Speaker Verification and Spoken Language Identification using a Generalized I-vector Framework with Phonetic Tokenizations and Tandem Features

Ming Li[12], *Wenbo Liu*[1]

[1]SYSU-CMU Joint Institute of Engineering, Sun Yat-Sen University, Guangzhou, China
[2]SYSU-CMU Shunde International Joint Research Institute, Shunde, China

liming46@mail.sysu.edu.cn, wenbobo.liu@gmail.com

Abstract

This paper presents a generalized i-vector framework with phonetic tokenizations and tandem features for speaker verification as well as language identification. First, the tokens for calculating the zero-order statistics is extended from the MFCC trained Gaussian Mixture Models (GMM) components to phonetic phonemes, 3-grams and tandem feature trained GMM components using phoneme posterior probabilities. Second, given the calculated zero-order statistics (posterior probabilities on tokens), the feature used to calculate the first-order statistics is also extended from MFCC to tandem features and is not necessarily the same feature employed by the tokenizer. Third, the zero-order and first-order statistics vectors are then concatenated and represented by the simplified supervised i-vector approach followed by the standard back end modeling methods. We study different system setups with different tokens and features. Finally, selected effective systems are fused at the score level to further improve the performance. Experimental results are reported on the NIST SRE 2010 common condition 5 female part task and the NIST LRE 2007 closed set 30 seconds task for speaker verification and language identification, respectively. The proposed generalized i-vector framework outperforms the i-vector baseline by relatively 45% in terms of equal error rate (EER) and norm minDCF values.

Index Terms: speaker verification, language identification, generalized i-vector, phonetic tokenization, tandem feature

1. Introduction

Total variability i-vector modeling has gained significant attention in both speaker verification (SV) and language identification (LID) domains due to its excellent performance, compact representation and small model size [1, 2, 3]. In this modeling, first, zero-order and first-order Baum-Welch statistics are calculated by projecting the MFCC features on those Gaussian Mixture Model (GMM) components using the occupancy posterior probability. Second, in order to reduce the dimensionality of the concatenated statistics vectors, a single factor analysis is adopt to generate a low dimensional total variability space which jointly models language, speaker and channel variabilities all together [1]. Third, within this i-vector space, variability compensation methods, such as Within-Class Covariance Normalization (WCCN) [4], Linear Discriminative Analysis (LDA) and Nuisance Attribute Projection (NAP) [5], are performed to reduce the variability for the subsequent modeling methods (e.g., Support Vector Machine (SVM), Logistic Regression [3] and Neural Network [6, 7] for LID and Probabilistic Linear Discriminant Analysis (PLDA) [8, 9] for SV, respectively).

Lei, et.al [10] and Kenny, et.al [11] recently proposed a generalized i-vector framework where decision tree senones (tied triphone states) in a general Deep Neural Network based Automatic Speech Recognition (ASR) system are employed as the new type of tokens for statistics calculation, rather than the conventional MFCC trained GMM components. Although the features used to calculate the first-order statistics remain the same (MFCC), the phonetically-aware tokens trained by supervised learning can provide better token separation and more accurate token alignment, which leads to significant performance improvement on SV tasks. Nevertheless, there are several other phonetic units (e.g. phonemes, trigrams, etc.) with larger scale that have the potential to be considered as tokens as well (especially for LID task). The frame level posterior probabilities of these phonetic tokens can also be converted into tandem features followed by the standard GMM to fit the conventional GMM framework.

This motivates us to investigate different alternative configurations of phonetic tokens and features for zero-order and first-order statistics calculation within this generalized framework and apply them to both SV and LID. First, we explore the commonly used phonemes as the phonetic tokens and extend to even larger units such as trigrams. In this way, the bag of trigrams vector in the vector space modeling [12] is exactly the zero-order statistics on these trigrams. Second, since the number of phonemes is much smaller than the number of tied triphone states, we converted the phoneme posterior probabilities into tandem features [13, 14] and then apply GMM on top of it to generate large components tokens. This is also motivated by the hierarchical phoneme posterior probability estimator in [15]. In this setup, the GMM statistics calculation remains the same except that the GMM is trained on the tandem features.

