A Unified Deep Neural Network for Speaker and Language Recognition

Fred Richardson[1]*, Doug Reynolds*[1]*, Najim Dehak*[2]

[1]MIT Lincoln Laboratory, Lexington, MA USA
[2]MIT CSAIL, Cambridge, MA USA

frichard@ll.mit.edu, dar@ll.mit.edu, najim@csail.mit.edu

Abstract

Significant performance gains have been reported separately for speaker recognition (SR) and language recognition (LR) tasks using either DNN posteriors of sub-phonetic units or DNN feature representations, but the two techniques have not been compared on the same SR or LR task or across SR and LR tasks using the same DNN. In this work we present the application of a single DNN for both tasks using the 2013 Domain Adaptation Challenge speaker recognition (DAC13) and the NIST 2011 language recognition evaluation (LRE11) benchmarks. Using a single DNN trained on Switchboard data we demonstrate large gains in performance on both benchmarks: a 55% reduction in EER for the DAC13 out-of-domain condition and a 48% reduction in C_{avg} on the LRE11 30s test condition. Score fusion and feature fusion are also investigated as is the performance of the DNN technologies at short durations for SR.

Index Terms: i-vector, DNN, bottleneck features, speaker recognition, language recognition

1. Introduction

The impressive gains in performance obtained using deep neural networks (DNNs) for automatic speech recognition (ASR) [1] have motivated the application of DNNs to other speech technologies such as speaker recognition (SR) and language recognition (LR) [2, 3, 4, 5, 6, 7, 8, 9, 10]. Two general methods of applying DNN's to the SR and LR tasks have been shown to be effective. The first or "direct" method uses a DNN trained as a classifier for the intended recognition task directly to discriminate between speakers for SR [5] or languages for LR [4]. The second or "indirect" method uses a DNN trained for a different purpose to extract data that is then used to train a secondary classifier for the intended recognition task. Applications of the indirect method have used a DNN trained for ASR to extract frame-level features [2, 3, 11], accumulate a multinomial vector [7] or accumulate multi-modal statistics [6, 8] that were then used to train an i-vector system [12, 13].

In this paper we investigate the use of a single DNN for both SR and LR tasks using two indirect methods. The first indirect method (bottleneck features or BNFs) uses frame-level features extracted from a DNN with a special bottleneck layer [14] and the second indirect method (DNN posteriors) uses posteriors extracted from a DNN to accumulate multi-modal statistics [6]. While performance gains have been reported in prior published work using each of these indirect methods separately on different SR and LR tasks, in this work we compare both methods on the same recognition task using the same DNN. In Section 4 we confirm that both indirect methods yield substantial reductions in error rates on SR and LR tasks with the largest gains realized using the BNF method. In Section 4.3 we show that further gains are possible by combining MFCC and BNF features into a tandem feature vector which is compared to fusing scores from multiple systems. In Section 4.4 the tandem features are shown to address a performance issue with the BNF approach for short duration test cuts.

This work is motivated by the need to attain high performance SR and LR under tight storage and computation constraints. The DNN feature extraction is more expensive than MFCC extraction and i-vector extraction also comes at a high cost. The possibility of using a single DNN to extract BNF features for both SR and LR is compelling when the same data is processed for both tasks. While we report some significant gains for system fusion in this work, the cost of extracting multiple i-vectors is high, so it is preferable to attain better performance with fewer i-vector extractions. The tandem systems shows the most promise as a single front end for SR and LR since it doesn't suffer from performance issues on SR at short durations and it sustains the performance gains realized by the BNF system for LR at all durations.

2. DNN's for SR and LR

A DNN classifier is essentially a multi-layer perceptron with more than two hidden layers that typically uses random initialization and stochastic gradient descent to initialize and optimize the weights [1, 15]. For speech applications, the input to a DNN is typically a stacked set of spectral features (e.g., MFCCs, PLPs) extracted from short (20ms) segments (frames) of speech. Typically a context of +/- 5 to 10 frames around the current input frame are used. The output of the DNN is a prediction of the posterior probability of the target classes for the current input frame.

