LATTICE MLLR BASED *M-VECTOR* SYSTEM FOR SPEAKER VERIFICATION

A. K. Sarkar[1]*, C. Barras*[1] *and V. B. Le*[2]

[1]LIMSI-CNRS, Université Paris-Sud, BP 133, 91403 Orsay, France
[2]Vocapia Research, 28 Rue Jean Rostand, Parc Orsay Université, 91400 Orsay, France
{sarkar,barras}@limsi.fr, levb@vocapia.com

ABSTRACT

The recently introduced *m-vector* approach uses Maximum Likelihood Linear Regression (MLLR) super-vectors for speaker verification, where MLLR super-vectors are estimated with respect to a Universal Background Model (UBM) without any transcription of speech segments and speaker *m-vectors* are obtained by uniform segmentation of their MLLR super-vectors. Hence, this approach does not exploit the phonetic content of the speech segments. In this paper, we propose the integration of an Automatic Speech Recognition (ASR) based multi-class MLLR transformation into the *m-vector* system. We consider two variants, with MLLR transformations computed either on the 1-best (hypothesis) or on the lattice word transcriptions. The former case is able to account for the risk of ASR transcription errors. We show that the proposed systems outperform the conventional method over various tasks of the NIST SRE 2008 core condition.

Index Terms: m-Vector, Lattice MLLR, MLLR Super-Vector, Session Variability Compensation, Speaker Verification

1. INTRODUCTION

Maximum Likelihood Linear Regression (MLLR) super-vectors are known to carry speaker related information. They were first introduced in speaker verification by Stolcke et al. [1], followed by several variants [2, 3]. In these systems, MLLR super-vectors are used for speaker modeling in a Support Vector Machine (SVM) framework, and an Automatic Speech Recognition (ASR) front-end is generally used for estimating several MLLR transformations for a given speaker speech segment with respect to pre-defined phonetic classes. MLLR transformations are then concatenated to form a MLLR super-vector.

Recently, a new way of representing speakers with MLLR super-vectors was proposed [4, 5, 6]. In [5], an eigen voice anchor modeling was proposed for speaker identification, where speakers are characterized by Speaker Characterization Vectors (SCVs). SCVs are estimated by projecting the speakers MLLR super-vector on an eigen voice space. The eigen voice space is generated by singular value decomposition of MLLR super-vectors pooled from many speakers. During the identification phase, SCV of the test utterance is scored against the known speakers with a cosine similarity measure. The proposed method was shown to be computationally very efficient and to perform significantly better than the anchor modeling techniques described in the literature [7, 8]. In [4], a system called *m-vector* was proposed, where speakers are represented by a uniform segmentation of their MLLR super-vector using an overlapped sliding window. Each segment of the MLLR super-vector is called an *m-vector*. Hence, each speaker is characterized by a number of *m-vectors* depending on the window size. During test phase, *m-vectors* of the test utterance are scored against the claimant specific *m-vectors*. Before scoring, *m-vectors* are conditioned for session variability compensation. It was shown in [4] that *m-vector* system is analogous to i-vector [9] based speaker verification system and showed promising performance. It is also able to capture more speaker related information available in the MLLR super-vector and hence shows significantly better performance compared to the system which uses full MLLR super-vectors to characterize the speakers (for details see [4]). To the best of our knowledge, these were the first approaches with MLLR super-vectors trying to depart from the conventional SVM modeling. However, they use Universal Background Model (UBM) for estimating the MLLR transformation without any phonetic knowledge of the speech segment.

In this paper, we propose an Automatic Speech Recognition (ASR) transcription based (phonetic) multi-class MLLR super-vector for *m-vector* in speaker verification. More precisely, ASR is used as a front-end for getting the transcriptions of the speech segments. Speech transcriptions are then used to compute phonetic class wise MLLR transformation with respect to Speaker Independent (SI) Hidden Markov Models (HMM). We consider two ways of using these speech transcriptions in the MLLR transformation. The first one is the 1-best hypothesis, which is conventionally used in ASR for MLLR transformation [10]. The other one is based on lattice word transcriptions [11], which are able to account for the risk of transcription errors. This results in the proposition of two *m-vector* systems called *ASR 1-best* and *ASR-Lattice*, respectively. The main difference with the conventional system [4] are that : (i) MLLR super-vectors are extracted using phonological knowledge (i.e. ASR speech transcriptions) using 1-best and lattice transcriptions in contrast to [4]; (ii) multi-class MLLR transformations are used with respect to phonetic classes in contrast to a single, global class in [4]. The experimental results show that the proposed system performs significantly better than conventional *m-vector* system, for various tasks of NIST SRE 2008 core condition.

