An On-line Incremental Speaker Adaptation Technique for Audio Stream Transcription

Diego Giuliani and Fabio Brugnara

Human Language Technology Research Unit
FBK-irst - Fondazione Bruno Kessler
Via Sommarive 18, 38123 Trento, Italy
{giuliani,brugnara}@fbk.eu

Abstract

In this paper, a novel on-line incremental speaker adaptation technique is proposed for real time transcription applications such as automatic closed-captioning of live TV programs. Differently from previously proposed methods, our technique does not operate at utterance level but instead speaker change detection and clustering as well as speaker adaptation occur over a short chunk of the incoming audio signal. Incremental adaptation based on feature space maximum likelihood linear regression (fMLLR) is conducted w. r. t. a Gaussian mixture model (GMM) modeling the acoustic training data. Individual speakers are represented by fMLLR transforms, and these transforms are used for speaker clustering and for performing speaker adaptation. Speech recognition experiments show that the proposed incremental adaptation technique is effective, 6% relative reduction in word-error-rate (WER) w. r. t. a non-adaptive baseline system, when it is embedded in a on-line transcription system applied to transcribe television news broadcasts.

Index Terms: on-line speaker adaptation, fMLLR, real-time automatic speech recognition

1. Introduction

There are applications, such as closed-captioning of live TV programs and a meeting assistant, which require that automatically generated transcripts are delivered with low latency, that is within a small time interval w. r. t. the moment in which words are actually uttered [1, 2, 3]. In these applications, the transcription system needs to sequentially process the incoming audio stream and to handle speech from multiple speakers appearing in casual order. To cope with acoustic variability and improve recognition performance, techniques developed for on-line incremental speaker adaptation can be applied in combination with on-line speaker change detection and speaker clustering [4, 5, 6]. In the approaches investigated so far, the transcription system operates on utterance-like units. Speaker change detection is performed and a cluster label, necessary for speaker adaptation, is assigned to each incoming utterance. By relying on cluster labels, it is possible to incrementally enrich speaker adaptation parameters exploiting all past speaker utterances in the same cluster.

A Viterbi decoder, using partial traceback [7], can deliver output progressively during an utterance, with a delay tipically limited to a few words. A goal of this work was to introduce adaptation without counteracting this capability. Therefore, multiple processing of an incoming utterance before performing the final decoding step, as in [4, 5], is not a viable solution. To implement incremental adaptation under this tight operating condition, each incoming utterance is decoded having the system adapted based only on already seen data [6]. However, when there is a speaker change, an acoustic mismatch occurs between the incoming utterance and the adapted system. This has a negative impact on recognition performance, whose magnitude depends on how often speakers, or acoustic conditions, alternate in the audio stream and on the length of the utterances.

In this paper, to reduce adaptation and transcription latency, we propose to operate on small chunks of the audio signal instead of utterances. In this approach, for each new incoming chunk of the audio signal having a duration of 1 or 2 seconds, acoustic observations are first normalized by applying an fMLLR transformation, selected and estimated for the previous chunk, and immediately made available to the decoding process. Then, the algorithm determines whether the audio chunk belongs to one of the known speaker clusters or whether a new cluster must be created. For speaker clustering, we follow the integrated approach proposed in [6] in which individual speakers are represented by fMLLR transforms, and those transforms are used for speaker change detection and clustering as well as for speaker adaptation. Speaker clusters are built up as the audio progresses, and a new speaker is identified using a set of transforms that represent ”generic speakers” and speakers already seen. For each individual speaker cluster, incremental adaptation based on fMLLR is conducted w. r. t. a GMM modeling the training environment [8, 9, 10]. This avoids the need of an estimated transcription of adaptation data and makes the adaptation process independent from the decoding engine.

In addition to present in detail the proposed technique, the paper also discusses¡ specific issues related with the usage of the cepstral mean subtraction in an operating environment in which an utterance-like segmentation is not appropriate.

