UBM of 2048 Gaussian components and 600-dimensional i-vectors (see Table 3).

4.3. DNN Architecture for Bottleneck Extraction

The DNN used in the experimental part of this work follows the structure shown in Figure 1.

For two sets of experiments, we use the two feature sets (ASR and speaker recognition optimized features) as described in Section 3.1. In both cases, the feature vectors are preprocessed as follows: 31 frames are stacked together (central frame $\pm$ 15 frames of context); then, a Hamming window followed by DCT consisting of 0th to 5th bases are applied on the temporal trajectory of each MFCC (or ASR feature) coefficient [14]. The resulting feature vector is used as the input to the DNN.

The DNN consists of four hidden layers with 1500, 1500, 80 and 1500 hidden units, respectively. The 80-dimensional layer is the linear bottleneck layer, while the other three apply the sigmoid function as the activation function. The size of 80 for the bottleneck layer was chosen due to experiments performed in [18], for which 80 provided the best performance.

The DNN has an output layer, which applies a softmax function and consists of 2423 units corresponding to triphone tied-states. These states were obtained from the original triphone state tying obtained during GMM-HMM training. For the experiment shown in the first row of Table 2, an extended output target ("EOT") set considering 9824 triphone states was used.

The cost function that is optimized is the cross-entropy, and the DNN is trained using stochastic gradient descent.

4.4. I-vector PLDA System from Bottleneck Features

The speaker recognition system used for the experiments based on bottleneck features follows the same scheme as the one described in Section 4.2. The only difference is that MFCC features are replaced with the bottleneck features described in Section 4.3. Otherwise the same i-vector/PLDA speaker recognition system is trained on top of the bottleneck features.

5. Experiments and Results

We carried out a set of experiments in order to analyze the influence of different aspects when dealing with the speaker recognition systems based on the bottleneck features as summarized in Table 2.

The first aspect analyzed is the DNN input features, which are either optimized for ASR or speaker recognition ("ASR feat." vs. "MFCC"). Then, feature normalization is also analyzed (see column 2 of Table 2): in the experiments with features optimized for ASR, we applied per utterance mean normalization on top of the input vectors ("Utt. CMN"); while in the experiments using MFCCs, we used the floating window or short-term CMVN ("ST-CMVN"). Finally, for all the experiments, we show results either using "raw bottlenecks" or "normalized bottlenecks", i.e. applying or not applying short-term mean and variance normalization on top of the bottleneck features (right or left hand sides of the table, respectively) [11].

In this section, we comment on both the performance of the DNN as phone classifier and the final speaker recognition systems. The results in terms of speaker recognition performance can be compared to the performance of the baseline system based on MFCCs, which is shown in Table 1.

5.1. Frame Phone Accuracy of the DNN

In third column of Table 2, we can see the phone accuracy obtained for the validation set when training the DNN for the ASR task.

In terms of phone accuracy, we observed a degradation in performance when the derivatives are not included in the input feature vectors ($MFCC_{20dim}$ experiment). However, it should be mentioned that even without the derivatives, the context is taking into account since frames are stacked in the preprocessing of the input to the DNN (see Section 4.3).

Moreover, we see that the ASR features (with per utterance mean normalization) yield better performance in terms of phone accuracy than the MFCCs since they are expected to be optimized for ASR. As we will comment on later, this does not lead to a better performance in the speaker recognition task.

To see whether degradation in phone accuracy was due to the change between ASR features and MFCCs, or to the normalization (utterance CMN applied to ASR features or ST-CMVN applied to MFCCs), we carried out a experiment using ASR features normalized with to ST-CMVN, and in that case, the phone accuracy decreased to 47.56% on the validation set.

Finally, it should be noticed that experiments denoted by "UT" (under-trained) are those in which the training of the network was interrupted even when improvements on validation still existed (i.e. training was stopped few epochs before the

convergence). We did this in order to verify the hypothesis of poor correlation between phone accuracy of the DNN and discriminative power of the resulting bottleneck features for the task of speaker recognition.

5.2. Speaker Recognition Results

5.2.1. ASR Optimized Features

In the experiments based on ASR features as the input to DNN, applying short-term MVN (typically used for features in speaker recognition) on top of the resulting bottleneckfeatures yields a slight improvement in performance ($\sim$10% relative).

However, even though the phone accuracy reaches the highest values with these ASR features, bottleneck features obtained from these DNNs do not seem to be as discriminative as the ones obtained with DNNs trained using MFCCs optimized for the speaker recognition task.

This is also supported by experiment in first row of Table 2. In this experiment, the DNN was trained to classify 9824 triphone states (four times more than in the rest of the experiments), and the phone accuracy was 49.4%. However, the resulting bottleneck features provided similar results that the experiment with the same ASR features, but less triphone states as the DNN outputs.

Even so, these experiments based on bottleneck features outperform the baseline system (see Table 1).

5.2.2. Speaker Recognition Optimized Features

The bottleneck features provided by DNNs trained using MFCC parameterization seem more discriminative for the speaker recognition task.

