STREAM-BASED SPEAKER SEGMENTATION USING SPEAKER FACTORS AND EIGENVOICES

Fabio Castaldo^, Daniele Colibro, Emanuele Dalmasso^, Pietro Laface^ , Claudio Vair**
Politecnico di Torino, Italy^
{Fabio.Castaldo,Emanuele.Dalmasso,Pietro.Laface}@polito.it
Loquendo, Torino, Italy*
{Daniele.Colibro,Claudio.Vair}@loquendo.com

ABSTRACT

This paper presents a stream-based approach for unsupervised multi-speaker conversational speech segmentation.
The main idea of this work is to exploit prior knowledge about the speaker space to find a low dimensional vector of speaker factors that summarize the salient speaker characteristics.
This new approach produces segmentation error rates that are better than the state of the art ones reported in our previous work on the segmentation task in the NIST 2000 Speaker Recognition Evaluation (SRE). We also show how the performance of a speaker recognition system in the core test of the 2006 NIST SRE is affected, comparing the results obtained using single speaker and automatically segmented test data.

Index Terms— Speaker modeling, speaker segmentation, speaker factors, eigenvoices, speaker clustering

1. INTRODUCTION

Speaker recognition often requires a pre-segmentation step to detect the regions in a conversation corresponding to a putative single speaker. Clustering these regions is a reasonable approach when the speaker turns are frequent and the duration of a turn possibly short, as happens in the CallHome Corpus collected by the Linguistic Data Consortium that was used in the 2000 NIST speaker recognition evaluation [1].

This paper presents a stream-based approach for unsupervised multi-speaker conversational speech segmentation. It finds the number of speakers in a conversation and produces its segmentation results with a fixed latency compared to a real time speech audio stream.

In stream-based speaker segmentation there are at least three problems to be solved. First, Gaussian Mixture Models (GMMs) are commonly used for speaker segmentation, but short speaker turns do not allow reliable speaker models to be estimated [2]. Second, the detection of the conversation turns and the estimation of the speaker models require low complexity to cope with audio streaming. Two main approaches are used for this task. Speaker changes are detected using a sliding variable length analysis window and the Bayesian Information Criterion (BIC), as done for example in [3-4]. Alternatively, a preliminary blind segmentation is performed by analyzing signal slices of fixed length, as proposed for example in [2,5-7]. Finally, a good distance measure between models is necessary to decide the number of speakers in a conversation. Again the BIC criterion or the Cross Likelihood Ratio [8] can be used for speaker clustering.

The contribution of this work is an original solution for the first two problems. We draw on earlier work on eigenvoice speaker adaptation [9] and identification [10]. In particular, the main idea of this work is to exploit prior knowledge about the speaker space to find a low dimensional vector of speaker factors that summarize the salient speaker characteristics. These factors can be computed effectively using a small sliding window, and do not suffer the problem of data sparseness. We use the speaker factors to perform a preliminary segmentation step, and to estimate speaker models by constraining them in a previously estimated linear subspace. These constraints make it possible to estimate a reliable model even with small amounts of data [10].

Our approach performs a preliminary blind segmentation analyzing signal slices of fixed length. The length of a slice is chosen assuming that up to three speakers are present in every slice, and most frequently only two. The models of the putative speakers in the slice are rapidly synthesized using the estimated speaker factors. A speaker model estimated on the current slice becomes a new global model if it is far enough from the other global models. Otherwise, the slice frames related to the model contribute to the update of the nearest speaker global model.

This new technique produces segmentation error rates that are better than the state of the art ones reported in our previous work on the segmentation task in the NIST 2000 Speaker Recognition Evaluation. We also show that good results are obtained in the 2006 NIST multi-speaker conversation tests, where we compare the verification performance using automatically segmented training data with the one obtained using single speaker data.

The paper is organized as follows: In Section 2 we briefly recall the eigenvoice approach. Section 3 describes the first step of our technique, based on the computation of speaker factors using a sliding window on an audio slice. Section 4 illustrates the segmentation step inside each conversation slice. Section 5 gives details of the process that clusters the speakers models detected in a slice with the speaker models already found in the audio stream. Experimental results are presented in Section 6 and some concluding remarks are given in Section 7.

2. EIGENVOICES

Our speaker models are GMMs estimated from a common GMM root model, the so-called Universal Background Model (UBM) [2]. The models are trained by adapting only the Gaussian means, and share with the other speaker models the remaining UBM parameters. A supervector that includes all the speaker specific parameters is simply obtained by appending the adapted mean value of all the Gaussians in a single stream. The same procedure allows the UBM supervector to be obtained.