The rest of this paper is organized as follows. Section 2 introduces text-independent speaker adaptive acoustic modeling based on fMLLR. Incremental speaker clustering and adaptation are described in Section 3. Transcription systems are briefly described in Section 4. Experimental results are presented in Section 5. Some considerations on the computational load of the proposed adaptation technique are reported in Section 6. Final remarks are reported in Section 7.

2. fMLLR based speaker adaptation

To reduce the acoustic mismatch between training and testing acoustic conditions, state-of-the-art speech transcription systems embed acoustic adaptation. In fMLLR based adaptation, a single transformation matrix and a bias vector are used to linearly transform the input acoustic features as follows [11]:

$$\hat{o}_t = Ao_t + b = W\zeta_t$$

where W is the extended transformation matrix $[b, A]$ consisting of D row vectors $\{w_i\}$, each having $D + 1$ components, and ζ_t is the extended observation vector $[1, {o_t}^T]^T$ at time t.

The affine transformation matrix W is estimated on adaptation data in order to maximize the likelihood of the transformed acoustic observations w. r. t. the speech recognition models, i.e. continuous density hidden Markov models (HMMs), assuming a word level transcription of adaptation data [11]. Effectiveness of fMLLR in reducing the acoustic mismatch between the speech recognition models and the input acoustic data was proven on a number of different domains [11, 12, 9].

fMLLR offers an efficient and simple way for implementing speaker adaptive acoustic modeling allowing transformation of the acoustic data of each training and testing speaker instead of transforming acoustic model parameters. In [13, 14] we proposed a variant of the fMLLR-based speaker adaptive training in which transformation parameter estimation is carried out, both during training and testing, w. r. t. a set of “target” models which are different than the “recognition” models used for performing decoding of the test data. Leveraging on the possibility that the structure of the target and recognition models can be determined independently, in [8] we proposed to estimate transformation parameters, both during training and testing, with the aim of maximizing the likelihood of the transformed acoustic observations w. r. t. a GMM modeling the whole training data. With this *text-independent* variant, on-line adaptation can be performed without the need of estimating a transcription of the adaptation data making this variant very appealing when it is important to achieve computational efficiency and low latency adaptation [15, 10].

During training, speech segments in the training set are first clustered by using an agglomerative algorithm based the Bayesian information criterion (BIC) [16] and then for each cluster an fMLLR transformation is estimated w. r. t. the GMM and applied on acoustic observations in the cluster. Each cluster of speech segments can be interpreted as a speaker cluster. We also experimented by limiting the number of transformations estimated during training by imposing to 32 the number of clusters. This number was chosen in order to reduce the computational load for transformation selection during testing as it will be described in the following section. In this case, each cluster of speech segments is supposed to represent group of speakers. This approach was inspired by the works on linear vocal tract length normalization presented in [17, 18] where during training only a small set of fMLLR transformations are used, each of them trained only on speech segments associated to a specific warping factor.

3. Speaker clustering and adaptation

In incremental clustering for on-line speech-to-text systems [19], the clustering decision is made as soon as an audio segment is received. Being causal, this approach enables low-latency incremental speaker adaptation. In order to further reduce adaptation latency, instead of performing utterance-like segmentation of the incoming audio stream [5, 6], in this work, the incoming audio stream is progressively processed in short, non overlapping, chunks having constant duration, for example of 1 or 2 seconds.

In [6] an integrated approach to on-line speaker clustering and adaptation was proposed where individual speakers are represented by fMLLR transforms, and those transforms are used for both speaker clustering and adaptation. In this work, we adopted a similar approach.