This phoneme posterior probability (PPP) based tandem feature has been reported to be effective in both ASR [13, 14, 16] and LID tasks[17, 18] as front end features. GMM mean supervector modeling and conventional i-vector modeling are used to model this tandem feature in [17] and [18] for LID. In both methods, the tandem feature outperformed the shifted-delta-cepstral (SDC) feature by more than 30% relatively. We note that the conventional i-vector modeling on tandem features (in [18]) is a special case in this generalized i-vector framework where tandem features and the derived GMM components are considered as features and tokens, respectively.

Since the features for extracting tokens and the features for calculating the first-order statistics are not necessary the same [10], we show that in terms of first-order statistics calculation, MFCC is superior than tandem features for SV, and vice versa

This research is funded in part by CMU-SYSU Collaborative Innovation Research Center and the SYSU-CMU Shunde International Joint Research Institute.

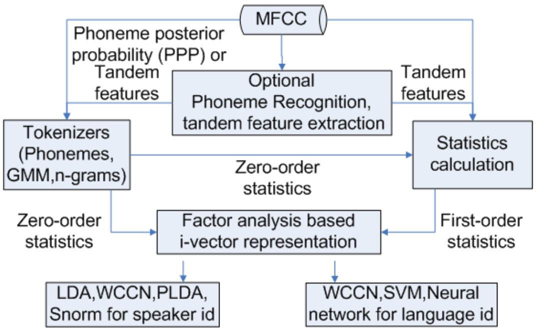
MFCC
Phoneme posterior probability (PPP) or Tandem features
Optional Phoneme Recognition, tandem feature extraction
Tandem features
Tokenizers (Phonemes, GMM,n-grams)
Zero-order statistics
Statistics calculation
Zero-order statistics
Factor analysis based i-vector representation
First-order statistics
LDA,WCCN,PLDA, Snorm for speaker id
WCCN,SVM,Neural network for language id

Figure 1: The generalized i-vector framework

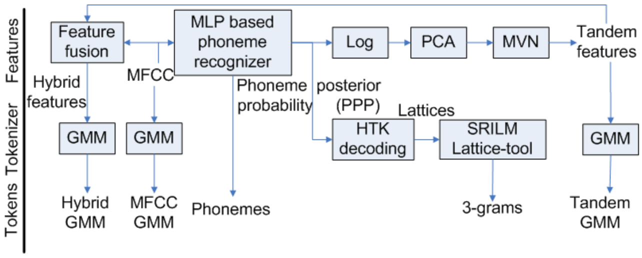

Figure 2: Tokens for zero-order statistics calculation

for LID. We further explore the hybrid features which concatenate the acoustic MFCC and the phonetic tandem features at the frame level for both purposes. This setup not only achieves better performance but also directly fit the conventional i-vector framework.

2. Methods

The overview of the proposed generalized i-vector framework is shown in Fig. 1. Our generalized framework extends the choices of tokens and features for statistics calculation while keeps the factor analysis, variability compensation and subsequent modeling the same way as the conventional i-vector method. Table 1 and fig. 2 demonstrates the five different tokens that we explored in this work as well as the processes to extract them. We first describe the statistics calculation, factor analysis based i-vector baseline and our simplified version simplified supervised i-vector in Sec 2.1. Then statistics calculation with new types of phonetic tokens and tandem features in the generalized i-vector framework is introduced in Sec 2.2.

2.1. I-vector baseline and the simplified supervised i-vector

Given a C component GMM UBM model λ with $\lambda_c = \{p_c, \mu_\mathbf{c}, \mathbf{\Sigma_c}\}, c = 1, \cdots, C$ and an utterance with a L frame feature sequence $\{\mathbf{y_1}, \cdots, \mathbf{y_L}\}$, the zero-order and centered first-order Baum-Welch statistics on the UBM are calculated as follows:

$$N_c = \sum_{t=1}^{L} P(c|\mathbf{y_t}, \lambda) \tag{1}$$

$$\mathbf{F_c} = \sum_{t=1}^{L} P(c|\mathbf{y_t}, \lambda)(\mathbf{y_t} - \mu_\mathbf{c}) \tag{2}$$