In the direct method for LR and SR, a DNN is used to predict the language or speaker class for a given frame of speech. Since the entire speech waveform is considered to belong to a single class, the frame-level DNN posteriors must be combined to make a single decision score. This can be accomplished either by simply averaging the DNN predictions or by training a secondary classifier that uses statistics derived from the DNN across the whole input as a single feature vector.

In contrast to the direct method, the indirect method uses a DNN that was trained on a different data set and possibly for a different purpose. In this work, we have used a DNN trained for an ASR task for both LR and SR. The ASR DNN is trained to predict sub-phonetic units or "senones" for each input frame [1]. In the following two subsections we describe how we use the ASR DNN output posteriors and BNFs in the context of an i-vector classifier.

This work was sponsored by the Department of Defense under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

2.1. DNN posteriors

A typical i-vector system uses zeroth, first and second order statistics generated using a Gaussian mixture model (GMM) [12]. Statistics are accumulated by first estimating the posterior of each GMM component density for a frame and using these posteriors as weights for accumulating the statistics for each component of the mixture distribution. The zeroth order statistics are the total occupancies across an utterance for each GMM component and the first order statistics are the occupancy weighted accumulations of feature vectors for each component. The i-vector is then computed using a dimension reducing transformation applied to the stacked first order statistics that is non-linear with respect to the zeroth order statistics.

An alternate approach to extracting statistics has been proposed in [6]. Statistics are accumulated in the same way as for the GMM but class posteriors from the DNN are used in place of GMM component posteriors. Once the statistics have been accumulated, the i-vector extraction is performed in the same way as it is from the GMM based posteriors. This approach has been shown to give significant gains for both SR and LR [6, 7, 16].

2.2. DNN bottleneck and tandem features

A DNN can also be used as a means of extracting features for use by a secondary classifier - including another DNN [17]. This is accomplished by sampling the activation of one of the DNN's hidden layers and using this as a feature vector. For some classifiers the dimensionality of the hidden layer is too high and some sort of feature reduction is necessary like LDA or PCA. In [14], a dimension reducing linear transformation is optimized as part of the DNN training by using a special bottleneck hidden layer that has fewer nodes. The bottleneck layer uses a linear activation without an offset and behaves very much like a LDA or PCA transformation on the activation of the previous layer [18, 14]. Matrix factorization was originally proposed in [18] to reduce the number of parameters of the output layer, but in our work we have chosen to use the second to last layer with the hope that the output posterior prediction will not be too adversely affected by the loss of information at the bottleneck layer. BNFs have been shown to work well for both LR and SR [2, 3, 10].

Tandem features consist of BNF features augmented with traditional MFCC or PLP feature. Tandem features provide a way of combining the benefits of both BNF and traditional front-end feature extraction into a single system. They have been shown to perform well for SR [11] but have not been evaluated on LR prior to this work.

One would expect to be able to train a classifier using BNFs that can perform at least as well as the original DNN given the same task and training data. As an example, if the last layer of a DNN is a linear bottleneck, then the final softmax layer is a simple linear classifier that is optimized for the BNFs. A DNN trained to discriminate between senones must attenuate the non-senone related information such as the channel, speaker, gender and sessions information so that the final linear classifier can effectively discriminate between senone classes. BNFs extracted from such a classifier should perform well as phonotatic feature for LR but may not perform as well for SR where speaker related information in the original signal is also needed. However, as shown later in experiments, with long enough utterances BNFs can perform well for SR possibly because we are able to observe enough data to learn a speakers' pronunciation characteristics.

3. Experimental Setup

Three different corpora are used in our experiments. The DNN itself is trained using a 100 hours subset of Switchboard 1 [19]. The 100 hour Switchboard subset is defined in the example system distributed with Kaldi [20]. The SR systems were trained and evaluated using the 2013 Domain Adaptation Challenge (DAC13) data [21]. The LR systems were evaluated on the NIST 2011 Language Recognition Evaluation (LRE11) data [22]. Details on the LR training and development data can be found in [23].

