# I-vector Based Deep Neural Network Acoustic Model Adaptation Using Multilingual Language Resource

*Haihua Xu*[1*], *Wei Rao*[1], *Xiong Xiao*[1], *Hao Huang*[3], *Eng-Siong Chng*[1,2], *Haizhou Li*[1,2,4]

[1]Temasek Laboratories, Nanyang Technological University, Singapore
[2]Computer Engineering School, Nanyang Technological University, Singapore
[3]School of Information Science and Engineering, Xinjiang University, Urumqi, China
[4]Department of Electrical and Computer Engineering, National University of Singapore

## Abstract

I-vector adaptation of DNN-HMM acoustic models has shown clear performance improvement for speech recognition. In this paper, we study this technique on Babel task. we use Swahili as target language (training data of 50 hours) and another 6 languages as multilingual resources to train i-vector extractors respectively. Our study shows that i-vector extractors trained with more multilingual data only produce slightly improved results. Moreover, we compared two i-vectors adaptation methods, 1) concatenate i-vectors with spectral features; 2) predict a bias term adding it to spectral features from i-vectors using a NN. When DNN is trained from scratch, the two methods perform similarly. However, only the second method is appropriate in a cross-lingual transfer learning scenario. We investigate it as well, and results show further word error rate reduction can be gained.

**Index Terms**: I-vector, deep neural network, adaptation, multilingual, speech recognition

## 1. Introduction

Since Deep Neural Network (DNN) became dominant for acoustic modeling [1–4], research on how to adapt DNN based acoustic models is increasingly drawing attention in Automatic Speech Recognition (ASR) community. This is because, although DNN based acoustic models are able to yield significantly improved results compared with conventional GMM-HMM [5], they can still suffer from performance degradation due to training and testing speaker (or environment) mismatches. However, it is not straightforward to conduct DNN adaptation, since no explicit structure in DNN is responsible for modelling speaker (or environment) dependent characteristics. dependent characteristics. Moreover, due to limited adaptation data and large number of adapted parameters in DNN, direct weight update is obviously infeasible. Therefore, some circumvents must be adopted. Right now, there are various DNN adaptation methods that can be broadly divided into three categories [6].

The first category of adaptation work is targeted at model level adaptation referring to that input features are fixed while parameters of network are updated. In [7], Yao et al proposed to do adaptation by updating the bias of the top hidden layer of DNN using test data. In [8], a Speaker Adaptive Training (SAT) recipe was proposed to facilitate DNN based speaker adaptation, in which a speaker-dependent (SD) component is inserted between the bottom hidden layers of the original speaker-independent (SI) DNN (SI-DNN), and it is updated with the speaker-dependent data. More recently, [6] proposed to insert a speaker-dependent layer on top of each activation layer to scale the output of activations, as the connections between those activation and speaker-dependent layers are pairwise, the total parameter number is very limited, and the effectiveness has been demonstrated in unsupervised speaker adaptation work.

The second is mainly aimed at feature level adaptation, though DNN parameter update is also considered when necessary. Perhaps, one of the most relevant examples is fMLLR based DNN adaptation [5], in which GMM-HMM based fMLLR features are taken as input to adapt DNN indirectly. More directly, [9] and [10] successively proposed to train a speaker specific vector as a speaker code, in parallel with regular acoustic features to adapt the DNN with or without a sub-network.

Besides, the third category is using auxiliary features to conduct DNN adaptation [11]. For instance, i-vectors based speaker adaptation under DNN acoustic modeling framework belongs to such a category [4, 12–17]. It is known that i-vectors encapsulate speaker characteristic information [18, 19]. When they are concatenated with the regular acoustic features, both speaker peculiarity and phonetic characteristics are simultaneously learned by the DNN, realizing speaker invariance based ASR. The advantage of the framework lies in its simplicity and only one-pass training and decoding needed. In contrast to the concatenation framework, [20] has recently proposed another i-vector based SAT method. In this method, the speaker-dependent feature, which is output by a trained sub-network taking i-vectors as input, is added to the regular acoustic feature as a bias to adapt the SI-DNN.