3.1. Processing of the audio stream

Each incoming acoustic observation vector is copied in a buffer, normalized by applying the current fMLLR transformation matrix and delivered immediately to the recognition engine. When the buffer reaches the specified chunk size:

- Sufficient fMLLR statistics are collected
- A cluster label is assigned to the chunk according to the selection procedure described below
- A new transformation matrix is estimated for the selected cluster to be applied to the following chunk

The assumption is that the information in an audio chunk is enough to reliably perform speaker clustering and that the acoustic condition in the audio signal does not change too frequently over time so that the possible mismatch introduced by applying an fMLLR transformation selected on the basis of the content of the previous audio chunk does not impact too much. For the first audio chunk, a pre-trained global transformation is applied.

As a result, the recognition process is performed on normalized data while speaker labeling and gathering of sufficient statistics make use of the original acoustic observations.

3.2. Speaker change detection and clustering

Given an incoming chunk of the audio signal, the algorithm determines whether the audio chunk belongs to one of the known clusters of chunks or whether a new cluster must be created, being each cluster represented by an fMLLR transformation. Each cluster is assumed being formed by a collection of acoustically homogeneous chunks of signal. Ideally, a cluster represents speech from a speaker uttering under a certain acoustic condition, so, in the following, for the sake of simplicity, we will refer to clusters of audio chunks as speaker clusters.

Given the current audio chunk and the pool of fMLLR transformation matrices $\{W_p\}_{p=1}^{P}$ representing speaker clusters, transformations can be ranked according to the values of the auxiliary function. To this end, each available transformation matrix is plugged into the auxiliary function [17, 18]:

$$Q(\hat{W}, W_p) = \beta \log |A_p| - \tfrac{1}{2} \textstyle\sum_{d=1}^{D} (w_d^{(p)} G^{(d)} {w_d^{(p)}}^T - 2 w_d^{(p)} {k^{(d)}}^T)$$

where $G^{(d)}$ and $k^{(d)}$ are the fMLLR sufficient statistics gathered from the current audio chunk [11]. $\beta = \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t)$ is the total occupancy count where $\gamma_m(t)$ is the posterior probability, determined by the ”target” GMM, of Gaussian component m at time t. The best fitting transformation matrix is selected for the next processing steps.

Initially, the set of transformations is solely formed by the 32 transformation matrices pre-computed on training data as specified in Section 2. For the first audio chunk of the incoming audio stream, a new individual speaker profile, as it will be described later in the next section, is initialized and the set of available transformation is then enlarged to include the individual speaker transformation matrix estimated on the initial audio chunk. For the following audio chunks, if the best fitting transformation corresponds to one of the 32 pre-computed transformations it means that a new speaker is detected and a new speaker profile is instantiated for this speaker while if the best fitting transformation corresponds to an individual speaker transformation it means that the current chunk contains speech belonging to an already seen speaker and the corresponding profile needs to be updated based on the current audio chunk content as described in the next section.

3.3. Incremental adaptation

As anticipated above, when a new speaker is detected a new speaker profile is instantiated. A speaker profile consists of fMLLR sufficient statistics (G_p, k_p and β_p) and the corresponding fMLLR transformation matrix (W_p) [6]. Sufficient statistics are incrementally collected over time, chunk by chunk, w. r. t. to the GMM and the transformation matrix re-estimated accordingly.

As only little adaptation data is initially available for a speaker profile, a transformation matrix can be hardly estimated reliably. To tackle the problem of unreliable parameter estimation due to sparse adaptation data, smoothing of sufficient statistics [12, 20], *maximum a posteriori* fMLLR (fMAPLR) [21] and eigen space fMLLR [22, 23] were proposed in the past. In this work, we experimented with the two former approaches as follows.

Smoothing of sufficient statistics, that is G_p, k_p and β_p, is achieved by properly initializing sufficient statistics of a new speaker profile with prior sufficient statistics [20]. Prior sufficient statistics were gathered during acoustic model training for estimating the 32 pre-computed fMLLR transformations. Sufficient statistics of the selected pre-computed transformation are properly weighted and used to initialize sufficient statistics of the new profile. The weight of prior statistics was set to 1000 based on preliminary experiments.

