ANALYSIS OF DNN APPROACHES TO SPEAKER IDENTIFICATION

Pavel Matějka, Ondřej Glembek, Ondřej Novotný, Oldřich Plchot,
František Grézl, Lukáš Burget, and Jan "Honza" Černocký

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic
{matejkap,glembek,inotovny,grezl,burget,iplchot,cernocky}@fit.vutbr.cz

ABSTRACT

This work studies the usage of the Deep Neural Network (DNN) Bottleneck (BN) features together with the traditional MFCC features in the task of i-vector-based speaker recognition. We decouple the sufficient statistics extraction by using separate GMM models for frame alignment, and for statistics normalization and we analyze the usage of BN and MFCC features (and their concatenation) in the two stages. We also show the effect of using full-covariance GMM models, and, as a contrast, we compare the result to the recent DNN-alignment approach. On the NIST SRE2010, telephone condition, we show 60% relative gain over the traditional MFCC baseline for EER (and similar for the NIST DCF metrics), resulting in 0.94% EER.

Index Terms— automatic speaker identification, deep neural networks, bottleneck features, i-vector

1. INTRODUCTION

During the last decade, neural networks have experienced a renaissance as a powerful machine learning tool. Deep Neural Networks (DNN) have been also successfully applied to the field of speech processing. After their great success in automatic speech recognition (ASR) [1], DNNs were also found very useful in other fields of speech processing such as speaker [2, 3, 4] or language recognition [5, 6, 7]. In speech recognition, DNNs are often directly trained for the "target" task of frame-by-frame classification of speech sounds (e.g. phones). Similarly, a DNN directly trained for frame-by-frame classification of languages was successfully used for language recognition in [7]. However, this system provided competitive performance only for speech utterances of short durations.

In the field of speaker and language recognition, DNNs are usually used in more elaborate and indirect way: One approach is to use DNNs for extracting frame-by-frame speech features. Such features are than used in the usual way (e.g. input to i-vector based system [8]). The features can be directly derived from the DNN output posterior probabilities [9, **?**] and combined with the conventional features (PLP or MFCC) [10]. More commonly, however, bottleneck (BN) DNNs are trained for a specific task, where the features are taken from a narrow hidden layer compressing the relevant information into low dimensional feature vectors [6, 5, 11]. Alternatively, standard DNN (with no bottleneck) can be used, where the high-dimensional outputs of one of the hidden layers can be converted to features using a dimensionality reduction technique such as PCA [12].

The DNN for feature extraction needs to be trained first for a specific frame-by-frame classification task. It can be trained for the target task (i.e. classification of languages [7] or speakers [13, 12, 14]), but only limited success was reported with this approach. For both speaker and language recognition, excellent results were observed with features extracted using BN DNN trained for phone classification (i.e. similar to ASR) [11]. It is reasonable to expect such BN features to be suitable for language recognition, as the discrimination between speech sounds is also important for the discrimination between languages. However, it is somewhat counterintuitive to see these features perform well for speaker recognition. The DNN trained for phone classification should have the tendency to suppress the "unimportant" speaker related information.

In the standard i-vector-based approach [8], Gaussian Mixture Model (GMM) is used first to partition the feature space. For each utterance, feature frames are aligned with the GMM components. Using this alignment, sufficient statistics are collected, which are in turn used to extract i-vector as the fixed length low dimensional representation of the utterance. Large improvement in speaker recognition performance was, however, obtained [2, 3, 4, 15] with an alignment given by phone classification DNN as compared to the alignment from the unsupervised GMM. This is another example of successful DNN-based approach to speaker recognition.

The importance of the phone specific frame alignment may also explain the counterintuitive success with the BN features extracted using the DNN trained for phone classification. A GMM trained on such features will likely partition the feature space into phone-like clusters and the alignment obtained using such GMM may closely resemble the one obtained directly from the DNN outputs as proposed in [2, 3]. The appropriate alignment may be more important than the possible loss of speaker related information in the BN features. However, if this assumption is correct, then we may use GMM trained on BN features only to obtain the good frame alignment. Like in [2, 3], the alignment can be used to collect statistic and extract i-vectors using different set of features (e.g. conventional MFCC feature) where we do not risk the loss of speaker

This work was supported by the DARPA RATS Program under Contract No. HR0011-15-C-0038. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

This work was also supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

The work was also supported by Czech Ministry of Interior project No. VI20152020025 "DRAPAK" and European Union's Horizon 2020 programme under grant agreement No. 645523 BISON.

related information. In this paper, we experimentally verify these hypotheses. Note that the approach with the additional set of alignment features has been independently studied in [16], and is further analyzed in [17]. In this paper, however, provides a more thorough analysis and comparison with related approaches such as the DNN based alignment or system based on concatenated BN and MFCC features [11, 18].