In this paper, we improve the effectiveness of i-vector based DNN adaptation using multilingual resource from two perspectives. First, we use multilingual data to train i-vector extractor that includes universal back ground model (UBM), total variability matrix etc [4, 18], to extract robust i-vectors. We use 6 language data as source multilingual data and Swahili [21] as target language data from Babel program[1]. It is found using much more multilingual language data to train i-vector extractors only produces slightly better results. Secondly, we try to adapt the multilingual DNN using i-vector based speaker adaptive training method proposed in [20]. We call this procedure as cross-lingual transfer learning. Specifically, the multilingual DNN is not i-vector adapted, but we are intended to use i-vector to adapt the multilingual DNN when we do cross-lingual transfer learning on the target language. In this scenario, feature

[*] This work is supported by the DSO funded project MAISON DSOCL14045, Singapore

[1] http://www.iarpa.gov/index.php/research-programs/babel.

concatenation based adaptation method such as [4] is not applicable, due to the DNN training in [4] is always conducted from scratch. Briefly, we have a multilingual DNN trained with the method proposed in [23], we first tune it on the target Swahili language, obtaining a language specific DNN, then we perform i-vector based speaker adaptation on such a DNN using the method in [20], and get further performance improvement compared with that without i-vector adaptation employed.

## 2. Data resource description

All experimental data is from NIST OpenKWS15 evaluation program [21]. In OpenKWS15, it contains multilingual data, which is introduced to evaluate the effectiveness of multilingual training for surprise language (Swahili) under low-resource acoustic modeling condition. Specifically it refers to the Very Limited Language Pack (VLLP) case in that only about 3 hours of transcribed data is provided [21]. Except for the VLLP data, NIST also released FLP data as usual. In this paper we use FLP training data as the target data instead. Tables 1 and 2 describe the multilingual and the surprise language Swahili data respectively.

Table 1: Multilingual data description. Note that the “Id” is allotted by Babel program, and “Len. (hrs)” stands for overall hour length of the corresponding language data, while “Ave. Sec./Utt.” stands for average second per utterance.

| Language (Id) | Len. (hrs) | #spker | Ave. Sec./Utt. |
|---|---|---|---|
| Cantonese (101) | 141.3 | 952 | 6.35 |
| Pashto (104) | 78.4 | 959 | 4.03 |
| Turkish (105) | 77.2 | 963 | 3.38 |
| Tagalog (106) | 84.5 | 966 | 3.27 |
| Vietnamese (107) | 87.7 | 954 | 4.01 |
| Tamil (204) | 69.4 | 724 | 3.85 |

Table 2: Swahili (202) data description

| Data set | Len. (hrs) | #spker | Ave. Sec./Utt. |
|---|---|---|---|
| FLP training | 55.4 | 496 | 4.08 |
| *dev* | 10.7 | 120 | 3.55 |

To speed up the turnaround to develop an ASR system with less human intervention, NIST tried not to release manual lexicon during OpenKWS15 evaluation period. This requires each participants will either learn a lexicon, or use grapheme lexicon instead to build an ASR system. We choose to use grapheme lexicon for simplicity, and our vocabulary comes from the FLP transcriptions [21]. We note that NIST released a manual lexicon for the FLP system after evaluation. However word error rate (WER) difference is marginal between the two systems using these two lexicons, indicating Swahili language is very regular in terms of pronunciation rules. Besides, NIST also released varieties of text data to build language models (LMs) [21]. However for simplicity, we do not use the data, and only the FLP transcription is used to build trigram language model for testing instead.

## 3. Multilingual i-vector estimation

### 3.1. Multilingual i-vector extractor training

As mentioned, when target language is limited, the UBM cannot be well estimated, resulting in poorly estimated zero order and centered first-order statistics [4], which are essential for total variability matrix and i-vector estimation. Therefore it is necessary to employ more and diversified speaker data to estimate more robust i-vectors. In Babel program, each language data is limited, there are generally less than 100 hours even for the FLP case as shown in Table 1. However, there are many languages in the overall Babel data sets. Hence it is natural to think about using the multilingual data to train i-vector extractors.