Using these MFCC features as input to the network, different experiments have been carried out. Opposite to what was observed with the ASR features, when MFCCs with ST-CMVN are used as the input to the DNN, normalizing the resulting bottleneck features did not help or even resulted in slight degradation in performance.

Moreover, in the experiments marked as $\text{MFCC}_{\Delta+\Delta\Delta}$ in the table, we used a 60-dimensional vector of 20 MFCCs with derivatives (Δ and $\Delta\Delta$), while just the 20 MFCCs were used in the experiments denoted by MFCC_{20dim} (all short-term cepstral mean and variance normalized). Comparing this two rows of Table 2, we can see that adding the delta coefficients seems not to increase or even decrease the performance. It should be noticed that even without the derivatives, the context is taken into account due to the staking of frames done at the preprocessing of the input. These 20-dimensional feature vectors got worse phone accuracy but resulted in the best speaker recognition performance, so redundancy introduced by the derivatives helped only in terms of phone discrimination but not in speaker recognition. We see again that better ASR performance (in terms of phone accuracy) does not necessarily correspond to better speaker recognition performance. The hypothesis might be that a DNN optimized for the best discrimination among phoneme states would lead to loosing relevant information for speaker recognition.

Using the best configuration, we see relative improvements up to $\sim$37% in terms of EER with respect to the baseline system.

5.2.3. "Under-trained" DNN Experiments

In order to verify the hypothesis mentioned before, the last two rows of Table 2 show results of DNNs whose training has been stopped before reaching the optimal performance for ASR task (stopped few epochs before the convergence). For those DNNs, results in speaker recognition task give similar or even better performances even though the results did not reach the best values in term of the phone accuracy. Therefore, we see that suboptimal training of DNNs for ASR can result in better feature extractors (DNN with bottleneck layer) for speaker recognition.

	EER (%)	**DCF$_{08}^{min}$**	**DCF$_{10}^{min}$**
BaselineFull	1.99	0.104	0.383
$\text{MFCC}_{\Delta+\Delta\Delta}$	1.62	0.065	0.220
MFCC_{20dim}	**1.46**	**0.057**	**0.209**
BN+$\text{MFCC}_{\Delta+\Delta\Delta}$	**0.96**	**0.042**	**0.146**
BN+MFCC_{20dim}	1.26	0.051	0.216

Table 3: Comparison of performance on the NIST SRE'10, condition 5, female task for large-scale system: UBM of 2048 Gaussian components, 600-dimensional i-vectors.

5.2.4. Full Speaker Recognition System Results and Concatenation of bottleneck features and MFCC

Finally, a comparison in performance between large-scale speaker recognition systems (UBM with 2048 Gaussian components, and 600-dimensional i-vectors) can be seen in Table 3 for the best experiments described above (bottleneck features from MFCC-based DNNs). We see a relative improvement up to $\sim$27% in terms of EER when using bottleneck features from a DNN trained with ST-CMVN MFCCs without derivatives (same DNN as in the experiment shown in the fourth row of Table 2, but with large-scale system).

In the last two rows of Table 3, we also show results using bottleneck features (BN) concatenated with MFCCs (approach that was used in [19]), which provided the best performance (up to $\sim$52% of relative improvement in terms of EER). The bottleneck features used for this concatenation were the ones that provided the best performance in speaker recognition (from a DNN trained with ST-CMVN 20-dimensional MFCCs, row 4 in Table 2).

6. Discussion

According to results in this work, we see that suboptimal DNNs for ASR can provide better bottleneck features for speaker recognition than fully optimized DNNs for the speech recognition task. In order to further analyze that idea, apart from the "under-trained" experiments, we trained a DNN including a new hidden layer (with 1500 hidden units) between the bottleneck layer and the output layer (i.e. having 5 instead of 4 hidden layers). In that experiment, the phone accuracy was higher than in the rest of the experiments, but again, we did not observe any improvement in the speaker recognition performance.

Our hypothesis is that, since bottleneck features are discriminatively trained for phoneme recognition, they should suppress the information about speaker. We believe that the main benefit of using such features is that they lead to more sensible clustering of the acoustic feature space when training GMM-UBM (i.e. GMM components roughly corresponds to phonemes). This is also supported by our experiments using bottleneck features just for alignment of frames to UBM components, while the sufficient statistics for i-vector extraction are collected using MFCCs [18]. Therefore, for a good speaker recognition performance, we need bottleneck features, which

already provide good clustering, but at the same time do not suppress too much of the speaker information.

7. Conclusions

In this work, we studied whether not fully optimized networks trained for ASR could provide better bottleneck features for speaker recognition. Then, we analyzed the influence of different aspects (input features, short-term mean and variance normalization, "under-trained" DNNs) when training DNNs to optimize the performance of speaker recognition systems based on bottleneck features.We evaluated the performance of the resulting bottleneck features in the NIST SRE'10, condition 5, female task.

From the results obtained in this work, we observe that the best features for ASR task do not necessary perform the best when training a network, which is used as feature extractor for speaker recognition. Even though the phone accuracy of the DNN can increase with these features (ASR features), the best performance in speaker recognition was obtained using the typical MFCCs as used for speaker recognition tasks.