2. THEORETICAL BACKGROUND

2.1. i-vector Systems

The i-vectors [8] provide an elegant way of reducing large-dimensional input data to a small-dimensional feature vector while retaining most of the relevant information. The main principle is that the utterance-dependent Gaussian Mixture Model (GMM) supervector of concatenated mean vectors $\mathbf{s}$ is modeled as

$$\mathbf{s} = \mathbf{m} + \mathbf{T}\mathbf{w}, \tag{1}$$

where $\mathbf{m} = [\boldsymbol{\mu}^{(1)'}, \ldots, \boldsymbol{\mu}^{(C)'}]'$ is the Universal Background Model (UBM) GMM mean supervector (of C components), $\mathbf{T} = [\mathbf{T}^{(1)'}, \ldots, \mathbf{T}^{(C)'}]'$ is a low-rank matrix representing M bases spanning subspace with important variability in the mean supervector space, and $\mathbf{w}$ is a latent variable of size M with standard normal distribution.

The i-vector $\boldsymbol{\phi}$ is the Maximum a Posteriori (MAP) point estimate of the variable $\mathbf{w}$. It maps most of the relevant information from a variable-length observation $\mathcal{X}$ to a fixed- (small-) dimensional vector. $\mathbf{L}_{\mathcal{X}}$ is the precision of the posterior distribution.

The closed-form solution for computing the i-vector can be expressed as a function of the *zero-* and *first-order statistics*: $\mathbf{n}_{\mathcal{X}} = [N_{\mathcal{X}}^{(1)}, \ldots, N_{\mathcal{X}}^{(C)}]'$ and $\mathbf{f}_{\mathcal{X}} = [\mathbf{f}_{\mathcal{X}}^{(1)'}, \ldots, \mathbf{f}_{\mathcal{X}}^{(C)'}]'$, where

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

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

where $\gamma_t^{(c)}$ is the posterior (or occupation) probability of frame t being generated by the mixture component c. The tuple $\gamma_t = (\gamma_t^{(1)}, \ldots, \gamma_t^{(C)})$ is usually referred to as *frame alignment*. Note that this variable can be computed either using the GMM UBM or using a completely different model [2, 16, 17]. We will refer to this approach as a *two-model* approach later in this paper. The i-vector is then expressed as

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

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

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

with c being the GMM UBM component index, and the 'bar' symbols denote normalized variables:

$$\bar{\mathbf{f}}_{\mathcal{X}}^{(c)} = \boldsymbol{\Sigma}^{(c)-\frac{1}{2}} \left(\mathbf{f}_{\mathcal{X}}^{(c)} - N_{\mathcal{X}}^{(c)} \boldsymbol{\mu}^{(c)} \right) \tag{6}$$

$$\bar{\mathbf{T}}^{(c)} = \boldsymbol{\Sigma}^{(c)-\frac{1}{2}} \mathbf{T}^{(c)}, \tag{7}$$

where $\boldsymbol{\Sigma}^{(c)-\frac{1}{2}}$ is a symmetrical decomposition (such as Cholesky decomposition) of an inverse of the GMM UBM covariance matrix $\boldsymbol{\Sigma}^{(c)}$.

2.1.1. Two-Model Approach

The true frame alignment is a hidden variable in GMM modeling. Traditionally, it is computed using the GMM UBM. However, it was shown that it can be beneficial to use a different model for computing the frame alignment. As was described in the introduction section, DNNs can be used directly for posterior computation [2]. In this work (as was independently studied in [16] and [17]), a separate alignment GMM is being used for computing the frame posterior.

The natural condition for this approach to work is that the dimensionality of the alignment has to match between the two models (e.g., number of GMM mixtures has to be equal number of DNN posteriors). Also note that the *normalization GMM UBM* (i.e. the $\boldsymbol{\mu}^{(c)}$ and $\boldsymbol{\Sigma}^{(c)}$ parameters) should be computed via the same alignment as used in eq. (2) and (3).