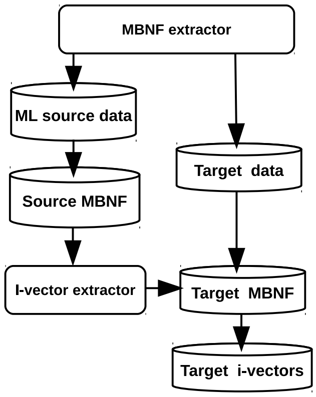

Figure 1: *MBNF based i-vector estimate diagram*

Moreover, which kind of front-end features is used to train i-vector extractor is also investigated as well. In speaker recognition area, [24, 25] successively showed it can yield improved results using BNF as front-end to train i-vector extractor. In ASR area, [20] also got improved results using i-vectors obtained from a similar setup to adapt DNN. In this work, except for using monolingual BNF, we also use MBNF as front-end to train i-vector extractor. To obtain MBNF, We follow the framework proposed in [22] to train MBNF extractor using all 6 multilingual language data from Table 1.

### 3.2. MBNF based i-vector extraction

Once MBNF extractor is ready, we can use it to generate MBNFs for the source (multilingual languages) and target language data respectively. The former is used to train i-vector extractor, and the latter is used to estimate the i-vectors for the target language (Swahili). This procedure is illustrated in Figure 1. Note that Figure 1 also describes the monolingual BNF based i-vector estimate procedure, except that the source and target languages are the same and only a mono-lingual BNF extractor is used instead.

## 4. I-vector based DNN adaptations

We try two different i-vector based DNN adaptation methods. The first is following the recipe proposed in [4]. We call it as feature concatenation based adaptation method. The second follows the recipe proposed in [20], as is depicted in Figure 2, and is a speaker adaptive training (SAT) by estimating feature bias.

### 4.1. Feature concatenation based adaptation

In method proposed in [4], i-vectors are appended to the regular acoustic features to adapt DNN. This method is simple but effective. Since it is not pursued to change the topology of DNN in this scenario (actually there are minor changes in [16]), the main focus is on how to extract robust i-vectors. For instance, [26] assumes the prior of i-vectors is not well described with

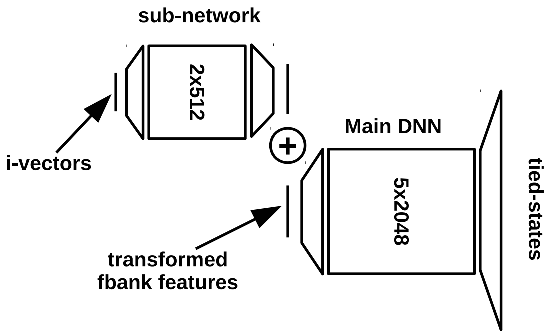

Figure 2: *Illustration of i-vector based DNN acoustic model adaptive training method, using a normalized output by a generic sub-network as a bias to add on the regular acoustic features, where the sub-network is trained by i-vectors, while the main DNN is trained by the cascaded features*

standard normal distribution for shorter utterances, a counting-smooth prior was introduced to estimate i-vectors of shorter utterance. [15] proposed using clean features to collect statistics that are necessary for noisy data i-vector estimate. As a result, parallel features are required.

Apart from robust i-vector estimate, i-vectors can be extracted either on utterance level [12] or on speaker level [4, 13]. In this work, both training and testing i-vectors are extracted at speaker level.

### 4.2. Speaker adaptive training by estimating feature bias

Recently, [20] proposed a framework for DNN based speaker adaptive training using i-vectors as illustrated in Figure 2. From Figure 2, i-vectors are first normalized by a sub-network, whose output is added to the regular acoustic features as a bias, yielding the cascaded features to train the main DNN. The training can be done with two steps. First, once the main DNN is trained (by some kind of acoustic features) and i-vectors of training data are ready, the sub-network can be trained, while the main DNN is kept fixed at this moment. After that, we fix the sub-network and go back to update the main DNN with the cascaded features. For decoding work, once i-vectors are ready, only one-pass decoding is needed.

We notice that such a DNN adaptive training procedure is reminiscent of the conventional GMM-HMM based speaker adaptive training. Besides, [27] justified the motivation of the framework. Actually, the idea is reminiscent of fMPE training [28]. Specifically, if we use $\boldsymbol{y}_t = \boldsymbol{x}_t + net(\boldsymbol{v_t})$ to represent what Figure 2 depicts, it is a kind of “similar” to fMPE transform formula $\boldsymbol{y}_t = \boldsymbol{x}_t + \mathrm{M}\boldsymbol{h}(t)$. Both are aimed to estimate an appropriate bias to solve speaker or environment mismatch issue.