where $c = 1, \cdots, C$ is the GMM component index and $P(c|\mathbf{y_t}, \lambda)$ is the occupancy posterior probability for $\mathbf{y_t}$ on λ_c. The corresponding centered mean supervector $\tilde{\mathbf{F}}$ is generated by concatenating all the $\tilde{\mathbf{F}}_\mathbf{c}$ together:

$$\tilde{\mathbf{F}}_\mathbf{c} = \frac{\sum_{t=1}^{L} P(c|\mathbf{y_t}, \lambda)(\mathbf{y_t} - \mu_\mathbf{c})}{\sum_{t=1}^{L} P(c|\mathbf{y_t}, \lambda)}. \tag{3}$$

The centered mean supervector $\tilde{\mathbf{F}}$ can be projected as follows:

$$\tilde{\mathbf{F}} \rightarrow \mathbf{Tx}, \tag{4}$$

where $\mathbf{T}$ is a rectangular total variability matrix of low rank and $\mathbf{x}$ is the so-called i-vector [2]. Considering a C-component GMM and D dimensional acoustic features, the total variability matrix $\mathbf{T}$ is a $CD \times K$ matrix which is estimated the same way as learning the eigenvoice matrix in [19] except that here we consider that every utterance is produced by a new speaker [2].

As shown in fig. 3, we recently proposed the simplified supervised i-vector method [20, 21] which achieves comparable performance to the conversion i-vector baseline and at the same time reduces the computational cost by a factor of 100. Since this method relies on the same set of statistics and is more efficient, it is employed as the factor analysis based dimensionality reduction method for all the experiments in this work.

Table 1: The proposed methods with different combinations of tokens and features for zero-order and first-order statistics calculation (here phonemes refer to the monophone states)

Methods	Tokens	Feature for first order statistics
Baseline	MFCC GMM	MFCC
Phonemes-MFCC	Phonemes	MFCC
Tandem-GMM-MFCC	Tandem-GMM	MFCC
Trigrams-MFCC	Trigrams	MFCC
Tandem-GMM-Tandem	Tandem-GMM	Tandem
Trigrams-Tandem	Trigrams	Tandem
Hybrid-GMM-Hybrid	Hybrid-GMM	MFCC+Tandem

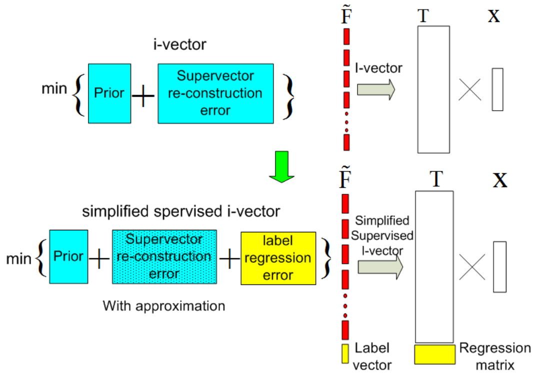
i-vector
min
Prior
Supervector re-construction error
F̃
I-vector
T
x
simplified spervised i-vector
min
Prior
Supervector re-construction error
label regression error
With approximation
F̃
Simplified Supervised I-vector
T
X
Label vector
Regression matrix

Figure 3: Schematic of the factor analysis based i-vector and simplified supervised i-vector modeling [20, 21]

2.2. Statistics calculation in the generalized framework

In our generalized i-vector framework, the zero-order and first-order statistics for the j^{th} utterance are calculated as follows:

$$N_c = \sum_{t=1}^{L} P(c|\mathbf{z_t^j}, \hat{\lambda}) \tag{5}$$

$$\mathbf{F_c} = \sum_{t=1}^{L} P(c|\mathbf{z_t^j}, \hat{\lambda})(\mathbf{y_t^j} - \hat{\mu}_\mathbf{c}) \tag{6}$$

$$\hat{\mu}_\mathbf{c} = \frac{\sum_{j=1}^{J} \sum_{t=1}^{L} P(c|\mathbf{z_t^j}, \lambda)\mathbf{y_t}}{\sum_{j=1}^{J} \sum_{t=1}^{L} P(c|\mathbf{z_t^j}, \lambda)}. \tag{7}$$