The idea behind the eigenvoice approach proposed in [9] for speaker adaptation is that a small number of basis vectors (the eigenvoices vectors) can be obtained offline by Principal Component Analysis from a large set of reference speakers, and that a small number of parameters in this subspace can summarize the speaker characteristics in the large supervector space. These parameters are the speaker factors. A supervector for a new speaker $\boldsymbol{\mu}_s$ is modeled by a linear combination of the eigenvoices $\mathbf{e}_n$ *(n = 1,...N)* through the speaker factors according to:

$$\boldsymbol{\mu}_s = \boldsymbol{\mu}_{UBM} + \mathbf{E}_s \times \mathbf{x}_s \tag{1}$$

where $\boldsymbol{\mu}_{\mathrm{UBM}}$ is the UBM supervector. $\mathbf{E}_s$ a low rank matrix, including the *N* eigenvoice $\mathbf{e}_n$ vectors corresponding to the largest eigenvalues. Matrix $\mathbf{E}_s$ allows projecting the speaker factors subspace in the supervector domain. The *N*-dimensional vector $\mathbf{x}_s$ holds the speaker factors for the current speaker utterance. It is estimated to maximize the probability of the speaker model $\boldsymbol{\mu}_s$, given the observed data. The main advantage of such an approach is that the number of parameters that it is necessary to estimate is small. This allows robust models to be obtained using a small number of observations.

In the experiments described in this paper, the UBM and the speaker GMMs consist of mixtures of 256 Gaussians, the observation vector includes 13 Mel frequency cepstral coefficients and their first derivatives, and the number of eigenvoices is limited to *N*=20. The eigenvoices were obtained performing Principal Component Analysis on a set of 1433 female and 1183 male speaker models, estimated on data coming from the multilingual Callfriend database, and from the Italian, Swedish, and Brasilian Portuguese SpeechDat corpora.

3. SLICE PRE-SEGMENTATION

Since the UBM and the factor loading matrix $\mathbf{E}_s$ describing the speaker subspace are computed a priori, it is possible to characterize a speaker using only its speaker factors $\mathbf{x}_s$. For the sake of efficiency, in the pre-segmentation step we make use of the information summarized in $\mathbf{x}_s$ directly, rather than exploiting it to synthesize a speaker supervector.

As in our previous approach, we perform a preliminary blind segmentation by analyzing successive audio slices of fixed length. The best results in our experiments were obtained by setting the size of the audio slice to 60 seconds.

The first step of our technique computes the speaker factors for each frame of the audio slice using a sliding window. The estimation of the factor $\mathbf{x}(t)$ is performed on a symmetric window centered on frame *t*. The size of the window depends on a trade-off. The use of a large window increases the stability of the speaker factors, but also the probability of incorporating frames of different speakers. In our experiments the window size has been set to 100 frames. The speaker factors are estimated to maximize the probability of the observations with respect to the subspace represented by matrix $\mathbf{E}_s$ as follows [9]:

$$\mathbf{x}_s = \mathbf{A}^{-1} \cdot \mathbf{b} \tag{2}$$

where the elements of matrix **A** and vector **b** are:

$$a_{k,j} = \sum_{m=1}^{M} \left(\sum_{t=1}^{T} \gamma_m(o_t) \right) \cdot \frac{\mathbf{E}'_{k,m} \cdot \mathbf{E}_{j,m}}{\boldsymbol{\sigma}_m^2} \tag{3}$$

$$b_k = \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(o_t) \cdot \frac{\mathbf{E}'_{k,m} \cdot (\mathbf{o}_t - \boldsymbol{\mu}_m)}{\boldsymbol{\sigma}_m^2} \tag{4}$$

In these equations, T is the number of observation frames, *M* is the number of Gaussians in the supervector, $\boldsymbol{\mu}_m$ and $\boldsymbol{\sigma}_m$ are the mean and the diagonal covariance of the UBM respectively, and $\gamma_m(o_t)$ represents the posterior probability of Gaussian *m* at time *t* given the compete observation sequence.
The speaker factors are computed incrementally, by removing the leftmost frame of the window and adding a new one to its right.
The complexity of the process is thus limited to a small (*N*x*N*) matrix inversion and a matrix (*N*x*N*) by vector (*N*x*1*) product for each frame.