All systems use the same speech activity segmentation generated using a GMM based speech activity detector (GMM SAD). The i-vector system uses MAP and PPCA to estimate the $\mathbf{T}$ matrix. Scoring is performed using PLDA [24]. With the exception of the input features or multi-modal statistics, the i-vector systems are identical and use a 2048 component, diagonal covariance GMM UBM and a 600 dimensional i-vector subspace. All LR systems use the discriminative backend described in [23].

The front-end feature extraction for the baseline LR system uses 7 static cepstra appended with 49 SDC. Unlike the front-end described in [23], vocal track length normalization (VTLN) and feature domain nuisance attribute projection (fNAP) are not used. The front-end for the baseline SR system uses 20 MFCCs including C0 and their first derivatives for a total of 40 features.

The DNN was trained using 4,199 state cluster ("senone") target labels generated using the Kaldi Switchboard 1 "tri4a" example system [20]. The front-end for the DNN uses 13 Gaussianized PLP coefficients and their first and second order derivatives (39 features) stacked over a 21 frame window (10 frames to either side of the center frame) for a total of 819 input features. The GMM SAD segmentation is applied to the stacked features.

The DNN has 7 hidden layers of 1024 nodes each with the exception of the 6^{th} bottleneck layer which has 64 nodes. All hidden layers use a sigmoid activation function with the exception of 6^{th} layer which is linear and has no offset [14]. The DNN training is preformed on an nVidia Tesla K40 GPU using custom software developed at MIT/CSAIL.

4. SR and LR Experiments

In this section we present experiments with the indirect DNN approaches on some well defined SR and LR benchmarks. The SR systems were trained and evaluated using the 2013 Domain Adaptation Challenge (DAC13) [21]. The DAC13 is a specified set of hyper-parameter, enroll, and test lists developed to exhibit a data domain shift for a SR task and has been reported on in several publications [16, 25, 26]. The LR systems were evaluated on the NIST 2011 Language Recognition Evaluation (LRE11) data [22] which covers 24 languages coming from telephone and broadcast audio and has test durations of 3, 10, and 30 seconds. Details on the LR training and development data can be found in [23]. The metrics reported are equal error rate (EER) and minimum decision cost functions (DCF) with a prior or 0.01 for SR and the Cavg (language averaged DCF) with a prior of 0.5 for LR.

4.1. Speaker recognition experiments

Two sets of experiments were run on the DAC13 corpora: "in-domain" and "out-of-domain". For both sets of experiments, the UBM and $\mathbf{T}$ hyper-parameters are trained on Switchboard (SWB) data. The other hyper-parameters (whitening, within, and across covariances) are trained on 2004-2008 speaker

Features	Posteriors	EER(%)	DCF*1000
MFCC	GMM	2.71	0.404
MFCC	DNN	2.27	0.336
BNF	GMM	**2.00**	**0.269**
BNF	DNN	2.79	0.388

Table 1: In-domain DAC13 results

Features	Posteriors	EER(%)	DCF*1000
MFCC	GMM	6.18	0.642
MFCC	DNN	3.27	0.427
BNF	GMM	**2.79**	**0.342**
BNF	DNN	3.97	0.454

Table 2: Out-of-domain DAC13 results

recognition evaluation (SRE) data for the in-domain experiments and SWB data for the out-of-domain experiments (see [21] for more details). The DAC13 test data consists of condition 5 of the NIST 2010 SRE (SRE10) [27]. Tables 4.1 and 4.1 summarize the results for the in-domain and out-of-domain experiments with the first row of each table corresponding to the baseline system. While the DNN-posterior technique with MFCCs gives a significant gain over the baseline system for both sets of experiments, as also reported in [6] and [16], an even greater gain is realized using BNF with a GMM. However, using both BNFs and DNN-posteriors degrades performance.

4.2. Language recognition experiments

The experiments run on the LRE11 task are summarized in Table 4.2 with the first row corresponding to the baseline system and the last row corresponding to a fusion of 5 "post-evaluation" systems (see [23] for details). BNFs with GMM posteriors out performs the other systems configurations including the 5 system fusion. Interestingly, BNFs with DNN-posteriors show more of an improvement over the baseline system than in the speaker recognition experiments.