We adopt this framework because the main DNN is not trained from scratch, which makes it possible to keep the knowledge the main DNN initially learned. Moreover, the subsequent i-vector based speaker adaptive training just amounts to a fine-tuning process over the main DNN with cascaded features. This is different from method in [4], since the DNN in [4] is trained from scratch for each adaptive training. Specifically in our case, an initial multilingual DNN[2] is trained from the data in Table 1. At the beginning , we do cross-lingual knowledge transfer learning for Swahili language, yielding a new DNN. Then we use i-vectors to adapt the new DNN, to see if further performance improvement can be obtained. As mentioned earlier, the method in Figure 2 is suitable for our requirement.

## 5. Experimental setup

Experiments are conducted using Kaldi toolkit[3]. We build our ASR system from conventional GMM-HMM to DNN-HMM acoustic models. For GMM-HMM, we train up to SAT GMM-HMM using MLE criterion with 40 *dim* features, which are transformed with LDA plus MLLT over PLP plus pitch features [29], using a 4-1-4 (9) context frame window. For DNN training, input features are filter-bank plus pitch (FBank+pitch) features, and all DNNs have five hidden layers, each with 2048 neurons. Input features are sequentially mean normalized, hamming windowed, and DCT transformed before they are fed into DNN, and they are configured the same with [22] in the case of no i-vectors considered. All frame windows are set 21 with a context 10-1-10. Training criteria for DNN are cross-entropy (CE) and state-level MBR (sMBR) sequence criteria respectively.

For MFCC feature based i-vector extraction, we use 23 *dim* features that have 20 *dim* MFCC plus 3 *dim* pitch features. For BNF based i-vector extraction, the BNFs are 30 *dim*. They are generated by the stacked DNNs [30]. The topology of the stacked DNNs are 1500-1500-80-1500-#tied-states and 1500-1500-30-1500-#tied-states respectively, and also take FBank+pitch features as inputs, but use different frame contexts. See [22] for details. In all cases, i-vector extractors are trained with features including static, plus the first and second delta features respectively. To yield robust i-vector extractor estimate, we remove those silent speech segments using a simple energy based VAD. Throughout experiments, we keep using 2048 mixtures for all UBMs, and the corresponding dimension of i-vectors are fixed with 100.

Some tricks for i-vector extraction are noteworthy. First, during i-vector extractor training process, we treat each utterance as different speaker, so as to accumulate statistics to better model intra-speaker variability. Secondly, we concatenated all utterances from the same speaker when we extract i-vectors, as mentioned in Section 4.1. We found both factors are important to affect performance. Last but not least, all i-vectors are length normalized as advocated in [31], and we have not tried different i-vector dimensions, as [4] and [13] showed it is not a significant factor to affect the performance.

## 6. Results

### 6.1. Monolingual i-vector based DNN adaptation

Table 3 reports our baseline results with two i-vector based DNN adaptation methods, where i-vector extractors are trained with only the target Swahili data.

From Table 3, we see two i-vector based DNN adaptation methods are consistently making performance improvement to different extent with either DNN CE or sMBR sequential training. First of all, Results in Table 3 reveal that adopting BNF features for i-vector extractor training is more effective than MFCC+pitch. Secondly, the “Concat” and the “Bias” adaptation methods are comparable in terms of performance, partic-

[2] The initial multilingual DNN can be trained with or without i-vector adaptation. But if it is the latter case, it would be very time-consuming using the proposed method in [20]. In this paper, the initial DNN is not i-vector adapted.