We have decoupled the procedure into two stages—alignment and base statistics extraction and normalization. Let us therefore use the *Alignment–Base* terminology when referencing other relevant blocks, most importantly the feature extraction.

2.2. Stacked Bottleneck Features (SBN)

Bottleneck Neural-Network (BN-NN) refers to such topology of a NN, one of whose 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 BN-NN and reading off the vector of values at the bottleneck layer. We have used a cascade of two such NNs for our experiments. The output of the first network is *stacked* in time, defining context-dependent input features for the second NN, hence the term Stacked Bottleneck Features.

The NN input features are 24 log Mel-scale filter bank outputs augmented with fundamental frequency features from 4 different f_0 estimators (Kaldi, Snack[1], and two according to [19] and [20]). Together, we have 13 f_0 related features, see [21] for more details. The conversation-side based mean subtraction is applied on the whole feature vector. 11 frames of log filter bank outputs and fundamental frequency features are stacked together. Hamming window followed by DCT consisting of 0^{th} to 5^{th} base are applied on the time trajectory of each parameter resulting in $(24+13) \times 6 = 222$ coefficients on the first stage NN input.

The configuration for the first NN is $222 \times D_H \times D_H \times D_{BN} \times D_H \times K$, where K is the number of targets. The dimensionality of the bottleneck layer, D_{BN} was fixed to 80. This was shown as optimal in [6]. The dimensionality of other hidden layers was set to 1500. The bottleneck outputs from the first NN are sampled at times $t-10$, $t-5$, t, $t+5$ and $t+10$, where t is the index of the current frame. The resulting 400-dimensional features are input to the second stage NN with the same topology as first stage. The 80 bottleneck outputs from the second NN (referred as SBN) are taken as features for the conventional GMM/UBM i-vector based SID system.

We experimented with monolingual and multilingual BN features. In the case of multilingual training, we adopted training scheme with block-softmax, which divides the output layer into parts according to individual languages. During training, only the part of the output layer is activated that corresponds to the language that the given target belongs to. Detailed description can be found in [22, 23].

3. EXPERIMENTS

3.1. SBN training data

For training the neural networks, the IARPA Babel Program data[2] were mainly used. This data simulate the case of what one could collect in limited time from a completely new language. It consists mainly of telephone conversational speech, but scripted recordings

[1] http://kaldi.sourceforge.net, www.speech.kth.se/snack/

[2] Collected by Appen, `http://www.appenbutlerhill.com`

Table 1. *Comparison of Multilingual SBN features. We show results for systems using 512 component UBM with diagonal or full (F) covariance matrices. We used 400 dimensional i-vectors in all cases. (The subscript numbers to the right of the feature labels denote their dimensionality.)*

	Alignment Features		Base Features		$\mathrm{DCF}^{\min}_{\mathrm{new}}$	$\mathrm{DCF}^{\min}_{\mathrm{old}}$	EER
1	MFCC	60	MFCC	60	0.518433	0.134960	0.025645
2	BN	80	BN	80	0.290264	0.087904	0.019258
3	BN+MFCC	140	BN+MFCC	140	0.247710	**0.059497**	0.015008
4	BN	80	MFCC	60	0.344090	0.088209	0.019264
5	BN	80	BN+MFCC	140	**0.227594**	0.061843	**0.014236**
6	BN+MFCC	140	MFCC	60	0.327048	0.081493	0.016975
7	F-MFCC	60	F-MFCC	60	0.498147	0.122591	0.023369
8	F-BN	80	F-BN	80	0.268191	0.078103	0.017026
9	F-BN+MFCC	140	F-BN+MFCC	140	**0.231375**	0.061035	0.013422
10	BN	80	F-MFCC	60	0.375041	0.078981	0.015487
11	BN	80	F-BN+MFCC	140	0.256686	**0.059143**	**0.012602**
12	F-BN	80	BN+MFCC	140	0.257967	0.066634	0.015065
13	F-BN	80	F-BN+MFCC	140	0.262086	0.061344	0.013177