4.3. Score and feature fusion

Scores from the four speaker recognition systems in Tables 4.1 and 4.1 were fused by combining them with uniform weights. Out of all possible pair-wise combinations, the BNF/GMM+MFCC/DNN systems yielded the best performance. The results are summarized in Table 4.3. For the out-of-domain case the 4 system fusion is actually worse than fusing just the BNF/GMM+MFCC/DNN systems perhaps due to the poorer performance of the MFCC/GMM system in this condition. For the in-domain case the BNF/GMM+MFCC/DNN system fusion comes very close to fusing all four systems. While it is possible that better performance could be attained by estimating the optimal weights for combining scores on held-out data or via cross-validation, we believe that the naive fusion using uniform weights is a good indication of how well fusion works between these different systems. The best in-domain score fusion gives a performance gain of almost 20% relative to the BNF/GMM system alone while the best out-of-domain score fusion gives a relative gain of only about 9%.

Features	Posteriors	30s	10s	3s
SDC	GMM	5.26	10.7	20.9
SDC	DNN	4.00	8.21	19.5
BNF	GMM	**2.76**	**6.55**	**15.9**
BNF	DNN	3.79	7.71	18.2
5-way fusion [23]		3.27	6.67	17.1

Table 3: LRE11 results C_{avg}

Fusion	EER(%)	DCF*1000
BNF/GMM	2.00	0.269
All 4 systems	1.61	0.236
BNF/GMM + MFCC/DNN	1.65	0.237
Tandem/GMM	**1.55**	**0.229**

Table 4: Fusion of all system and the top 2 system on DAC13 in domain task. The system notation used is [feature]/[posterior].

Fusion	EER(%)	DCF*1000
BNF/GMM	2.79	0.342
All 4 systems	2.88	0.355
BNF/GMM + MFCC/DNN	2.54	0.326
Tandem/GMM	**2.44**	**0.323**

Table 5: Fusion of all system and the top 2 system on DAC13 out-of-domain task. The system notation used is [feature]/[posterior].

Also included in Table 4.3 is the result of stacking 20 MFCC features with the 64 BNFs and retraining the GMM i-vector system with the resulting 84 tandem features [28]. The performance for the tandem feature system is slightly better than score fusion for the DAC13 task. The tandem approach may be of interest in limited resource scenarios where it is not possible to run more than one i-vector system.

Score fusion experiments using the four language recognition systems in Table 4.2 were carried out by training a discriminative backend on the development data over all two system combinations and comparing the top performing pair to the fusion of all four systems. As in the DAC13 fusion experiments, the BNF/GMM+MFCC/DNN gave the best performance of all two system combinations. The results are summarized in Table 4.3. While the fusion gains are relatively modest (roughly a 10% relative improvement across the durations), the fusion of just the BNF/GMM+MFCC/DNN is only slight worse than the fusion of all four systems. DET plots for the SR out-of-domain task and LRE11 30s duration of the baseline, DNN, fused and tandem systems are shown in Figures 4.3 and 4.3 respectively.

The tandem system performs worse than score fusion on the LRE11 task but is on par with the BNF/GMM system. This may be because the features used for score fusion and for the tandem features are not the same. The MFCC features used in the tandem system perform well on the SR task but are not as suited to the LR task as the SDC features used in the SDC/DNN system. However, the tandem/GMM system's result suggests that one could use the same tandem feature representation for both LR and SR and still realize a gain on the SR task. This may be of interest in situations where i-vectors are extracted with one set of hyper parameters and then used for both LR and SR.