To implement fMAPLR, parameters of the prior distribution of the parameters of the fMLLR transformation were estimated from the parameters of the 32 pre-computed transformations [10]. At run time, given this prior distribution and sufficient statistics of a speaker profile, the fMLLR transformation matrix is estimated through fMAPLR.

4. Transcription system description

In this section the batch transcription system, which processes in block the input audio stream, is briefly described.

The input audio signal is first divided into homogeneous non overlapping segments using a start-end point detector (SEPD) and an acoustic classifier, based on GMMs. Word transcription is then generated in one decoding pass by using a 4-gram language model. When using speaker adaptively trained models, before decoding, the obtained labeled speech segments are grouped by a segment clustering method based on the BIC. The resulting segmentation and clustering is then exploited to perform cluster-wise feature normalization.

Acoustic models (AMs) were trained on about 223 hours of speech data obtained merging the Italian Broadcast News corpus with a corpus made of recordings of speeches delivered in the Italian Parliament. Three sets of HMMs were trained, in all cases AMs were state-tied, cross-word, speaker-independent triphone HMMs. Output probability distributions were modeled by mixtures of Gaussian probability density functions having diagonal covariance matrices. A phonetic decision tree was used for tying states and for defining the context-dependent allophones.

Each audio signal frame is parametrized into a 52-dimensional observation vector composed of 13 mel frequency cepstral coefficients plus their first, second and third order time derivatives. Cepstral mean subtraction (CMS) is performed on static features on a segment-by-segment basis.

For the first set of AMs, a projection of acoustic feature space, based on heteroscedastic linear discriminant analysis (HLDA), is applied to obtain 39-dimensional acoustic observations. AMs are then trained on these HLDA projected acoustic data through a conventional maximum likelihood procedure. This set of AMs is denoted in the following as *Baseline* AMs.

The second set of AMs, were speaker adaptively trained as described below [8, 14, 24]. A GMM with 1024 Gaussian components, having a diagonal covariance matrix, is first trained on the original 52-dimensional observation vectors. Acoustic observations in each, automatically determined, cluster of speech segments, are then normalized by applying an affine transformation estimated w. r. t. the GMM through fMLLR [11]. 3 fMLLR iterations were performed. After normalization of training data, an HLDA transformation is estimated and then applied to project the set of 52 normalized features into a 39-dimensional feature space. Triphone HMMs are trained on these normalized, HLDA projected, acoustic features through a conventional maximum likelihood procedure. This set of AMs is denoted in the following as *fMLLR* AMs.

The third set of AMs is trained in similar way as the second set but this time the number of automatically determined clusters of speech segments is limited to 32 as motivated in Section 2. As a result, during training only 32 fMLLR transformations were estimated, while for the second set of models more than 4,200 transformations were estimated. This third set of AMs is denoted in the following as *fMLLR_cl32* AMs.

A fourgram language model (LM) with a recognition vocabulary of 157K words was employed for decoding. The LM was trained using an Italian text corpus of 1.2G words mainly formed of texts from the news domain.

5. Experimental results

Recognition experiments were conducted on a set of 7 Italian news broadcasts having a total of about 35,900 transcribed words. Results of batch transcription experiments are reported in Table 1. The result obtained with the baseline system, 16.7% WER.

Table 1: *Recognition results (WER%) in batch mode.*

	Baseline	fMLLR	fMLLR_ 32cl
No Adaptation	16.7	-	-
1 fMLLR Iter	-	15.2	15.2
3 fMLLR Iter	-	14.5	14.7

When speaker adaptively trained models are used, cluster-based fMLLR adaptation is performed before decoding by estimating and applying an fMLLR transformation on each speech segment cluster as described in Section 2. Results show that a tangible improvement is achieved w. r. t. the baseline. When using acoustic models trained on training data grouped into only 32 speech segment clusters, performance slightly decrease w. r. t. the case in which no limit was imposed during training on the number of automatically determined speech segment clusters. From Table 1 we can also see that performing 3 fMLLR iterations during test, as done during training, leads to significant improvement w. r. t. performing a single iteration. In on-line adaptation, however, for computational efficiency, only a single fMLLR iteration is performed.