Table 2. *Comparison of Multilingual SBN features (both alignment and base are identical). Results for systems using 2048 component UBM. We used 600 dimensional i-vectors. (The subscript numbers to the right of the feature labels denote their dimensionality.)*

	Alignment Features		Base Features		$\mathrm{DCF}^{\min}_{\mathrm{new}}$	$\mathrm{DCF}^{\min}_{\mathrm{old}}$	EER
1	MFCC	60	MFCC	60	0.383004	0.103862	0.019917
2	BN	80	BN	80	0.225030	0.066790	0.016824
3	BN+MFCC	140	BN+MFCC	140	**0.159120**	**0.047686**	**0.010597**
4	F-MFCC	60	F-MFCC	60	0.310579	0.082529	0.015846
5	F-BN	80	F-BN	80	0.201339	0.057185	0.012364
6	F-BN+MFCC	140	F-BN+MFCC	140	0.185950	0.051507	0.012120

as well as far field recordings are present. We used 11 languages to train our multilingual SBN feature extractor. The languages are Cantonese, Pashto, Turkish, Tagalog, Vietnamese, Assamese, Bengali, Haitian, Lao, Tamil, Zulu. More details about the characteristics of the languages can be found in [24]. The phone-state target labels were obtained using forced-alignment with our BABEL ASR system [25].

We also report results for monolingual English SBN variant. We use selection of 250 hours of data derived from Fisher English Part 1 and 2 with 2423 tied triphone states.

3.2. Test Set and Evaluation Metric

NIST SRE 2010 data extended core condition (telephone-telephone) female part was used as the evaluation data. The detection cost function (DCF) is used as a primary evaluation metric. We report two numbers: $\mathrm{DCF}^{\min}_{\mathrm{old}}$ and $\mathrm{DCF}^{\min}_{\mathrm{new}}$ which correspond to the primary evaluation metric for the NIST speaker recognition evaluation in 2008 and 2010 respectively. The difference is that in 2010 NIST focus more on lower false alarm scenario. Third operating point—EER is also reported. For more details see evaluation plans of NIST SRE [3].

3.3. System Description

Voice Activity Detection (VAD) was performed by Neural network with two outputs—speech/non-speech. The NN is trained on Czech CTS data where we artificially added noise with different levels of SNR to 30% of the database. NN has 2 hidden layers with 300 neurons. We used a block of 31 frames of 15 Mel filter bank energies as input features. For the *interview data*, we removed interviewer based on ASR transcripts provided by NIST.

As the baseline features, we used MFCC 19 + energy augmented with their delta and double delta coefficients, making 60 dimensional feature vectors. The analysis window was 20 ms long with shift of 10 ms. First we removed silence frames according to VAD, after which we applied short-time (300 frames) cepstral mean and variance normalization.

The PRISM set [26] was chosen as the base training dataset platform. It contains the following telephone data: NIST SRE 2004, 2005, 2006, 2008, 2010 Switchboard II Phases 2 and 3, Switchboard Cellular Parts 1 and 2, Fisher English Parts 1 and 2 giving 9670 female speakers.

Female gender-dependent UBM was represented as a full or diagonal covariance 512- or 2048-component GMM. It was trained on a subset of PRISM, giving 7815 files equally distributed between telephone and microphone condition. The variance flooring was used in each iteration of EM algorithm during the UBM training.

Female gender-dependent i-vector extractors were trained (in 10 iterations of a joint Expectation Maximization and Minimum Divergence steps) using the entire PRISM set. The results are reported with 400 or 600 dimensional i-vectors.

LDA and PLDA were trained on the same data as the i-vector extractor, except for the Fisher data that was excluded, resulting in 2472 female speakers.

[3] www.itl.nist.gov/iad/mig/tests/sre/

Table 3. *Comparison of SBN features trained on 250 hours from English Fisher. Systems setup: 2048G UBM, 600 dim. i-vectors. (The subscript numbers to the right of the feature labels denote their dimensionality.)*

Alignment Features		Base Features		$\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$	$\mathrm{DCF}_{\mathrm{old}}^{\mathrm{min}}$	EER
MFCC	60	MFCC	60	0.383004	0.103862	0.019917
BN	80	BN	80	0.222194	0.077508	0.020246
DNN(2423)	60	MFCC	60	0.209405	0.055947	0.012080
BN+MFCC	140	BN+MFCC	140	**0.139851**	**0.040596**	**0.009381**

