

# SPEAKER TRACKING IN BROADCAST AUDIO MATERIAL IN THE FRAMEWORK OF THE THISL PROJECT

Laurent Couvreur and Jean-Marc Boite

Faculté Polytechnique de Mons, TCTS Lab  
Boulevard Dolez 31  
B-7000 Mons (Belgium)  
Email : {lcouv,boite}@tcts.fpms.ac.be

## ABSTRACT

In this paper, we present a first approach to build an automatic system for broadcast news speaker-based segmentation. Based on a *Chop-and-Recluster* method, this system is developed in the framework of the THISL project. A metric-based segmentation is used for the *Chop* procedure and different distances have been investigated. The *Recluster* procedure relies on a *bottom-up* clustering of segments obtained beforehand and represented by non-parametric models. Various hierarchical clustering schemes have been tested. Some experiments on BBC broadcast news recordings show that the system can detect real speaker changes with high accuracy (mean error  $\simeq 0.7s$ ) and fair false alarm rate (mean false alarm rate  $\simeq 5.5\%$ ). The *Recluster* procedure can produce homogeneous clusters but it is not already robust enough to tackle too complex classification tasks.

## 1. INTRODUCTION

THISL (THematic Indexing of Spoken Language)<sup>1</sup> is an ESPRIT Long Term Research project that is investigating the development of a news-on-demand system using speech recognition, natural language processing and text retrieval. The main goal is to build a system for a BBC newsroom application : broadcast news recordings are daily recognized and transcriptions are indexed for later information retrieval [1, 2]. According to the requirements of the BBC, the main industrial advisor of the project, the retrieval engine should be able to return the identity of the speakers within the sections which correspond to any query submitted to the system. Thus, the input audio stream that is recognized should be also segmented according to speaker identity.

Automatic segmentation of an audio stream is frequently introduced as an efficient method to improve performances of adaptive speech recognizers. The problem of acoustic segmentation has been often addressed for a few years [7, 8, 11, 5, 6, 9] : it con-

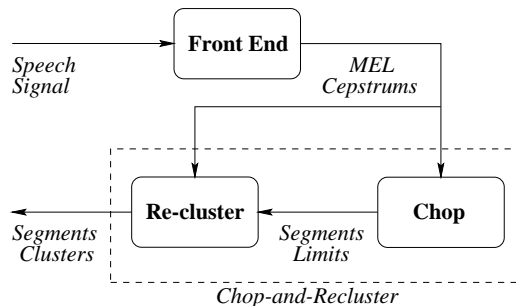


Figure 1: *Chop-and-Recluster* algorithm for speaker tracking in audio material.

sists in clustering audio segments into homogeneous clusters according to speaker identity, background conditions (e.g., clean speech or noisy speech) or channel conditions (e.g., microphone speech or telephone speech). In the framework of the THISL project, we are primarily concerned by identifying each speaker individually every time he/she speaks within a program. Speaker tracking can broadly be defined as segmenting and labelling a spoken audio stream associated with an unknown number of unknown speakers into homogeneous regions according to speaker identity. In practice, “speaker” means a speaker with constant background and channel conditions, because acoustic changes do not necessarily correspond to speaker changes. If acoustic changes in broadcast news programmes (e.g., change from microphone speech to telephone speech, change from clean speech to noisy speech) may be clues of speaker changes, they can also be very confusing for a segmentation algorithm based on acoustic features. On the other hand, it is hard to discriminate between two speakers if their acoustic features are too similar.

Various schemes have already been proposed to perform automatic segmentation of an audio stream according to speaker identity. These algorithms are generally based on a *Chop* or *splitting* procedure followed by a *Recluster* or *merging* procedure. There are many approaches to detect acoustic speaker changes, as reported in [6] :

This work was supported by the ESPRIT Long Term Research Project THISL (23495).

<sup>1</sup><http://www.dcs.shef.ac.uk/research/groups/spandh/projects/thisl/>

**Decoder-based splitting.** The segments boundaries are set according to information provided by a recognizer which decodes the spoken audio stream at first (e.g., possible speaker changes are at every silence locations).