The paper is organized as follows: Section 2 and 3 describe the conventional 1-best hypothesis and lattice based MLLR transformation, respectively. *m-vector* concept is described in Section 4. Section 5 describes the proposed system. Section 7 describe session variability compensation and scoring method. Test phase and baseline system are described in Section 8 & 6, respectively. Experimental setup is described in Section 9. Results and discussion are presented in Section 10. Finally, the paper concludes with Section 11.

This work was partly realized as part of the Quaero Program funded by OSEO (French State agency for innovation).

2. CONVENTIONAL (1-BEST) MLLR

MLLR [10] is commonly used for speaker adaptation in HMM-based ASR systems. It estimates an affine transformation with respect to a Speaker Independent (SI) HMM in the Maximum Likelihood (ML) sense for a given speech data as [10]:

$$\hat{\theta} = \arg \max_{\theta} log\ p(X_r|\theta) \quad (1)$$

where X_r represents the feature vectors of r^{th} speaker. $\hat{\theta}$ denotes the adapted model parameter of the speaker for a state/tied state, s in SI model as, $\left\{\hat{\mu_s} = A\mu_s + b; \quad \hat{\Sigma_s} = \Sigma_s\right\}$, where μ_s and Σ_s are the Gaussian mean and covariance matrix of s^{th} state in SI (HMM) model, respectively. (A, b) is called the MLLR transformation.

3. LATTICE MLLR

Automatic transcriptions of telephone conversations present typical word error rates of $20-30\%$, so the conventional MLLR transformation using Eq.(1) with the 1-best hypothesis often misses the correct acoustic model. To account for the transcription errors, lattice-based MLLR transforms [12, 13] are estimated using the word-lattice output of an ASR system obtained in a first-pass decoding, converted into a model-level graph using the pronunciation variants in the lexicon. MLLR transformation for a given speech data of r^{th} speaker is estimated as:

$$\hat{\theta} = \arg \max_{\theta} \sum_{s_r \epsilon \mathbf{S}} p(s_r|X_r, \theta)\ log\ p(X_r, s_r|\theta) \quad (2)$$

where $p(s_r|X_r, \theta)$ is the probability of aligning the r^{th} speaker training data, X_r with respect to state sequence s_r using SI (HMM) model, θ. $\mathbf{S}$ indicates all possible alignment state sequences of X_r with respect to the SI model. In the conventional approach in Eq.(1), $p(s_r|X_r, \theta)$ is set to 1 for the 1-best hypothesis (i.e. 1-best state sequence) and 0 for others. Details about the use of lattice MLLR approach for speaker verification can be found in [11].

4. M-VECTOR TECHNIQUE

Recently, speaker verification using the *m-vector* technique has been introduced in [4]. In this approach, speakers are characterized by their *m-vectors* which are extracted from their MLLR super-vectors by uniform segmentation using overlapped sliding windows. Fig.1 graphically illustrates MLLR super-vector estimation of the r^{th} speaker using his/her training data with respect to a SI HMM or UBM model.

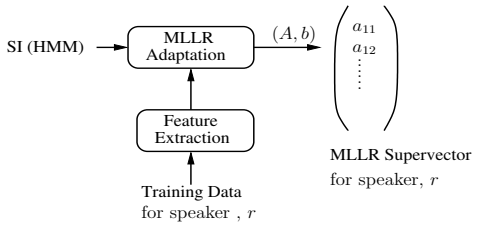

Fig. 1. *MLLR super-vector extraction from MLLR transformation of the r^{th} speaker using his/her training data with respect to a speaker independent HMM.*

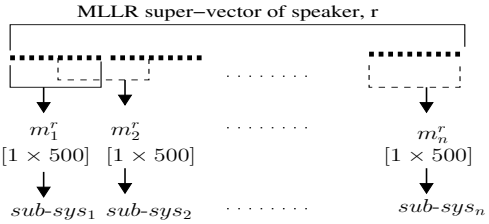

Fig. 2. m-vector *extraction for the r^{th} speaker from his/her MLLR super-vector using an overlapped sliding window of* 500 *elements with* 50% *overlap of its adjacent* m-vectors.