Table 2: Performance of the proposed methods on the NIST SRE 2010 core condition 5 female part task (original trials)

ID	Methods	Tokens	Token language	Token number	Feature for first order statistics	EER %	norm old minDCF
1	conventional i-vector baseline	MFCC-GMM		1024	MFCC	**3.13**	**0.176**
2	Phonemes-MFCC	monophone states	English	123	MFCC	2.76	0.151
3	Phonemes-MFCC	monophone states	Mandarin	537	MFCC	4.51	0.212
4	Phonemes-MFCC	monophone states	Czech	138	MFCC	4.53	0.231
5	Phonemes-MFCC	monophone states	Hungarian	186	MFCC	4.73	0.221
6	Phonemes-MFCC	monophone states	Russian	159	MFCC	4.80	0.219
7	Fusion of methods 2+3+4+5+6					2.76	0.136
8	Tandem-GMM-MFCC	Tandem-GMM	English	1024	MFCC	**2.50**	**0.12**
9	Tandem-GMM-Tandem	Tandem-GMM	English	1024	Tandem	3.11	0.16
10	Trigrams-MFCC	Trigrams	English	1024	MFCC	4.48	0.234
11	Hybrid-GMM-Hybrid	Hybrid-MFCC	English	1024	Hybrid	**1.97**	**0.96**
12	Fusion of methods 2+11					**1.67**	**0.82**

where $c = 1, \cdots, C$ is the new token index and $P(c|\mathbf{z_t^j}, \hat{\lambda})$ is the posterior probability for the j^{th} utterance's feature vector at time t on the c^{th} token. Note that the feature ($\mathbf{z_t}$) used to calculate the posterior probability $P(c|\mathbf{z_t}, \hat{\lambda})$ and the feature ($\mathbf{y_t}$) for cumulating the first-order statistics $\mathbf{F_c}$ are not necessarily the same. They can be different just as shown in Table 1. Global mean $\hat{\mu}_\mathbf{c}$ is computed using all the training data in the same way as the mean parameter estimation in GMM. Similarly, we also calculated the second-order statistics for the simplified supervised i-vector modeling.

The proposed methods with different combinations of tokens and features for statistics calculation are shown in Table 1. First, in the conventional i-vector baseline, both $\mathbf{z_t}$ and $\mathbf{y_t}$ in (5,6) are MFCC features and the tokens are the MFCC trained GMM components. Second, in the Phonemes-MFCC system, the tokens are the phonemes and the posterior probability $P(c|\mathbf{z_t}, \hat{\lambda})$ is the phoneme posterior probability (PPP). We employed the multilayer perceptron (MLP) based phoneme recognizer [22] with acoustic models from five different languages, namely Czech, Hungarian, Russian, English and Mandarin. The models for the first three languages were trained on SpeechDat-E databases and provided in [22]. Additionally, we trained the English and Mandarin based models both with 1000 neurons in all nets using the switchboard, fisher databases and the call friend, call home databases, respectively.

Since there are only limited amount of phoneme tokens (around 8 times less than the GMM components for English), the system performance is affected due to the broad coverage of each phoneme token. Here we propose two different methods to generate tokens with comparable size of GMM components. First, the PPP features are converted into tandem features by log transform, principal component analysis (PCA) and mean variance normalization (MVN) [13, 14, 17] as shown in fig. 2. Then we directly consider this tandem feature as $\mathbf{z_t}$ in (5,6) and train a GMM on top of it to generate the Tandem-GMM tokens. In this setup, the entire GMM statistics calculation remains the same except that the GMM model is trained on the tandem features. Second, we increase the time scale of tokens and adopt the trigrams as the new type of tokens. As shown in fig. 2, HTK toolkit [23] is used to decode the PPP features and output a lattice file for each utterance which is further processed into n-gram counts and n-gram indexes by the lattice-tool toolkit [24]. The decoded n-gram counts are considered as the posterior probability and the mean of features within this n-gram's range is accounted as $\mathbf{y_t}$ where t indexes the whole n-gram here.