Fusion	30s	10s	3s
BNF/GMM	2.76	6.55	15.9
All 4 systems	**2.22**	**5.41**	**14.5**
BNF/GMM + SDC/DNN	2.31	5.69	14.7
Tandem/GMM	2.67	6.71	15.9

Table 6: LRE11 fusion C_{avg}

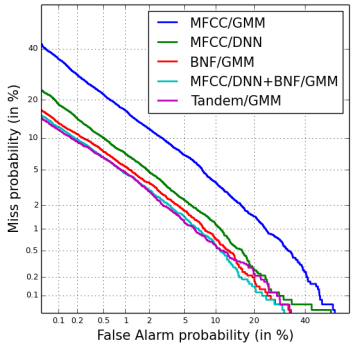

Figure 1: DAC13 out-of-domain DET plot

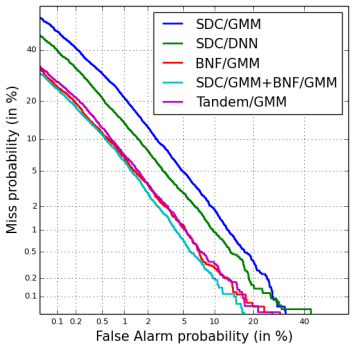

Figure 2: LRE11 30sec duration

4.4. Duration impact on SR performance

The DAC13 test data consists of 5 minute telephone conversation from SRE10 with an average of about 2.5 minutes of speech data per a test cut. Since the BNFs for SR may be sensitive to having enough data to learn a speaker's pronunciation characteristics, we next examined the impact of shorter test durations on SR performance. A test set was created by extracting one 15 second segment from each of the DAC13 test cuts. The set of test trials were the same as for DAC13 except that the models were scored against the 15sec test cuts instead of the full conversation sides. The GMM SAD segmentation was used to ensure that each 15 second segment contained as much speech as possible. The performance and DET curves for the baseline system, both indirect DNN methods, their fusion, and the tandem system are shown in Table 4.4 and Figure 4.4. The fusion of the two DNN systems is significantly better than any of the other systems, but the performance gain comes at the cost of extracting two i-vectors. The performance of the BNF/GMM system is much worse than the baseline MFCC/GMM and the MFCC/DNN systems. One explanation is that the 15 second utterances do not contain enough phonotactic information to discriminate between different speakers effectively. Fortunately the combination of the MFCC and BNF features in the tandem approach appears to compensate for this performance loss using just BNFs.

5. Conclusions

This paper has described the development of a DNN BNF i-vector system and demonstrated substantial performance gains when applying the system to both the DAC13 SR and LRE11 LR benchmarks. For the DAC13 task the BNF/GMM system was shown to reduce the error rates of the baseline

Fusion	EER(%)	DCF*1000
MFCC/GMM	6.07	0.790
MFCC/DNN	6.10	0.783
BNF/GMM	7.72	0.861
BNF/GMM+MFCC/DNN	4.84	0.680
Tandem/GMM	5.70	0.739

Table 7: DAC13 in domain results for the 15 sec task.

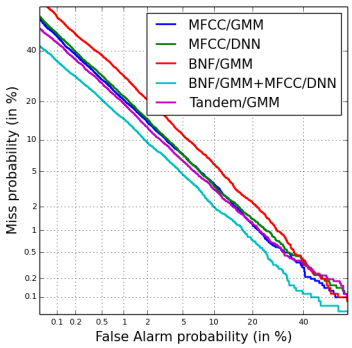

Figure 3: DAC13 in-domain DET plots for 15sec duration cuts

MFCC/GMM system by 26% for EER and 33% for DCF for the in-domain task and 55% for EER and 47% for DCF for the out-of-domain task. On LRE11, the same BNFs decreased EERs at 30s, 10s, and 3s durations by 48%, 39%, and 24%, respectively, and even out performed a 5 system fusion of acoustic and phonetic based recognizers.

Further reductions in error were demonstrated on the DAC13 SR task using score fusion or tandem features. Fusing the BNF/GMM and MFCC/DNN system scores reduces the error rates relative to the BNF/GMM system by 18% for EER and 12% for DCF for the in-domain task and by 9% for EER and 5% for DCF for the out-of-domain task. Using tandem features lead to a larger reduction in error rate of 23% for EER and 15% for DCF for the in-domain task and 13% for EER and 6% for DCF by for the out-of-domain task. Score fusion on the LRE11 task lead to 16%, 13% and 8% reduction in C_{avg} on the 30s, 10s and 3s durations conditions. While the tandem features did not lead to significant changes in performance on the LRE11 task, their good performance on DAC13 along with their apparent robustness at short duration SR suggests the possibility of developing a single tandem front-end and a single i-vector extractor for both SR and LR applications. Future work will investigate the impact of data selection and hyper-parameter training on SR and LR performance when using the same i-vectors for both tasks.