As per [4], two *m-vectors* extraction are considered,
Full: speakers are represented by their full MLLR super-vectors [5].
Overlapped: MLLR super-vector is uniformly segmented using an overlapped sliding window, as illustrated on Fig. 2. In this case, a speaker is represented by several *m-vectors* which are processed separately and hence constitute several sub-systems. The size of window and overlap control the number of *m-vectors* to be derived from a MLLR super-vector. When the size of the MLLR super-vector is not a multiple of the window size, an additional *m-vector* is extracted by placing the window at the end point of the super-vector, so as to cover all elements of the super-vector.

During the test phase, *m-vectors* of the test utterance are scored against the claimant specific *m-vectors*. Before scoring, *m-vectors* are conditioned for session variability compensation similarly to i-vectors. Linear Discriminant Analysis (LDA) [14] is applied on the *m-vectors* (before conditioning) to discriminant the speakers. In the overlapped case, each sub-system has its own LDA projection matrix. LDA is implemented using a number of example from 890 non-target speakers.

5. PROPOSED SYSTEMS

5.1. ASR 1-best m-vector system

In this system, MLLR transformations of a speech segment is calculated using Eq.(1) i.e. 1 best hypothesis. We use 42 dimensional feature vectors and two phonetic classes (vowels and consonants) and obtain a 42×42 dimensional MLLR transformation for each phonetic class (the bias b is discarded since it does not provide significant gain in our setup). Totally, we get a $(2 \times 42 \times 42) = 3528$ dimensional MLLR super-vector. During training, target speakers are represented by their *m-vectors* extracted from their MLLR super-vectors.

5.2. ASR-Lattice m-vector system

This system is similar to *ASR 1-best m-vector system*. The only difference is that it uses lattice MLLR concept for MLLR transformations as Eq.(2).

6. BASELINE SYSTEM

A single class UBM based *m-vector* system is considered as the baseline system [4]. In this system, a global MLLR transformation is calculated with respect to the UBM without any speech transcriptions of the speech segments. It is similar to Eq.(1), where the UBM is considered as the SI model. This results in a 1764 dimensional

MLLR super-vector (for 42 dimensional feature vectors). Target speakers are then represented by the *m-vectors* extracted from their MLLR super-vectors. Details about the system can be found in [4].

7. SESSION VARIABILITY COMPENSATION AND SCORING

For session variability compensation, we apply the Eigen Factor Radial (EFR) technique recently proposed [15] on *m-vectors*. EFR iteratively normalize the length of the i-vectors, w as Eq.(3) to handle the session variability.

$$\hat{w} \leftarrow \frac{V^{-\frac{1}{2}}(w - \overline{w})}{\sqrt{(w - \overline{w})^t V^{-1}(w - \overline{w})}} \tag{3}$$

where V and $\overline{w}$ indicate the covariance matrix and mean vector of the training i-vectors, respectively in successive iteration. During test, the score between the two conditioned i-vectors (i.e. $\hat{w}_1, \hat{w}_2$) are calculated using the Mahalanobis distance measure,

$$score(\hat{w}_1, \hat{w}_2) = (\hat{w}_1 - \hat{w}_2)^t \Omega^{-1} (\hat{w}_1 - \hat{w}_2) \tag{4}$$

where Ω is the within-class covariance matrix calculated using development data set.

Several other session variability compensation techniques can be found in literature, namely LDA [14], Within Class Covariance Normalization (WCCN) [16] and Probabilistic (P)-LDA [17, 18] etc., but it was shown in [15] that EFR performs better than conventional LDA + WCCN method. We thus use the EFR algorithm for conditioning the *m-vectors* and handling the session variability, and Mahalanobis distance measure for scoring. Two iterations are considered during conditioning for all systems presented in the paper.

8. TEST PHASE

m-vectors are extracted from the test utterance using their respective systems and projected on the particular LDA space. Finally, LDA projected *m-vectors* are conditioned using the EFR algorithm and scored against the claimant specific *m-vector* obtained during training phase. In case of the *overlapped* method, scores of the different sub-systems are fused for a particular LDA dimension across all sub-systems. For fusion, equal weights are given to all sub-systems as,

$$fuse\ score = \frac{1}{N_{subsys}} \sum_{i=1}^{N_{subsys}} score(\tilde{m}_i^r, \tilde{m}_i^{test}) \tag{5}$$

where $\tilde{m}_i^r$ and $\tilde{m}_i^{test}$ are the conditioned *m-vectors* of claimant, r and test utterance for the i^{th} subsystem, respectively. $score(.,.)$ indicates the scoring function between the two *m-vectors*.