Both tandem features and MFCC features can be used (as $\mathbf{z_t}$) to train a GMM tokenizer and both could be projected on tokens (as $\mathbf{y_t}$) for calculating the first-order statistics. Therefore, we further explore the hybrid features which concatenate the acoustic MFCC feature and the phonetic tandem features at the frame level for both purposes. This hybrid feature level fusion setup not only achieves better performance but also directly fit the conventional i-vector framework.

3. Experimental results

3.1. Results on SV

We first conducted experiments on the NIST 2010 speaker recognition evaluation (SRE) corpus [25]. Our focus is the female part of the common condition 5 (a subset of tel-tel) in the core task. We used equal error rate (EER) and the normalized old minimum decision cost value (norm old minDCF) as the metrics for evaluation [25]. For cepstral feature extraction, a 25ms Hamming window with 10ms shifts was adopted. Each utterance was converted into a sequence of 36-dimensional feature vectors, each consisting of 18 MFCC coefficients and their first derivatives. We employed the Czech phoneme recognizer [22] to perform the voice activity detection (VAD) by simply dropping all frames that are decoded as silence or speaker noises. Feature warping is applied to mitigate variabilities.

The training data for NIST 2010 task include Switchboard II part1 to part3, NIST SRE 2004, 2005, 2006 and 2008 corpora on the telephone channel. The gender-dependent GMM UBMs consist of 1024 mixture components. Token numbers are shown in Table 2 and the tandem feature dimension is 52. Both LDA (500→150) and WCCN are adopted for variability compensation. The PLDA implementation is based on the UCL toolkit [8] where the sizes of speaker loading matrix and variability loading matrix are 150 and 80, respectively. Simple weighted linear summation is adopted here as the score level fusion.

In Table 2, we can see that the English Phonemes-MFCC system outperformed the i-vector baseline (3.13%→2.76% EER) by using only 123 phoneme tokens which supports our claim that phonetic tokens help. Since majority of the NIST SRE data samples are from English, other language based phoneme tokens are not as effective as the English one and combining systems with phoneme tokens from multiple languages only improved the cost value. This might be more useful in the multi-lingual or multi-dialects SV scenarios. So we only apply the English phoneme recognizer for other phonetic tokens. Furthermore, in system ID 8 and 9, we adopt the tandem-GMM components as the tokens and evaluated different features for the first-order statistics calculation. Results show that MFCC feature is better than tandem feature in this case for SV tasks.

Table 3: Performance on the NIST LRE 2007 general language recognition closed set 30 seconds task

ID	Methods	Tokens	Token language	Token number	Feature for first order statistics	EER %	min C_{avg}%
1	MFCC-GMM-MFCC baseline	MFCC-GMM		2048	MFCC	**2.59**	**2.61**
2	Phonemes-MFCC	monophone states	Czech	138	MFCC	3.43	3.54
3	Phonemes-MFCC	monophone states	Hungarian	186	MFCC	3.80	3.93
4	Phonemes-MFCC	monophone states	Russian	159	MFCC	3.43	3.40
5	Fusion of methods 2+3+4					**2.50**	**2.56**
6	Tandem-GMM-Tandem	Tandem-GMM	Czech	2048	Tandem	2.30	2.42
7	Tandem-GMM-Tandem	Tandem-GMM	Hungarian	2048	Tandem	2.22	2.15
8	Tandem-GMM-Tandem	Tandem-GMM	Russian	2048	Tandem	2.50	2.47
9	Fusion of methods 6+7+8					**1.81**	**1.80**
10	Trigrams-Tandem	Trigrams	Czech	2048	Tandem	4.17	4.39
11	Trigrams-Tandem	Trigrams	Hungarian	2048	Tandem	4.08	4.18
12	Trigrams-Tandem	Trigrams	Russian	2048	Tandem	5.0	5.26
13	Fusion of methods 10+11+12					**2.97**	**3.08**
14	Fusion of methods 1+9					**1.34**	**1.41**

When applying GMM on top of the tandem features, the number of tokens become comparable to the baseline GMM size which leads to the significant performance enhanced by 16.2% relative EER reduction. Trigrams tokens based system did not improve the performance which might be because its scale is too large for SV compared to those triphone states in [10].