Figure 1 shows the first two parameters of the speaker factors computed with a sliding window of 100 frames on the first 60 seconds of file `haap.wav` in the NIST 2000 Speaker Recognition Evaluation set. This slice includes the first part of the conversation between a female and a male speaker. In Figure 1(a) the color of the points, and the line width, are associated to the reference labels. It can be observed that the values of the speaker factors for a reference speaker are spread in a relatively large region rather than being centered on a single point. Due to the small size of the analysis window (1 second only), the speaker factor values are affected by the phonetic content of the window. However, the cluster of points related to a speaker is well separated with respect to the second speaker cluster. Thus, a simple Euclidean distance among speaker vectors can be used as an approximation of a distance between supervectors such as the Kullback-Leibler divergence [11].

In this step we perform a preliminary segmentation assuming the presence of two speakers in a slice. The set of 6000 speaker factor vectors estimated for the current slice is used to estimate the two *N*-dimensional Gaussians that best fit the data. These Gaussians represent the models of two putative speakers in the speaker factors space. The color and line width in Figure 1(b) identify the cluster of points belonging to the Gaussians estimated on the same slice of Figure 1(a). A simple 2-state HMM using the estimated Gaussian models, with 0.9 and 0.1 self and transition probabilities respectively, allows the segmentation of each conversation slice to be performed by means of the Viterbi algorithm. Figure 1(c) shows the points associated with the putative speaker labels selected by the Viterbi segmentation process.

4. SLICE SEGMENTATION

The second step of the algorithm aims at verifying the accuracy of the two speaker segmentation hypothesis. We compute a unique speaker factor vector **x** for all the slice frames, and two other speaker factors vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$, using the frames assigned to each putative speaker by the previous segmentation step. Each speaker factor vector is projected back to the supervector model space by the eigenvoice matrix **E** using (1), to rapidly synthesize two speaker models and a "slice model". Thus, our approach exploits the a priori information given by matrix **E** to create a speaker model from a small number of observation frames. We then compute the log-likelihood ratio score

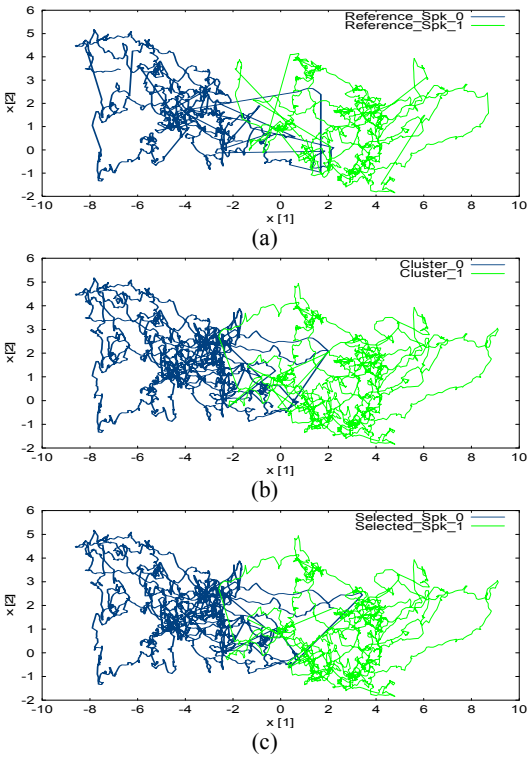

Figure 1*: Plot of the first two parameters of the speaker factors computed with a sliding window of 100 frames on the first 60 second slice of file* `haap.waw` *in the NIST 2000 Speaker Recognition Evaluation set. The colors, and line width, identify the initial clusters (b), and the speakers using the reference labels (a) and the segmentation labels (c).*

Table 1: Segmentation errors on the SwitchBoard 2-speaker segmentation task

Segment speakers	Segmentation Errors	
	Previous approach	Speaker factor approach
Female	11.4 %	8.4 %
Male	7.0 %	6.8 %
Mixed	4.9 %	4.2 %
Total	7.6 %	6.2 %

Table 2**:** *Segmentation errors on CallHome data*

N. of speakers	2	3	4	5	6	7
Errors (%)	8.7	15.7	15.1	20.2	25.5	29.8

$$D_{12} = \frac{L_1 + L_2 - L}{n_1 + n_2} \quad (5)$$

where L_i is log-likelihood of speaker model i computed on the n_i frames assigned to cluster i , and L is the log-likelihood of all the frames in the slice computed with the slice model.