Table 2: *Recognition results (WER%) by the baseline system with two different segmentations and several CMS variants.*

	Cepstral Mean Subtraction			
	Segment		Window	
	Current	Prec.	Acausal	Causal
BatchSeg	16.7	19.7	16.9	17.7
SEPD	17.2	19.8	17.1	17.7

To measure the impact of the segmentation and CMS on performance we conducted a series of experiments by using the segmentation as obtained by the off-line audio partitioner (denoted as *BatchSeg*) and using the SEPD module alone. Results achieved using conventional segment based CMS are reported in the second column of Table 2. It can be noted that best performance, 16.7% WER, is achieved when the refined segmentation is used (row *BatchSeg*), while using the SEPD module alone (row *SEPD*) worsens recognition performance (17.2% WER). This may be due to the long spurious segments obtained with the SEPD module alone for which CMS is not well suited. In fact, the SEPD module is a fast energy-based start-end point detector that detects and removes silence regions, while the batch segmentation also exploits a classifier to split segments according to a male-speech/female-speech/noise/music classification. In both cases, however, cepstral mean is computed and subtracted on the whole current segment, so that the segment is available for decoding only after it is fully processed. This configuration is not suited for on-line real time processing. To cope with this problem we first conducted experiments performing CMS on the current segment by using the cepstral average estimated on the preceding segment. Results, reported in column *Segment Prec.* of Table 2, show a tangible decrease of performance. Then, we experimented with CMS relaxing the concept of segment and computing a cepstral mean for each incoming acoustic observation instead of for each segment. We explored two approaches. In the first approach, an incoming acoustic observation is normalized by subtracting a cepstral average computed on a causal window, 5 seconds long, of preceding acoustic observations (column *Window Causal*). In the second approach, an incoming acoustic observation is normalized by subtracting a cepstral average computed on an acausal window, 1.5 seconds long, spanning 1 second before and 0.5 seconds after the acoustic observation itself (column *Window Acausal*). From recognition results reported in Table 2 it can be seen that, at the cost of delaying the delivery of the transcription by 0.5 seconds, we can obtain recognition results aligned to that obtained with the segment based CMS, that is 17.1%WER vs. 17.2% WER with the SEPD segmentation. This result was improved by performing CMS during training of acoustic models in the same way as it is performed during testing, that is performing CMS based on the acausal window: a WER of 16.9% is achieved in this way to be compared with 17.1% in Table 2. The configuration leading to this result represents the set up for the on-line experiments discussed below. In addition, based on the above results with the baseline AMs, speaker adaptively trained models were also trained performing CMS based on the acausal window.

Table 3 reports recognition results obtained in on-line mode in which the SEPD module is in operation and a fixed delay by 0.5 seconds is introduced for performing CMS based on a running window spanning 1.5 seconds as specified above. Incremental speaker clustering and adaptation is performed on an audio chunk of a duration 1 or 2 seconds as described in Section 3. Reported results show that speaker adaptation is effective in improving the results of the baseline system: from 16.9% WER to 15.9% WER in case of a 1 second audio chunk and fMLLR_32cl AMs with smoothing of sufficient statistics. When using AMs trained without imposing any constraint on the number of speaker clusters during training (column *fMLLR*) a slightly worse result is obtained: 16.3% WER. This is in contrast with results reported in Table 1 for batch experiments. This is likely due to an existing bias in favor of fMLLR_32cl AMs in on-line adaptation as speaker clustering and initialization of fMLLR sufficient statistics for a new cluster are based on sufficient statistics and transformations corresponding to the 32 clusters of training data. Furthermore, we can note that varying the size of the audio chunk, from 1 to 2 seconds, does not lead to significant performance differences. Finally, the two techniques for robust estimate of fMLLR transformations with little data show to be equivalently effective as it results comparing performance reported in columns *Smoothing* and *fMAPLR* for the fMLLR_32cl AMs.