**Model-based splitting.** If speaker models are trained beforehand for every speaker, the audio stream can be parsed in terms of these models. Speaker changes are identified as model decision changes.

**Metric-based splitting.** The speaker changes are found at maxima of a distance signal measured between two contiguous windows shifted along the speech signal.

The first method is really unadapted to our problem, while the second is unpractical since no speaker models are available a priori. Thus, we decided to implement the third one. An acoustic distance is computed for every pair of windows along the speech signal. The peaks are detected thanks to a two thresholds criterion : one threshold for minimum peak value and another to avoid too close changes.

Once segments are available, they are grouped into clusters whose members are hopefully as similar as possible. As described in [8], many clustering algorithms may be implemented, varying from agglomerative methods to divisive methods. Only results for agglomerative methods are presented in this paper. One major problem of all these clustering schemes is the stop criterion. For the THISL problem, it is assumed that the number of speakers is known for each broadcast news programme. Thus, the merging process is repeated until the number of groups is equal to the known number of speakers. Then, the elements of such a partition can be hand-labelled with the true name of the speakers and this information could enrich the indexed database.

This paper is organized as follows : in section 2, we describe the segmentation algorithm (i.e., the *Chop* or *splitting* procedure and the *Recluster* or *merging* procedure); in section 3, some results are given for experiments performed to show the effectiveness of both procedures for radio broadcast news bulletins provided by BBC. A full integrated system is tested and performances are discussed. In section 4, we compare our system with previous works and propose some possible improvements to our first approach in order to achieve more efficiency and robustness.

## 2. PROPOSED ALGORITHM

The algorithm proposed in this paper rests on a *metric-based* splitting technique, which was considered as the easiest one to implement in our first approach to the speaker tracking problem. After splitting, a hierarchical clustering, also called *bottom-up* clustering is used to merge similar segments into desired clusters.

As depicted in Figure 1, both procedures are based on acoustic features, which are Mel-warped cepstral features.

### 2.1. Splitting Procedure

First, the speech signal is expressed in terms of acoustic feature vectors. As classically for metric-based splitting method, the speaker changes are found at maxima of an acoustic distance signal measured between two neighboring windows which are shifted along the speech signal. Consider the two collections of feature vectors  $V^l = \{v_k^l\}_{k=1,\dots,K}$  and  $V^r = \{v_{k'}^r\}_{k'=1,\dots,K'}$  containing respectively the feature vectors from the left window and the right window around the discrete time where the distance is measured. As previously proposed in [4], a *K-means* clustering is first performed over each vector collection. Members of each cluster are assumed to be drawn from a multivariate Gaussian distribution :  $V_i^l \sim N(\bar{\mu}_i^l, \Sigma_i^l)$  and  $V_j^r \sim N(\bar{\mu}_j^r, \Sigma_j^r)$  where  $i$  and  $j$  are the cluster indexes for each window. After estimating the mean column vector  $(\bar{\mu}_i^l, \bar{\mu}_j^r)$  and the full covariance matrix  $(\Sigma_i^l, \Sigma_j^r)$  for every cluster in both windows, the acoustic distance between the two collections of distributions is computed as follows :

$$D_S^{lr} = \frac{d^{lr}}{d} \cdot \frac{\max_{i,j} d_{ij}^{lr}}{\min_{i,j} d_{ij}^{lr}} \quad (1)$$

where  $d_{ij}^{lr}$ ,  $d^{lr}$  and  $d$  denote respectively a cluster distance between the Gaussian cluster  $i$  from the left window and the Gaussian cluster  $j$  from the right window, a cluster distance based on a single Gaussian cluster for each window, and the mean of the previously defined distance over all pairs of windows. Actually, the so-called cluster distance is a distance between two n-dimensional Gaussian distributions. Several cluster distances have been tested like in [3] : the Kullback-Leibler (2) and the Bhattacharyya (3) distances if full covariance matrices are assumed, or the Mahalanobis (4), the Euclidian (5) and the L2 (6) distances if diagonal matrices are assumed. The distance  $d^{l2}$  between two Gaussian distributions  $N1(\bar{\mu}_1, \Sigma_1)$  and  $N2(\bar{\mu}_2, \Sigma_2)$  may be expressed as :