9. EXPERIMENTAL SETUP

All experiments are performed on NIST SRE 2008 core condition (male speakers) as per plan [19]. There are 1270 speech utterances for training 1270 target models. Each utterance is approximatively 5 minutes long with 2.5 minutes of speech in average.

For spectral analysis, 42 dimensional vectors including 12 Mel-PLP feature, log-energy and F_0 along with their first- and second-order derivatives are extracted from the speech signal each 10 ms using a 30 seconds Hamming window over bandwidth 0-3800Hz. Voice activity detection is applied as a pre-processing step to discard less energized or silent frames. Finally, detected speech segments are normalized to zero mean and unit variance at the utterance level.

The Large Vocabulary Continuous Speech Recognition (LVCSR) system used MLLR transforms estimation is similar to the LIMSI RT'04 LVCSR system [20]. The acoustic models are trained on about 2000 hours of manually transcribed Conversational Telephone Speech (CTS) data using the PLP+F_0 features concatenated with additional MLP features [21]. The model sets cover about $48k$ phone contexts, with $11.5k$ tied states and 32 Gaussians per state. Silence is modeled by a single state with 1024 Gaussians. Two manually derived phonetic classes: vowels and consonants are used for MLLR transformations, estimated on the PLP+F_0 features only. Audio segments aligned with the silence model after the decoding are not considered for the MLLR transformation relying on the ASR transcriptions, but are kept for the baseline system.

A male UBM of 512 mixture component Gaussian Mixture Models (GMMs) having diagonal covariance matrix, is trained using data from NIST 2004 SRE. All systems use a single iteration for MLLR transformation. LDA and EFR algorithm are implemented using 890 non-target data from NIST 2004-2005, Switchboard II part 1, 2 & 3, Switchboard cellular part 1 & 2, with about 15 sessions per speaker. It gives totally 12399 utterances i.e. 12399 MLLR super-vectors.

10. RESULTS AND DISCUSSION

Table 1 compares the speaker verification performance of the proposed systems with the baseline system on NIST 2008 SRE core condition over various tasks. Each task is associated with different condition (e.g. telephone, interview, microphone type etc) of target speaker training and testing (e.g. Det 1: microphone-microphone, Det 7: telephone-telephone etc). The performance of the overlapped *m-vector* systems are shown for *m-vector* size of 500 elements which correspond to size of the sliding window and gives the best results for all systems in our experiments. For fusion, equal weights are given to all systems. From table, we can make the following observations:

- *Overlap* method shows better results compared to the *full* in all respective systems with LDA in terms of EER and MinDCF. This indicates that *overlapped* method is able to capture more speaker relevant information from MLLR super-vector than *full*.
- Fusion of *overlap* system with *full* further improves the speaker verification performance in all respective systems. This indicates that *full* system also contains the complementary information (which is not covered by *m-vectors* i.e. presents on full MLLR super-vector) for the *overlap* system.
- The proposed ASR based *m-vector* systems significantly show better performance than baseline system in terms of EER and MinDCF value. Lattice based system shows again lower EER and MinDCF compared to 1-best. This also reflects its (lattice) accountability of erroneous in speech transcription for MLLR transformation over the 1-best hypothesis method.

Fig.3 compares the Detection Error Tradeoff (DET) plots of the proposed systems with the baseline over various tasks on NIST 2008 SRE core condition. From Fig.3, we can observe that the proposed systems perform consistently better than the baseline system over a large region of the DET curve.