Table 3: *Recognition results (WER%) in on-line mode.*

	Baseline	fMLLR_32cl Smoothing	fMLLR_32cl fMAPLR	fMLLR Smoothing
No Adap.	16.9	-	-	-
1 sec	-	15.9	16.0	16.3
2 sec	-	15.9	15.9	16.3

6. Computational load

The whole adaptation process, including speaker clustering and incremental adaptation, on a single core of a multicore Intel Xeon 2.5Gz processor, takes 27% and 17% of real time when the input stream is processed in chunks of 1 second or 2 seconds, respectively. These numbers refer to usage of fMLLR_32cl models with smoothing of the sufficient statistics. The computational load introduced by the proposed on-line incremental adaptation approach is therefore compatible with the requirements of a real time transcription system, especially considering that nowadays multicore CPUs are commonly used.

7. Conclusions

We have proposed a novel adaptation technique for on-line incremental speaker clustering and adaptation, which is well suited for real time transcription applications in which both low adaptation latency and low computational load are required.

We have built upon our previously developed technique for text-independent on-line incremental adaptation which was previously tested only in the context of a telephonic application for which speaker change detection and clustering were not required [10]. To cope with speaker change detection and clustering needed for on-line incremental adaptation we relied on the approach proposed by Breslin at al. [6] in which individual speakers are represented by fMLLR transforms, and those transforms are used for both speaker clustering and adaptation. Our technique, however, differs for three main aspects from previous works [4, 5, 6]: a) fMLLR transforms are estimated w. r. t. a GMM avoiding the need of an estimated transcription of adaptation data. This leads to a text-independent approach fully decoupled from the decoding engine; b) speaker clustering as well as speaker adaptation are performed as the audio progresses, chunk by chunk, avoiding segmentation of the audio stream into utterance-like units and thus allowing a low adaptation latency; c) normalized acoustic data are available to the decoding process with a delay of only 0.5 seconds, needed for effective implementation of CMS.

The recognition results achieved in on-line transcription of television news broadcasts show that the proposed technique is effective resulting in 6% relative reduction in WER w. r. t. a non-adaptive baseline system.

8. Acknowledgements

This work was partially supported by the European project EU-BRIDGE, under the contract FP7-287658, funded by the European Commission under the Seventh Framework Programme for Research and Technological Development.

9. References

[1] G. Boulianne, J.-F. Beaumont, M. Boisvert, J. Brousseau, P. Cardinal, C. Chapdelaine, M. Comeau, P. Ouellet, and F. Osterrath, "Computer-assisted closed-captioning of live TV broadcasts in French," in *Proc. of INTERSPEECH*, 2006.

[2] H. Meinedo, M. Viveiros, and J. P. Neto, "Evaluation of a live broadcast news subtitling system for Portuguese," in *Proc. of INTERSPEECH*, 2008, pp. 508–511.

[3] S. Renals, T. Hain, and H. Bourlard, "Recognition and Understanding of Meetings: the AMI and AMIDA Projects," in *Proc. of ASRU Workshop 2007*, Kyoto, Japan, Dec. 2007.

[4] Z.-P. Zhang, S. Furui, and K. Ohtsuki, "On-line incremental speaker adaptation with automatic speaker change detection," in *Proc. of ICASSP*, Washington, DC, USA, 2000, pp. II961–II964.

[5] D. Liu, D. Kiecza, A. Srivastava, and F. Kubala, "Online speaker adaptation and tracking for real-time speech recognition," in *Proc. of INTERSPEECH*, 2005, pp. 281–284.

[6] C. Breslin, K. K. Chin, M. J. F. Gales, and K. Knill, "Integrated online speaker clustering and adaptation," in *Proc. of INTERSPEECH*, 2011, pp. 1085–1088.