Finally, the Hybrid-GMM-Hybrid single system achieved 1.97% EER and 0.96 norm old minDCF, which outperformed the i-vector baseline by relatively 37% and 45%, respectively. This is very promising since in this setup the entire GMM i-vector framework remains the same, only features are enhanced to the hybrid ones. Moreover, since this Hybrid-GMM-Hybrid setup already covers information from methods ID 1,8 and 9, we only fuse English Phonemes-MFCC system with it at the score level to generate the final results. Results show that these two methods are complementary to each other. Compared to the i-vector baseline, the proposed methods achieved 46% and 53% relative error reduction in terms of EER and norm old minDCF.

3.2. Results on LID

We also adopted the 2007 NIST Language Recognition Evaluation (LRE) [26] 30 seconds closed set general task as the evaluation database for LID. Data of target languages from Call Friend, OGI Multilingual, OGI 22 languages, NIST LRE 1996, NIST LRE 2003, NIST LRE 2005, NIST LRE 2007 supplemental training as well as a subset of NIST SRE 2004-2006 were used as our training data. We first extracted the 56 dimensional MFCC-SDC feature, then employed phoneme recognizers [22] to perform speech activity detection. We divided the features of each training conversation into multiple 30 seconds (3000 frames) segments. There are totally 81848 training segments, 2158 testing utterances, and 30212 testing trials. A 2048 components GMM UBM model was trained from 20000 training segments randomly selected from the training data. After statistics vectors were calculated, the simplified supervised i-vector modeling was applied. The back end variability compensation method (WCCN) and the classification method (second order polynomial kernel SVM) are the same as in [21, 7]. The performance is reported in EER and optimum average cost C_{avg} value as suggested by [26].

From Table3, we can observe that phoneme tokens from a single language did not improve the LID performance, potentially due to the limited amount of phoneme tokens. However, when we combined systems with phoneme tokens from different languages, the overall performance was enhanced (method 5). This makes sense because phonetic or phonotactic LID systems usually employ parallel phoneme recognizers from different languages [12, 27]. Furthermore, the combined tandem-GMM-tandem system (method 9) achieved 1.81% EER which outperformed the i-vector baseline by 30% relatively. This finding matches with the SV results which indicates that applying GMM on top of phoneme tokens are necessary and tandem features are more effective than MFCC as features for the first-order statistics calculation in LID. We note that this method (ID 6-8) is exactly the same as the one presented in [18], and is a special case in our generalized framework. Moreover, we can see that the Trigrams-Tandem systems (method 10-13) is less effective than the Tandem-GMM-Tandem system which matches the results in SV experiments. The underlying reason might be that the trigrams are too long to be considered as tokens and the trigrams posterior counts do not sum to 1.

Finally, by fusing the proposed phonetic tokens based methods with the i-vector baseline at the score level (method 14), the overall system performance was enhanced. The proposed generalized i-vector framework outperformed the i-vector baseline by relatively 48% and 46% in terms of EER and min C_{avg}, respectively. Our future works include applying the Hybrid-GMM-Hybrid method on the LID task and considering other types of phonetic tokens with relatively smaller scale in this generalized i-vector framework.

4. Conclusions

This paper presents a generalized i-vector framework with phonetic tokenizations and tandem features for speaker verification and language identification tasks. First, the tokens for calculating the zero-order statistics is extended from the MFCC trained GMM components to phonetic phonemes, 3-grams and tandem feature trained GMM components using phoneme posterior probabilities. We show that the Tandem-GMM tokens are superior than the phonemes and trigrams in terms of performance. Since the features for extracting tokens and the features for calculating the first-order statistics are not necessary the same , we show that in terms of first-order statistics calculation, MFCC is superior than tandem features for SV, and verse visa for LID. We further explore the hybrid features which concatenate the acoustic MFCC and the phonetic tandem features at the frame level for both purposes. This setup not only achieves better performance but also fit the conventional i-vector framework. Score level fusion of systems with different tokens and features further improves the overall system performance.

5. References

[1] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, “Language recognition via i-vectors and dimensionality reduction,” in *Proc. INTERSPEECH*, 2011, pp. 857–860.

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

[3] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, “Language recognition in ivectors space,” in *Proc. INTERSPEECH*, 2011, pp. 861–864.