Table 1. *Comparison of performance of the proposed systems with baseline system on NIST 2008 SRE core condition over different tasks.*

System	m-vector		LDA	DET task: (%) EER (MinDCF)					
	extraction method	dim.	Proj. dim.	1	3	4	5	6	7
Baseline	(A1) Full	1764	50	15.00 (0.0614)	15.54 (0.0641)	15.55 (0.0612)	10.53 (0.0467)	9.32 (0.0485)	6.67 (0.0362)
	(A2) Overlapped	500	50	14.92 (0.0588)	15.34 (0.0611)	13.00 (0.0514)	9.55 (0.0380)	7.70 (0.0378)	5.74 (0.0271)
	Fusion (A1,A2)	-	-	**13.95 (0.0557)**	**14.37 (0.0579)**	**12.63 (0.0491)**	**8.55 (0.0353)**	**7.70 (0.0382)**	**5.51 (0.0259)**
Proposed-ASR 1 best	(B1) Full	3528	50	12.86 (0.0492)	13.30 (0.0510)	10.36 (0.0451)	8.45 (0.0337)	6.34 (0.0395)	3.89 (0.0214)
	(B2) Overlapped	500	50	12.51 (0.0480)	12.88 (0.0498)	9.61 (0.0417)	7.69 (0.0293)	5.89 (0.0354)	3.18 (0.0160)
	Fusion(B1,B2)	-	-	**11.82 (0.0445)**	**12.09 (0.0464)**	**8.77 (0.0407)**	**7.34 (0.0274)**	**5.89 (0.0346)**	**3.13 (0.0144)**
Proposed- ASR Lattice	(C1) Full	3528	50	11.98 (0.0470)	12.46 (0.0487)	9.39 (0.0410)	8.20 (0.0322)	7.05 (0.0379)	3.42 (0.0180)
	(C2) Overlapped	500	50	11.92 (0.0455)	12.25 (0.0471)	8.52 (0.0392)	7.09 (0.0260)	5.74 (0.0351)	2.97 (0.0162)
	Fusion(C1,C2)	-	-	**11.21 (0.0426)**	**11.52 (0.0439)**	**8.07 (0.0382)**	**6.75 (0.0259)**	**5.54 (0.0345)**	**2.88 (0.0147)**

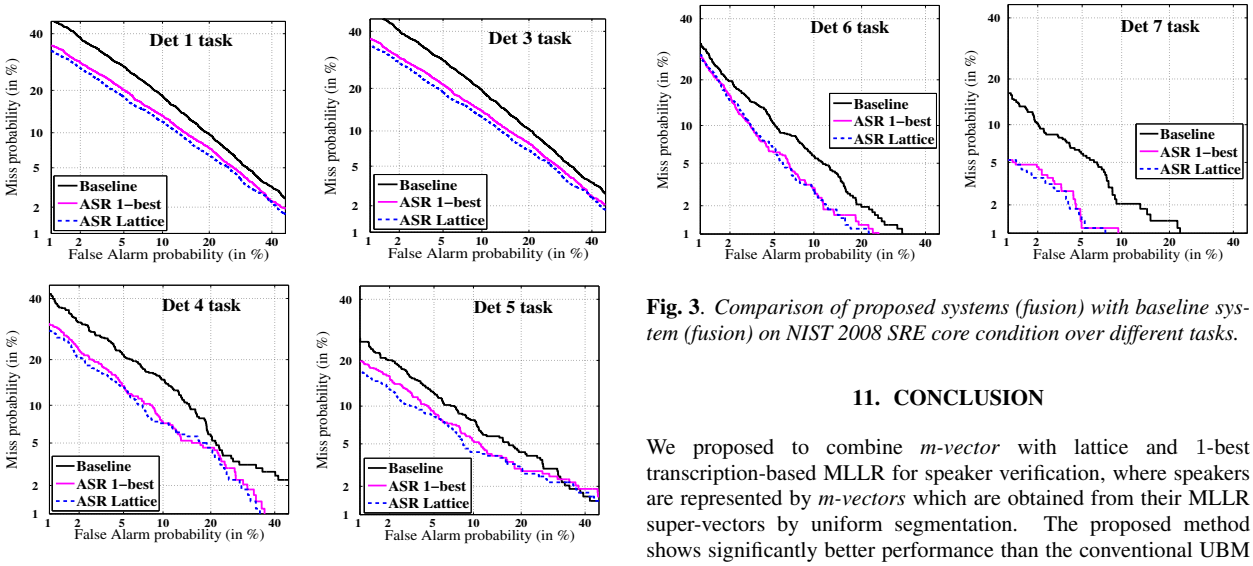

Fig. 3. *Comparison of proposed systems (fusion) with baseline system (fusion) on NIST 2008 SRE core condition over different tasks.*

11. CONCLUSION

We proposed to combine *m-vector* with lattice and 1-best transcription-based MLLR for speaker verification, where speakers are represented by *m-vectors* which are obtained from their MLLR super-vectors by uniform segmentation. The proposed method shows significantly better performance than the conventional UBM based *m-vector* system. Lattice based system accounts for the risk of ASR transcription errors (generally $20 - 30\%$ word error rate) and hence also shows better performance than 1-best conventional method, on various tasks on NIST SRE 2008 core condition.