[7] P. Brown, J. Spohrer, P. Hochschild, and J. Backer, "Partial traceback and dynamic programming," in *Proc. of ICASSP*, Paris, France, May 1982, pp. 1629–1632.

[8] G. Stemmer, F. Brugnara, and D. Giuliani, "Adaptive Training Using Simple Target Models," in *Proc. of ICASSP*, Philadelphia, PA, March 2005, pp. I–997–1000.

[9] D. Giuliani and F. Brugnara, "Experiments on Cross-System Acoustic Model Adapatation," in *Proc. of ASRU Workshop 2007*, Kyoto, Japan, Dec. 2007, pp. 117–122.

[10] D. Giuliani, R. Gretter, and F. Brugnara, "On-line speaker adaptation on telephony speech data with adaptively trained acoustic models," in *Proc. of ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 4385–4388.

[11] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[12] Y. Li, H. Erdogan, Y. Gao, and E. Marcheret, "Incremental online feature space MLLR adaptation for telephony speech recognition," in *Proc. of ICSLP*, Denver, Colorado, Sep. 2002, pp. 1417–1420.

[13] D. Giuliani, M. Gerosa, and F. Brugnara, "Speaker Normalization through Constrained MLLR Based Transforms," in *Proc. of INTERSPEECH*, Jeju Island, Korea, Oct. 2004, pp. 2893–2897.

[14] D. Giuliani, M. Gerosa, and F. Brugnara, "Improved automatic speech recognition through speaker normalization," *Computer Speech and Language*, vol. 20, pp. 107–123, 2006.

[15] P. Kenny, V. Gupta, G. Boulianne, P. Ouellet, and P. Dumouchel, "Feature Normalization Using Smoothed Mixture Transformations," in *Proc. of INTERSPEECH*, Pittsburgh, PA, Sept. 2006, pp. 17–21.

[16] S. S. Chen and P. S. Gopalakrishnan, "Speaker, Environment and Channel Change Detection and Clustering Via the Bayesian Information Criterion," in *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[17] D. Kim, S. Umesh, M. Gales, T. Hain, and P. Woodland, "Using VTLN for Broadcast News Transcription," in *Proc. of INTERSPEECH*, Jeju Island, Korea, Oct. 2004.

[18] P. Akhil, S. Rath, S. Umesh, and D. Sanand, "A computationally efficient approach to warp factor estimation in VTLN using EM algorithm and sufficient statistics," in *Proc. of INTERSPEECH*, Brisbane, Australia, Sept. 2008, pp. 1713–1716.

[19] D. Liu and F. Kubala, "Online speaker clustering," in *Proc. of ICASSP*, 2003, pp. I–573–575.

[20] C. Breslin, H. Xu, K. Chin, M. Gales, and K. Knill, "Prior Information for Rapid Speaker Adaptation," in *Proc. of INTERSPEECH*, Chiba, Japan, Sept. 2010, pp. 1644–1647.

[21] X. Lei, J. Hamaker, and X. He, "Robust Feature Space Adaptation for Telephony Speech Recognition," in *Proc. of INTERSPEECH*, Pittsburgh, PA, Sept. 2006, pp. 773–766.

[22] K. Visweswariah, V. Goel, and R. Gopinath, "Structuring linear transforms for adaptation using training time information," in *Proc. of ICASSP*, Orlando, Florida, May 2002.

[23] X. Cui, J. Xue, and B. Zhou, "Improving Online Incremental Speaker Adaptation with Eigen Feature Space MLLR," in *Proc. of ASRU Workshop*, Merano, Italy, Dec. 2009, pp. 136 – 140.

[24] G. Stemmer and F. Brugnara, "Integration of Heteroscedastic Linear Discriminant Analysis (HLDA) into Adaptive Training," in *Proc. of ICASSP*, Toulouse, France, May 2006, pp. I–1185–1188.