[3] https://github.com/kaldi-asr/kaldi

Table 3: Results of i-vector based DNN adaptation, with i-vector extractors trained with Swahili FLP data, where "Concat" represents feature concatenation based adaptation method, and "Bias" represents SAT by estimating feature bias method.

| Systems | WER(%) | |
|---|---|---|
| | CE/MLE | sMBR |
| Baseline (without i-vector adaptation) | | |
| GMM-HMM, SAT | 54.8 | - |
| DNN-HMM | 48.5 | 45.6 |
| Monolingual i-vector extractor systems (DNN-HMM) | | |
| MFCC+pitch, "Concat" | 48.4 | 45.2 |
| MFCC+pitch, "Bias" | 47.6 | 45.0 |
| BNF, "Concat" | 47.2 | 44.6 |
| BNF, "Bias" | 47.1 | 44.4 |

ularly in the case of DNN sMBR training. Thirdly, the best absolute WER reductions are 1.4% (2.88% relatively) and 1.2% (2.63%) in both DNN CE and sMBR training cases respectively, when the "Bias" SAT method is employed using i-vectors that are estimated with BNF based i-vector extractor. However, the improvements are moderate compared with that reported from previous work [4, 13]. There are several factors accounting for this situation. One of the main reasons might be due to that training utterances are very short (average length is less than 5 seconds), resulting in poor i-vector estimate, see Table 2. Besides, Babel data is rather challenge, as can be seen from our baseline results in Table 3.

### 6.2. Multilingual i-vector based DNN adaptation

Table 4 compares the performance of two i-vector based DNN adaptation methods by using MFCC+pitch and MBNF features. We note that i-vector extractors in this table are trained with multilingual data in Table 1, and the target Swahili language data is not included in all cases. Actually, we have not gained much improvement when we include Swahili language data to train multilingual i-vector extractors .

Table 4: Results of i-vector based DNN adaptations, where i-vectors are estimated with the extractors that are trained with multilingual data

| Systems (DNN-HMM) | WER(%) | |
|---|---|---|
| | CE | sMBR |
| MFCC+pitch, "Concat" | 47.8 | 45.1 |
| MFCC+pitch, "Bias" | 47.3 | 44.7 |
| MBNF, "Concat" | 47.4 | 43.9 |
| MBNF, "Bias" | 47.4 | 44.8 |

Comparison between Tables 4 and 3 indicates multilingual based i-vector extractors generally yield better results. Especially with MFCC+pitch features, we consistently get better results. However, the best result is from MBNF based "Concat" method, which yields 1.7% absolute WER reduction (3.73% relatively) in the case of DNN sMBR training. Overall, i-vector extractors trained with MBNF yield mixed results. Compared with Table 3, Table 4 sees no improvements in the DNN CE training cases with either method. Besides, it appears that the "Bias" SAT method suffers from over-fitting when i-vector extractor is MBNF trained.

### 6.3. Multilingual DNN based speaker adaptive training

Different from Sections 6.1 and 6.2, in which the target DNNs to be adapted are monolingual DNNs, In this section, we conduct multilingual DNN based adaptation, where only the "Bias" SAT method can be adopted. Table 5 shows our i-vector based multilingual DNN adaptation results, using multilingual DNN as baseline. Specifically, we first have a multilingual DNN trained with the multilingual data in Table 1. See [22] for the details of multilingual DNN training, but notice that we use 6 languages instead of 4 languages in this paper. Then, we do cross-lingual knowledge transfer learning using the target Swahili FLP data, yielding a new DNN, which yields the results as shown in the first row of Table 5. After that, we do i-vector based adaptation training over such a new DNN, using the "Bias" adaptive training method. In Table 5, two kinds of multilingual features, MFCC+pitch and MBNF features, are employed to train i-vector extractors respectively.

Table 5: Results of i-vector based multilingual DNN adaptation, using "Bias" adaptive training method, and multilingual DNN results as baseline

| Systems (DNN-HMM) | WER(%) | |
|---|---|---|
| | CE | sMBR |
| Multilingual (no i-vectors) | 46.6 | 44.1 |
| +multilingual MFCC+pitch | 45.4 | 43.4 |
| +MBNF | 45.3 | 43.3 |

Comparing the first row of Table 5 and the second row of Table 3, we see the DNN multilingual training gets 1.9% and 1.4% absolute WER reductions with the DNN CE and sMBR trainings respectively. Moreover, from Table 5, i-vector adapted DNN systems consistently get moderate performance improvements over the corresponding multilingual DNN systems. The improvements from i-vector based adaptive training indicates though the DNN is trained with a lot of multilingual data, it still needs speaker information (i-vectors) to normalize speaker dependent variations, yielding better results on the target language.