According to the results, applying ST-MVN to the MFCCs before training the DNN yields the best performance, and performing that normalization on top of the bottleneck features helps just when input features to the DNN are those optimized for ASR (ASR features with CMN per utterance).

Moreover, the performed experiments do not show much correlation between the frame-by-frame phoneme states classification and the ability of the resulting bottlenecks to discriminate between speakers: the best phone accuracy does not yield the best performance in the speaker recognition task. For example, with just 20 dimensional MFCC feature vectors in which the derivatives have not been added (although context is included when preprocessing the input) we obtained the best results in speaker recognition, while the performance in phone accuracy degrades.

Finally, using bottleneck features from a DNN trained on MFCCs with ST-CMVN, we obtained up to 37% of relative improvement with respect to the baseline system (i-vector based on MFCCs).

Further work will be carried out in order to evaluate these optimized bottlenecks in other conditions and to explore more deeply the concatenation of MFCC and bottlenecks as the input to the speaker recognition system [19]. The hypothesis is that bottleneck features from a ASR network provide good clustering for the UBM training, while MFCCs provide the discriminative information for speaker recognition. Also, stacked bottleneck features used in other works [18] will be explored (they can provide better results although the source of the improvement should be still investigated).

8. Acknowledgments

Thanks to the Speech@FIT group at Brno University of Technology for hosting Alicia Lozano-Diez during her four month research stay in 2015 funded by *Ayuda a la movilidad predoctoral para la realización de estancias breves en centros de I+D, 2014, Ministerio de Economía y Competitividad*, Spain (EEBB-I-15-10381).

This work was supported by project *CMC-V2: Caracterización, Modelado y Compensación de Variabilidad en la Señal de Voz* (TEC2012-37585-C02-01), funded by *Ministerio de Economía y Competitividad*, Spain; and by Czech Ministry of Interior project No. VI20152020025 "DRAPAK", and Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

9. References

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

[2] Patrick Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey 2010: The Speaker and Language Recognition Workshop, Brno, Czech Republic, June 28 - July 1, 2010*, 2010, p. 14.

[3] Geoffrey E. Hinton et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[4] František Grézl, Martin Karafiát, and Lukáš Burget, "Investigation into bottle-neck features for meeting speech recognition," in *Proc. Interspeech 2009*. 2009, number 9, pp. 2947–2950, International Speech Communication Association.

[5] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez-Gonzalez, J. Gonzalez-Rodriguez, and P. J. Moreno, "Automatic language identification using deep neural networks," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014.

[6] Fred Richardson, Douglas A. Reynolds, and Najim Dehak, "A unified deep neural network for speaker and language recognition," *CoRR*, vol. abs/1504.00923, 2015.

[7] Daniel Garcia-Romero and Alan McCree, "Insights into deep neural networks for speaker recognition," in *Proceedings of Interspeech 2015*. 2015, pp. 1141–1145, International Speech Communication Association.

[8] Yao Tian, Meng Cai, Liang He, and Jia Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification," in *Proceedings of Interspeech 2015*. 2015, pp. 1151–1155, International Speech Communication Association.

[9] Mitchell McLaren, Yun Lei, and Luciana Ferrer, "Advances in deep neural network approaches to speaker recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015, pp. 4814–4818.

[10] MdJahangir Alam, Pierre Ouellet, Patrick Kenny, and Douglas OShaughnessy, "Comparative evaluation of feature normalization techniques for speaker verification," in *Advances in Nonlinear Speech Processing*, vol. 7015 of *Lecture Notes in Computer Science*, pp. 246–253. Springer Berlin Heidelberg, 2011.

[11] Luciana Ferrer, Yun Lei, Mitchell McLaren, and Nicolas Scheffer, "Study of senone-based deep neural network approaches for spoken language recognition," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 24, no. 1, pp. 105–116, 2016.

[12] NIST, "The nist year 2010 speaker recognition evaluation plan," www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf, 2010.

[13] Simon J. D. Prince and James H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*, 2007, pp. 1–8.

[14] 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 *Proceedings of Interspeech 2014*. 2014, pp. 3002–3006, International Speech Communication Association.

[15] Luciana Ferrer, Harry Bratt, Lukáš Burget, Jan Černocký, Ondřej Glembek, Martin Graciarena, Aaron Lawson, Yun Lei, Pavel Matějka, Oldřich Plchot, and Nicolas Scheffer, "Promoting robustness for speaker modeling in the community: the prism evaluation set," in *Proceedings of SRE11 Analysis Workshop in 2011*, 2011, pp. 1–7.

[16] NIST, "The nist year 2008 speaker recognition evaluation plan," www.itl.nist.gov/iad/mig/tests/sre/2008/sre08_evalplan_release4.pdf, 2008.

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

[18] Pavel Matějka, Ondřej Glembek, Ondřej Novotný, Oldřich Plchot, František Grézl, Lukáš Burget, and Jan "Honza" Černocký, "Analysis of dnn approaches to speaker identification," *in Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[19] Fred Richardson, Doug Reynolds, and Najim Dehak, "A unified deep neural network for speaker and language recognition," in *Proceedings of Interspeech 2015*. 2015, pp. 1146–1150, International Speech Communication Association.