$$d_{KL}^{l2} = \frac{1}{2}(\bar{\mu}_2 - \bar{\mu}_1)^T (\Sigma_2^{-1} + \Sigma_1^{-1})(\bar{\mu}_2 - \bar{\mu}_1) + \frac{1}{2}tr(\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 - 2I) \quad (2)$$

$$d_{BHA}^{l2} = \frac{1}{4}(\bar{\mu}_2 - \bar{\mu}_1)^T (\Sigma_2 + \Sigma_1)^{-1}(\bar{\mu}_2 - \bar{\mu}_1) + \frac{1}{2} \log \frac{|\Sigma_1 + \Sigma_2|}{2\sqrt{|\Sigma_1 \Sigma_2|}} \quad (3)$$

$$d_{MAH}^{l2} = \frac{1}{n}(\bar{\mu}_2 - \bar{\mu}_1)^T (\Sigma_2 \Sigma_1)^{-1}(\bar{\mu}_2 - \bar{\mu}_1) = \frac{1}{n} \sum_{k=1}^n \frac{(\mu_{2k} - \mu_{1k})^2}{\sigma_{1k} \sigma_{2k}} \quad (4)$$

$$\begin{aligned}
d_{EUC}^{l_2} &= (\bar{\mu}_2 - \bar{\mu}_1)^T (\bar{\mu}_2 - \bar{\mu}_1) \\
&= \sum_{k=1}^n (\mu_{2k} - \mu_{1k})^2 \quad (5)
\end{aligned}$$

$$d_{L_2}^{l_2} = \sqrt{\int_{\mathbb{R}^n} [N_2(\bar{\mu}_2, \Sigma_2) - N_1(\bar{\mu}_1, \Sigma_1)]^2 d\bar{X}} \quad (6)$$

A closed-form expression exists for the distance given by (6) if Gaussian distributions are assumed. Namely,

$$\begin{aligned}
(d_{L_2}^{l_2})^2 &= \prod_{k=1}^n \frac{1}{2\sigma_{1k}\sqrt{\pi}} + \prod_{k=1}^n \frac{1}{2\sigma_{2k}\sqrt{\pi}} \\
&+ 2 \prod_{k=1}^n \frac{1}{\sqrt{2\pi}\sqrt{\sigma_{1k}^2 + \sigma_{2k}^2}} \exp \left[ \frac{1}{2} \frac{\left( \mu_{1k} \frac{\sigma_{2k}}{\sigma_{1k}} + \mu_{2k} \frac{\sigma_{1k}}{\sigma_{2k}} \right)^2}{\sigma_{1k}^2 + \sigma_{2k}^2} - \left( \frac{\mu_{1k}^2}{\sigma_{1k}^2} + \frac{\mu_{2k}^2}{\sigma_{2k}^2} \right) \right] \quad (7)
\end{aligned}$$

Once the acoustic distance (1) has been computed for every pair of windows, it is processed to find relevant maxima according to a two thresholds criterion. The first threshold corresponds to the minimum value for detecting a peak. The second threshold guarantees a minimum delay between two consecutive changes and allows to eliminate adjacent changes that are too close : only peaks corresponding to the highest values are kept while all other peaks within the threshold around these maxima are discarded.

To justify the use of *K-means* clustering over each window before estimating the acoustic distance, one may suggest that the clustering will increase the discriminant power of the distance. For example (Figure 2), we observe that the existing peaks are strengthened if the method with *K-means* clustering is used. Moreover, some peaks at no-change locations are less disturbing for the peak detector. Unfortunately, clustering the feature vectors before computing the acoustic distance is very time-consuming. This computational burden depends on the number of clusters which should be chosen judiciously. In order to fasten this clustering, the centroids of the current analysis window are initialized with the centroids of the previous analysis window. Working like this, we hope that strong adaptation (i.e., CPU-consuming adaptation) of the centroids is needed only when there is a significant acoustic change.