If score D_{12}, measuring the distance between the two models, is less than a threshold, the entire slice is assigned to a single speaker. Otherwise, at least 2 speakers are present in the slice, and we check for a 3-speaker hypothesis. The pre-segmentation process is performed again to find 3 speaker factor Gaussian models. A new Viterbi segmentation is obtained, and the log-likelihood ratio score is computed using the three models according to:

$$D_{123} = \frac{L_{1*} + L_{2*} + L_{3*} - L}{n_{1*} + n_{2*} + n_{3*}} \quad (6)$$

where the log likelihood L_{i*} and the segment length n_{i*} refer to the new 3 speaker segmentation.
The final decision to assign either two or three speakers to the slice is taken based on the difference between D_{12} and D_{123}.

5. STREAM PROCESSING

In the last step, the speaker models estimated in the current slice are compared with the models of the speakers detected in previous slices of the conversation. In particular, the slice segmentation process generates a temporal segmentation and a GMM model for each speaker detected in the current slice. In the streaming approach, the slice frames are no more used for segmentation purposes, only the models of speakers detected in the previously processed slices – the global models - are kept.

Since the global speaker models are adapted from the UBM models, they can be effectively compared with the models of the speaker detected in the current slice by using the Kullback-Leibler divergence as distance metric [11].

For the sake of efficiency, the segmentation results of the current slice are not modified using the information provided by the global speaker models. We have experimentally verified, however, that a simple method to get slice and global speaker models better aligned consists in using larger overlapping slices. Using overlapping slices allows obtaining more stable results on segmentation tasks involving more than two speakers, while still coping with the streaming constraints.

In our experiments, the slice size is set to 90 seconds, but only the results for the first 60 seconds of each slice are retained. The next audio slice is obtained by appending the next 60 seconds of the conversation to the last 30 seconds of the previous slice. The segmentation is performed on this new slice, and the obtained slice speaker models are compared with the global models. A slice speaker becomes a new global model if no global model can be found that is similar to it, according to the Kullback-Leibler distance measure. Otherwise, the nearest global model is updated using the slice speaker model by means of MAP adaptation of the means only.

6. EXPERIMENTAL RESULTS

We report the results of our approach on the speaker segmentation tasks of the NIST 2000 evaluation both on the Switchboard and on the multilingual CallHome data [1]. All scores have been obtained using the scoring script *seg_scoring.v2.1.pl* provided by NIST, ignoring collar periods of 250 ms, as is usual for these tests. The segmentation was performed without knowing the number of speakers taking part in the conversations.
Table 1 compares the segmentation error rates obtained on the SwitchBoard 2-speaker segmentation task using our previous approach, described in [7], and using the new approach based on

speaker factors. A reduction in the error rates of 18% is obtained by the streamed segmentation approach, which takes on average about 6% of the real time, on a Pentium Mobile 2.13 GHz system. These results are in line with to the ones reported in [1] for one of the best system participating in the NIST 2000 evaluation.

The performance of our approach has also been assessed in the CallHome n-speaker segmentation subtask. Again our results, shown in Table 2, are comparable with the one presented in [1].

6.1. Speaker recognition experiments

We assessed the quality of our speaker clustering procedure in the NIST 2006 Speaker Recognition Evaluation core test involving two wire (2w) recordings. In the four wire (4w) condition, each audio file in the enrollment and verification lists includes a single side of a conversation, i.e. the voice of a single speaker, whereas in the 2w condition a whole conversation between two speakers is supplied as a training or test audio file. We report here the results referring to the 4w training and the 2w testing condition. This test has the goal of producing the likelihood that one of the two speakers involved in a conversation is the target speaker.

To obtain a one-to-one comparison with the 4w test condition, we defined a new "unofficial" 2w test, by summing the two sides of all the recordings in the list of the core NIST 2006 1conv4w test. We performed the tests after preprocessing the summed audio files using our speaker segmentation approach that produces two audio tracks, each containing the voice of a single unknown speaker. Both audio tracks produced after segmentation are scored against the target model, and the best score is produced as the matching result. The bottom DET curve in Figure 2 refers to the results produced by the combination of a GMM and of a Phonetic GMM systems described in [12] on the 4w core test condition. The upper DET curve shows the performance obtained in this unofficial 2w test, including the effect of automatic segmentation. It is worth noting that a decision based on the best score of the two sides, even neglecting the segmentation errors, will produce less accurate results compared with the corresponding 4w tests, due to the increase of the probability of false alarms (FA). This is demonstrated by the results - represented by the middle DET curve in Figure 2 - obtained in the same tests, without segmentation, but taking into account both sides of the conversation and applying the best matching decision rule used for the 2w tests. This result highlights the impact of the anticipated increase of the FAs on system accuracy, even in the absence of automatic speaker segmentation.