[4] A. Hatch, S. Kajarekar, and A. Stolcke, “Within-class covariance normalization for SVM-based speaker recognition,” in *Proc. INTERSPEECH*, vol. 4, 2006, pp. 1471–1474.

[5] W. Campbell, D. Sturim, and D. Reynolds, “Support vector machines using gmm supervectors for speaker verification,” *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[6] P. Matejka, O. Plchot, M. Soufifar, O. Glembek, L. DHaro, K. Vesely, F. Grezl., J. Ma, S. Matsoukas, and N. Dehak, “Patrol team language identification system for darpa rats p1 evaluation,” in *Proc. INTERSPEECH*, 2012.

[7] K. Han, S. Ganapathy, M. Li, M. Omar, and S. Narayanan, “Trap language identification system for rats phase ii evaluation,” in *Proc. INTERSPEECH*, 2013.

[8] S. Prince and J. Elder, “Probabilistic linear discriminant analysis for inferences about identity,” in *Proc. ICCV*, 2007, pp. 1–8.

[9] P. Matejka, O. Glembek, F. Castaldo, M. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cernocky, “Full-covariance ubm and heavy-tailed plda in i-vector speaker verification,” in *Proc. ICASSP*, 2011, pp. 4828–4831.

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

[11] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, “Deep neural networks for extracting baum-welch statistics for speaker recognition,” in *Proc. ICASSP*, 2014.

[12] H. Li, B. Ma, and C. Lee, “A vector space modeling approach to spoken language identification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 271–284, 2007.

[13] H. Hermansky, D. P. Ellis, and S. Sharma, “Tandem connectionist feature extraction for conventional hmm systems,” in *Proc. ICASSP*, vol. 3, 2000, pp. 1635–1638.

[14] D. P. Ellis, R. Singh, and S. Sivadas, “Tandem acoustic modeling in large-vocabulary recognition,” in *Proc. ICASSP*, vol. 1, 2001, pp. 517–520.

[15] J. Pinto, S. Garimella, H. Hermansky, H. Bourlard, *et al.*, “Analysis of mlp-based hierarchical phoneme posterior probability estimator,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 2, pp. 225–241, 2011.

[16] Q. Zhu, A. Stolcke, B. Y. Chen, and N. Morgan, “Using mlp features in sris conversational speech recognition system,” in *Proc. INTERSPEECH*, 2005.

[17] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, “Shifted-delta mlp features for spoken language recognition,” *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 15–18, 2013.

[18] L. DHaro, R. Cordoba, C. Salamea, and J. Echeverry, “Extended phone log-likelihood ratio features and acoustic-based i-vectors for language recognition,” in *Proc. ICASSP*, 2014.

[19] P. Kenny, G. Boulianne, and P. Dumouchel, “Eigenvoice modeling with sparse training data,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, 2005.

[20] M. Li, A. Tsiartas, M. Van Segbroeck, and S. S. Narayanan, “Speaker verification using simplified and supervised i-vector modeling,” in *Proc. ICASSP*. IEEE, 2013, pp. 7199–7203.

[21] M. Li and S. Narayanan, “Simplified supervised i-vector modeling with application to robust and efficient language identification and speaker verification,” *Computer speech and language*, 2014.

[22] P. Schwarz, P. Matejka, and J. Cernocky, “Hierarchical structures of neural networks for phoneme,” in *Proc. ICASSP*, 2006, pp. 325–328, software available at http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context.

[23] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK book*. Entropic Cambridge Research Laboratory Cambridge, 1997, vol. 2.

[24] A. Stolcke *et al.*, “Srilm-an extensible language modeling toolkit,” in *Proc. INTERSPEECH*, 2002.

[25] NIST, “The NIST 2010 Speaker Recognition Evaluation Plan,” *www.itl.nist.gov/iad/mig/tests/spk/2010/index.html*, 2010.

[26] NIST., “The 2007 nist language recognition evaluation,” *http://www.itl.nist.gov/iad/mig/tests/lre/2007/*, 2007.

[27] M. Zissman, “Language identification using phoneme recognition and phonotactic language modeling,” in *Proc. ICASSP*, vol. 5, 1995, pp. 3503–3506.