12. REFERENCES

[1] A. Stolcke et al., "MLLR Transforms as Features in Speaker Recognition," in *Proc. of EUROSPEECH*, 2005, pp. 2425–2428.

[2] Z. N. Karam and W. M. Campbell, "A Multi-class MLLR Kernel for SVM Speaker Recognition," in *Proc. of ICASSP*, 2008, pp. 4117–4120.

[3] M. Ferras et al., "Constrained MLLR for Speaker Recognition," in *Proc. of ICASSP*, 2007, pp. 53–56.

[4] A. K. Sarkar, J. F. Bonastre, and D. Matrouf, "Speaker Verification using m-vector Extracted from MLLR Super-vector," in *Proc. of 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 21–25.

[5] A. K. Sarkar and S. Umesh, "Eigen-voice Based Anchor Modeling System for Speaker Identification using MLLR Supervector," in *Proc. of INTERSPEECH*, 2011, pp. 2357–2360.

[6] Nicolas Scheffer, Yun Lei, and Luciana Ferrer, "Factor Analysis Back Ends for MLLR Transforms in Speaker Recognition," in *Proc. of INTERSPEECH*, 2011, pp. 257–260.

[7] D. Sturim et al., "Speaker Indexing in Large Audio Databases using Anchor Models," in *Proc. of ICASSP*, 2001, pp. 429–432.

[8] A. K. Sarkar and S. Umesh, "Fast Computation of Speaker Characterization Vector using MLLR and Sufficient Statistics in Anchor Model Framework," in *Proc. of INTERSPEECH*, 2010, pp. 2738–2741.

[9] N. Dehak et al., "Front-End Factor Analysis for Speaker Verification," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 19, pp. 788–798, 2011.

[10] C. Leggetter and P. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of HMMs," *Computer Speech and Language*, vol. 9, pp. 171–186, 1995.

[11] M. Ferras, C. Barras, and J. L. Gauvain, "Lattice-based MLLR for Speaker Recognition," in *Proc. of ICASSP*, 2009, pp. 4537–4540.

[12] M. Padmanabhan, G. Saon, and G. Zweig, "Lattice-Based Unsupervised MLLR for Speaker Adaptation," in *Proc. of the ISCA ITRW ASR2000*, 2000, pp. 128–131.

[13] L.F. Uebel and P.C. Woodland, "Improvements in Linear Tranformation based Speaker Adaptation," in *Proc. of ICASSP*, 2001, pp. 49–52.

[14] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, New York: John Wiley & Sons, 2001.

[15] P. M. Bousquet, D. Matrouf, and J. F. Bonastre, "Intersession Compensation and Scoring Methods in the i-vectors Space for Speaker Recognition," in *Proc. of INTERSPEECH*, 2011, pp. 485–488.

[16] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-Class Covariance Normalization for SVM-Based Speaker Recognition," in *Proc. of ICSLP*, 2006, pp. 1471–1474.

[17] M. Senoussaoui et al., "Mixture of PLDA Models in I-Vector Space for Gender-Independent Speaker Recognition," in *Proc. of INTERSPEECH*, 2011, pp. 25–28.

[18] Simon J.D. Prince, "Computer Vision: Models Learning and Inference," in *Cambridge University Press, 2012, In press*.

[19] The NIST Year 2008 Speaker Recognition Evaluation Plan., "http://www.itl.nist.gov/iad/mig/tests/sre/2008/sre08_evalplan_release4.p .

[20] R. Prasad et al., "The 2004 BBN/LIMSI 20xRT English Conversational Telephone Speech Recognition System," in *Proc. of INTERSPEECH*, 2005, pp. 1645–1648.

[21] P. Fousek, L. Lamel, and J. L. Gauvain, "Transcribing Broadcast Data using MLP Features," in *Proc. of INTERSPEECH*, 2008, pp. 1433–1436.