Comparing the DET curves, it is interesting to observe that the main source of accuracy degradation is not the segmentation procedure, but the presence of both sides of the conversation in the trials. Further degradation of the results is due to the occurrence of overlapped speech in the summed 2w signals. Overlapped speech does not occur in the 4 wire condition.

Although the segmentation obtained with the new approach is more precise, the speaker recognition performance improvement is not statistically significant compared with our previous segmentation approach.

7. CONCLUSIONS

A new approach to unsupervised multi-speaker conversational speech segmentation has been presented. It has demonstrated its capability of detecting and segmenting conversations with unknown number of speakers. It exploits the a priori information given by a set of eigenvoices, estimated offline, to rapidly estimate

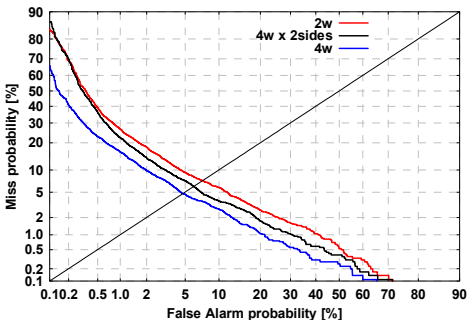

Figure 2: *DET plot comparing the results of the 4w condition with the 4w but best of the two side condition (see text) and the segmented 2w condition. The legend labels are ordered by decreasing EER.*

a set of speaker factors on a sliding window, and to create reliable speaker model from a small number of observation frames. Good results have been presented for multi-speaker segmentation and verification of conversational speech using standard data and tools. The speaker recognition tests on the 2w condition show that the gap between the DET curves of the 4w and 2w test conditions is small. Moreover, automatic segmentation accounts for about a half of the system accuracy degradation, the other half can be imputed to the detection of a speaker on the 2 sides of a conversation.

8. REFERENCES

[1] A. Martin, and M.A. Przybocki, "Speaker recognition in a Multi-Speaker Environment," Proc. Eurospeech, Vol. 2, pp. 787–790, 2001.

[2] R.B. Dunn, D. Reynolds, and T. Quatieri, "Approaches to Speaker Detection and Tracking in Conversational Speech," Digital Signal Processing, vol. 10, pp. 93-112, 2000.

[3] S.S Chen, and P.S. Gopalakrishnan, "Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion," Proc. Darpa News transcription and Understanding Workshop, Vol. 6, pp. 127-132, 1998.

[4] P. Delacourt, and C. Wellekens, "DISTBIC: a speaker- based segmentation for audio data indexing," Speech Communication, Vol. 32, No. 1-2, pp. 111-126, 2000.

[5] L. Wilcox, F. Chen, D. Kimber, and V. Balasubramanian, "Segmentation of Speech Using Speaker Identification," Proc. of ICASSP, pp. I.161- I.164, 1994.

[6] S. Meignier, J.F. Bonastre, and S. Igounet, "E-HMM approach for learning and adapting sound models for speaker indexing," Proc. 2001: a Speaker Odyssey, pp. 175-180, 2001.

[7] E. Dalmasso, P. Laface, D. Colibro, C. Vair, "Unsupervised Segmentation and Verification of Multi-Speaker Conversational Speech", Proc. Interspeech 2005, pp. , 2005.

[8] C. Barras, X. Zhu, S. Meignier, J. L. Gauvain, "Improving Speaker Diarization," in Proc. DARPA RT04, 2004.

[9] R. Kuhn, J.-C. Junqua, P. Nguyen, N. Niedzielski, "Rapid Speaker Adaptation in Eigenvoice Space". IEEE Trans. Speech and Audio Prcocessing, Vol. 8, n. 6, pp.695-707, 2000.

[10] O. Thyes, R. Kuhn, P. Nguyen, J.-C. Junqua, "Speaker Identification and Verification Using Eigenvoices", Proc. ICSLP 2000, Vol.2, pp. 242-245, 2000.

[11] M.N. Do, "Fast Approximation of Kullback-Leibler Distance for Dependence Trees and Hidden Markov Models", IEEE Signal Processing Letters, Vol. 10, n. 4, pp. 115-118, 2003.

[12] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, C. Vair, "Compensation of Nuisance Factors for Speaker and Language Recognition", IEEE Trans. on Audio, Speech, and Language Processing. Vol. 15-7, pp. 1969-1978, 2007.