Table 4. *All NIST SRE 2010 conditions for MFCC baseline vs. MFCC+BN. Systems setup: 2048G UBM, 600 dim. i-vectors*

	$\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$		$\mathrm{DCF}_{\mathrm{old}}^{\mathrm{min}}$		EER	
	MFCC	MFCC+BN	MFCC	MFCC+BN	MFCC	MFCC+BN
sre10c01	0.204324	0.159257	0.052725	0.030333	0.011898	0.006436
sre10c02	0.327492	0.249305	0.083562	0.057635	0.018893	0.011745
sre10c03	0.343349	0.229487	0.116152	0.055021	0.023327	0.010755
sre10c04	0.224067	0.156400	0.049074	0.033239	0.009591	0.006862
sre10c05	0.383004	0.159120	0.103862	0.047686	0.019917	0.010597

4. RESULTS

Tab. 1 presents a thorough analysis with scaled-down models (GMM with 512 components and 400 dimensional i-vector), which we built to allow fast turnaround of the experiments. We used multilingual BN features, which proved to be the best in our previous language recognition experiments [27].

First three lines of Tab. 1 present the conventional GMM i-vector system with different feature extractions. The first line shows the baseline results with MFCC coefficients. The results with BN features from the second line show relative improvement 25% in EER and 45% in $\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$ over the baseline. Line 3 presents results for the concatenated MFCC+BN features. Here, the relative improvement over the baseline is 40% in EER, and 50% in $\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$.

Second part of the Tab. 1 presents the results for the two-model approach. The result on line 4 is with MFCC base features and BN alignment features. The relative improvement over the baseline is 25% for EER and 35% for $\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$. This verifies the hypothesis that the BN features provides the frame alignment superior to MFCC features, and with such alignment MFCC can be successfully used as the base features. Nevertheless, the BN features seems to contain enough information relevant for speaker discrimination, which also seem to be complementary to MFCC features: The system based purely on BN features from line 2 still performs slightly better. Furthermore, with BN alignment, an additional improvement is obtained with concatenated MFCC+BN base features as shown in line 5. This combination produce the best results with diagonal covariance matrix yielding 45% in EER and 55% in both DCF points.

The bottom half of Tab. 1 summarizes and compares the results of full-covariance GMM for both alignment and base features. We see 10% relative improvement from full-covariance GMM when the same features are used as alignment and base features. On the other hand, the results are somewhat mixed in the case of the two-model approaches. We usually see about 10% relative degradation at $\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$ point. At the same time, for line 11, where full-covariance matrices are used to normalize statistics for base features (see eq. (6) and (7)), we obtain the best overall results for $\mathrm{DCF}_{\mathrm{old}}^{\mathrm{min}}$ and EER.

Tab. 2 shows the results with the same Multilingual features but bigger system with 2048 components in UBM and 600 dimensional i-vector. There is 18% relative improvement in all conditions when going from small(512/400) to big(2048/600) system. When summarizing the results for the large system, we generally see even bigger gains than for the small system in Tab. 1. Using mere BN single-model system, we see improvement 35% in EER and 56% in $\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$, and when using the concatenated MFCC+BN features-single model, we report relative improvement almost 60% in EER and almost 70% in $\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$ over the baseline results. As with the small system, we see the same behavior for full covariance GMM which calls for further analysis.

Tab. 3 shows results for similar system as Tab. 2 for big system, but the BN features are trained on 250 hours of data selected from Fisher English part 1 and 2. The relative gains over the baseline are lower for BN features, which proves to use Multilingual BN features over the monolingual even if the language is matched (we do not have English language in our Multilingual setup). But the concatenated MFCC+BN features yielded better results than with Multilingual BN. The relative gains over the baseline are 63% in EER which is 0.94% and relative gain 73% in $\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$. This system also beats the DNN alignment approach—alignment posteriors were extracted using the same DNN, except final DNN outputs were used as posteriors, resulting in 2423 output states, i.e. 2423 GMM components [2].

Tab. 4 compares the baseline MFCC system with the concatenated MFCC+BN features (2048-component diagonal GMM) on all NIST SRE 2010 conditions.