## 7. Conclusion

In this paper, we employed multilingual data resource to improve i-vector based DNN adaptation method, using two different kinds of adaptation frameworks. Improvements come from two aspects. First, we tried to use multilingual data to train i-vector extractors. Since multilingual data contains more diversified speakers, such i-vector extractor yields better i-vector estimate, and hence better DNN adaptation results. Especially, when the i-vector extractor is trained with multilingual BNF, it yields the best results. Secondly, using the speaker adaptive training framework , we also tried i-vector adaptation method in a cross-lingual transfer learning scenario, and got further improved results as well.

# 8. References

[1] D. G. E., D. Yu, L. Deng, and A. Acero., "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[2] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *ICASSP*, 2013.

[3] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proceedings of INTERSPEECH*, 2013.

[4] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-veoctrs," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2013.

[5] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

[6] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *IEEE Workshop on Spoken Language Technology (SLT)*, 2014.

[7] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent neural networks for automatic speech recognition," in *IEEE Workshop on Spoken Language Technology (SLT)*, 2012.

[8] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training using deep neural networks," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2014.

[9] O. Abdel-Humid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2013.

[10] S. Xue, O. Abdel-Hamid, H. Jiang, and L. Dai, "Direct adaptation of hybrid DNN/HMM model for fast speaker adapation in LVCSR based on speaker code," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2014.

[11] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2013.

[12] A. Senior and I. Lopez-Moreno, "Improving DNN speaker independence with i-vector inputs," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2014.

[13] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for French broadcast audio transcription," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2014.

[14] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "Reverberation robust acoustic modelling using i-vectors with time delay neural networks," in *Proceeding of INTERSPEECH*, 2015.

[15] C. Yu, A. Ogawa, M. Delcroix, T. Yoshioka, T. Nakatani, and J. H. L. Hansen, "Robust i-vector extraction for neural network adaptation in noisy environment," in *Proceeding os INTERSPEECH*, 2015.

[16] S. Garimella, A. Mandal, N. Strom, B. Hoffmeister, S. Matsoukas, and S. H. K. Parthasarathi, "Robust i-vector based adaptation of DNN acoustic model for speech recognition," in *Proceeding of INTERSPEECH*, 2015.

[17] P. Cardinal, N. Dehak, Y. Zhang, and J. Glass, "Speaker adaptation using the i-vector technique for bottleneck features," in *Proceeding of INTERSPEECH*, 2015.

[18] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Fron-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[19] S. Ranjan, G. Liu, and J. H. L. Hansen, "An i-vector PLDA based gender identification approach for severely distorted and multilingual DARPA RATS data," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.

[20] Y. Miao, H. Zhang, and F. Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 11, pp. 1938–1949, 2015.

[21] N. F. Chen, V. T. Pham, H. Xu, X. Xiao, V. H. Do, C. Ni, I.-F. Chen, S. Sivadas, C.-H. Lee, E. S. Chng, B. Ma, and H. Li, "Exemplar-inspired strategies for low-resource spoken keyword search in Swahili," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2016.

[22] H. Xu, V. H. Do, X. Xiao, and E.-S. Chng, "A comparative study of BNF and DNN multilingual training on cross-lingual low-resource speech recognition," in *Proceeding of INTERSPEECH*, 2015.

[23] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *ICASSP*, 2013.

[24] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letter*, vol. 22, no. 10, pp. 1671–1675, 2015.

[25] S. H. Ghalehjegh and R. C. Rose, "Deep bottleneck features for i-vector based test-independent speaker verification," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.

[26] P. Karanasou, M. Gales, and P. Woodland, "I-vector estimation using informative priors for adaptation of deep neural networks," in *Proceeding of INTERSPEECH*, 2015.

[27] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *Proceeding of INTERSPEECH*, 2014.

[28] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "Fmpe: Discriminatively trained features for speech recognition," in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2005.

[29] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Jrmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recogniton," in *ICASSP*, 2014.

[30] F. Grézl and M. Karafiát, "Semi-supervised bootstrapping approach for neural network feature extractor training," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2013.

[31] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proceeding os INTERSPEECH*, 2011.