Acknowledgments

The authors would like to thank Patrick Cardinal, Yu Zhang and Ekapol Chuangsuwanich at MIT CSAIL for sharing their DNN expertise and GPU optimized DNN training software.

6. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, pp. 82–97, November 2012.

[2] Y. Song, B. Jiang, Y. Bao, S. Wei, and L.-R. Dai, "I-vector representation based on bottleneck features for language identification," *IEEE Electronics Letters*, pp. 1569–1580, 2013.

[3] P. Matejka, L. Zhang, T. Ng, H. S. Mallidi, O. Glembek, J. Ma, and B. Zhang, "Neural network bottleneck features for language identification," in *Proceedings of IEEE Odyssey*, 2014, pp. 299–304.

[4] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *Proceedings of ICASSP*, 2014, pp. 5374–5378.

[5] T. Yamada, L. Wang, and A. Kai, "Improvement of distant-talking speaker identification using bottleneck features of dnn," in *Proceedings of Interspeech*, 2013, pp. 3661–3664.

[6] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proceedings of ICASSP*, 2014, pp. 1714–1718.

[7] Y. Lei, L. Ferrer, A. Lawson, M. McLaren, and N. Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," in *Proceedings of IEEE Odyssey*, 2014, pp. 287–292.

[8] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Proceedings of IEEE Odyssey*, 2014, pp. 293–298.

[9] O. Ghahabi and J. Hernando, "I-vector modeling with deep belief networks for multi-session speaker recognition," in *Proceedings of IEEE Odyssey*, 2014, pp. 305–310.

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

[11] A. K. Sarkar, C.-T. Do, V.-B. Le, and C. Barras, "Combination of cepstral and phonetically discriminative features for speaker verification," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1040–1044, Sept. 2014.

[12] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front end factor analysis for speaker verification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 19, no. 4, pp. 788–798, may 2011.

[13] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proceedings of Interspeech*, 2011, pp. 857–860.

[14] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *Proceedings of ICASSP*, 2014, pp. 185–189.

[15] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in *Proceedings of ICASSP*, 2013.

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

[17] K. Vesely, M. Karafiat, and F. Grezl, "Convolutive bottleneck network features for lvcsr," in *Proceedings of IEEE ASRU*, 2011, pp. 42–47.

[18] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proceedings of ICASSP*, 2013.

[19] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proceedings of ICASSP*, 1992, pp. 517–520.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J Silovsky, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," in *Proceedings of IEEE ASRU*, 2011.

[21] S. H. Shum, D. A. Reynolds, D. Garcia-Romero, and A. McCree, "Unsupervised clustering approaches for domain adaptation in speaker recognition systems," in *Proceedings of IEEE Odyssey*, 2014, pp. 265–272.

[22] "The 2011 nist language recognition evaluation plan," http://www.nist.gov/itl/iad/mig/lre11.cfm, 2011.

[23] E. Singer, P. Torres-Carrasquillo, D. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. Sturim, "The mitll nist lre 2011 language recognition system," in *Proceedings of IEEE Odyssey*, 2011, pp. 209–215.

[24] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proceedings of Interspeech*, 2011, pp. 249–252.

[25] H. Aronowitz, "Inter dataset variability compensation for speaker recognition," in *Proceedings of ICASSP*, 2014.

[26] O. Glembek, J. Ma, P. Matejka, B. Zhang, O. Plchot, L. Burget, and S. Matsoukas, "Domain adaptation via within-class covariance correction in i-vector based speaker recognition systems," in *Proceedings of ICASSP*, 2014.

[27] "The nist year 2010 speaker recognition evaluation plan," http://www.itl.nist.gov/iad/mig//tests/sre/2010/, 2010.

[28] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional hmm systems," in *Proceedings of ICASSP*, 2000.