5. CONCLUSION

We have analyzed the i-vector based systems with Deep Neural Network (DNN) Bottleneck (BN) features together with the traditional MFCC features, and we have demonstrated substantial gain for NIST SRE 2010, telephone condition. Our best results, with BN trained on Fisher English and BN stacked with baseline MFCC, outperformed the baseline system relatively by 63% at EER and 70% at the $\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$ point. This system also outperformed the DNN alignment approach by 20% relative at ERR and 30% relative at $\mathrm{DCF}_{\mathrm{new}}^{\mathrm{min}}$. We have also analyzed decoupling of the sufficient statistics extraction by using separate GMM models for frame alignment, and for statistics normalization, and we have analyzed the use of BN and MFCC features (and their concatenation) in the two stages. We have also shown the effect of using full-covariance variants of the GMM models.

6. REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath George Dahl, and Brian Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, November 2012.

[2] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *ICASSP*, 2014.

[3] Y. Lei, L. Ferrer, M. McLaren, and N. Scheffer, "Comparative study on the use of senone-based deep neural networks for speaker recognition," *Submitted to IEEE Trans. ASLP*, 2014.

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

[5] Y. Song et al, "i-vector representation based on bottle neck feature for language identification," in *IEEE Electronics Letters*, 2013.

[6] Pavel Matějka et al., "Neural network bottleneck features for language identification," in *IEEE Odyssey: The Speaker and Language Recognition Workshop*, Joensu, Finland, 2014.

[7] Ignacio Lopez-Moreno, Javier Gonzalez-Dominguez, and Oldřich Plchot, "Automatic language identification using deep neural networks," in *ICASSP 2014*, Florence, Italy, 2014.

[8] 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. PP, no. 99, 2010.

[9] M. Diez, A. Varona, M. Penagarikano, L.J. Rodriguez-Fuentes, and G. Bordel, "On the use of phone log-likelihood ratios as features in spoken language recognition," in *SLT*, 2012.

[10] Jeff Ma et al., "Improvements in language identification on the RATS noisy speech corpus," in *Interspeech 2013*, Lyon, France, 2013.

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

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

[13] Takanori Yamada, Longbiao Wang, and Atsuhiko Kai, "Improvement of distant-talking speaker identification using bottleneck features of DNN.," in *INTERSPEECH*. 2013, ISCA.

[14] Yaman S., Pelecanos J., and Sarikaya R., "Bottleneck features for speaker recognition," in *Odyssey*, 2012.

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

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

[17] Sandro Cumani, Olda Plchot, and Pietro Laface, "Comparison of hybrid dnn-gmm architectures for speaker recognition," in *ICASSP*, Submited to ICASSP 2016.

[18] Mitchell McLaren, Martin Graciarena, and Yun Lei, "Advances in deep neural network approaches to speaker recognition," in *ICASSP*, 2015.

[19] Kornel Laskowski and Jens Edlund, "A Snack implementation and Tcl/Tk interface to the fundamental frequency variation spectrum algorithm," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010.

[20] David Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis*, W. B. Kleijn and K. Paliwal, Eds., New York, 1995, Elseviever.

[21] 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*, 2014, pp. 3002–3006.

[22] Karel Veselý, Martin Karafiát, František Grézl, Miloš Janda, and Ekaterina Egorova, "The language-independent bottleneck features," in *Proceedings of IEEE 2012 Workshop on Spoken Language Technology*, 2012, pp. 336–341.

[23] Radek Fér, Pavel Matějka, Frantisěk Grézl, Oldřich Plchot, and Jan Černocký, "Multilingual bottleneck features for language recognition," *Interspeech 2015*, 2015.

[24] M. Harper, "The BABEL program and low resource speech technology," in *ASRU 2013*, Dec 2013.

[25] Martin Karafiát, František Grézl, Mirko Hannemann, Karel Veselý, and Jan "Honza" Černocký, "BUT BABEL System for Spontaneous Cantonese," in *Interspeech 2013*, Lyon, France, 2013, pp. 2589–2593.

[26] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, et al., "Promoting robustness for speaker modeling in the community: the prism evaluation set," *https://code.google.com/p/prism-set/*, 2012.

[27] Radek Fer, Matejka Pavel, Grezl Frantisek, Plchot Oldrich, and Cernocky Jan, "Multilingual bottleneck features for language recognition," in *Interspeech*, 2015.