## 2.2. Merging Procedure

Once an audio track has been chopped into speaker-constant segments, a hierarchical clustering is performed over the segments  $\mathcal{S} = \{s_1, s_2, \dots, s_L\}$  which are described by their acoustic features. At each step of this agglomerative clustering, the two nearest clusters are merged and a  $P$ -clusters partition of  $\mathcal{S}$  is obtained :  $\mathcal{P} = \{c_1, c_2, \dots, c_P\}$ . It starts with  $P = L$  and  $c_l = s_l, \forall l$ , and terminates with  $P = 1$  and  $c_1 = \cup_{l=1}^L s_l$ .

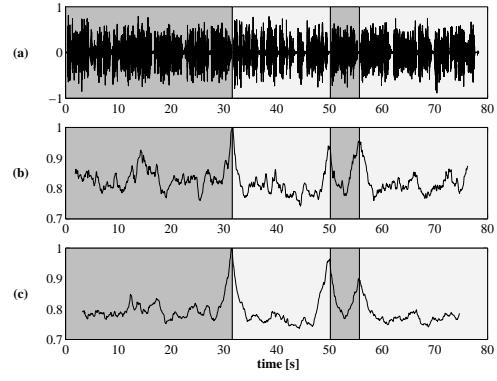


Figure 2: (a) Acoustic signal, (b) acoustic distance with 1-cluster windows and (c) acoustic distance with 3-clusters windows. In both case, a Bhattacharyya cluster distance is used and window length, window overlap and window shift are set respectively to 2.5s, 1s and 50ms. There are 3 real speaker changes, the first one corresponds to a change from microphone speech to telephone speech.

Remind that the number of speakers  $N$  in each programme is assumed to be known. Thus, the clustering is stopped once a  $N$ -clusters partition of the segments is reached (i.e., for  $P = N$ ). One may hope that each cluster ideally contains segments from one and only one speaker. To apply such a clustering, it is necessary to define an agglomerative scheme and a distance between two segments. First, each segment  $s_i$  is replaced by a non-parametric model (i.e., a codebook)  $\mathbf{m}_i$  estimated once again by *K-means* clustering of the feature vectors associated to this segment. The segment codebook is initialized with one centroid, the mean feature vector for the current segment. This first centroid is split into two new centroids and *K-means* clustering is performed. After convergence, the centroids are binary split and the *K-means* is applied once again. The process continues until the desired number of centroids is reached and the clustering has converged. The distance between the two segments  $s_{i_1}$  and  $s_{i_2}$ , given by their codebooks  $\mathbf{m}_{i_1} = \{m_w^{l_1}\}_{w=1, \dots, W_{i_1}}$  and  $\mathbf{m}_{i_2} = \{m_w^{l_2}\}_{w=1, \dots, W_{i_2}}$  is defined as follows (based on [4]) :

$$D_M^{l_1 l_2} = \frac{\sum_{i=1}^{W_{i_1}} \alpha_i^{l_1} + \sum_{j=1}^{W_{i_2}} \beta_j^{l_2}}{\sum_{i,j | \text{only for } \alpha_i^{l_1}} o_i^{l_1} o_j^{l_2} + \sum_{i,j | \text{only for } \beta_j^{l_2}} o_j^{l_2} o_i^{l_1}} \quad (8)$$

with  $\alpha_i^{l_1} = \min_j o_i^{l_1} o_j^{l_2} d_{ij}$  and  $\beta_j^{l_2} = \min_i o_i^{l_2} o_j^{l_1} d_{ij}$  where  $d_{ij}$ ,  $o_{i_1}$  and  $o_{j_2}$  denote respectively the Euclidian distance between centroid  $m_i^{l_1}$  from codebook  $\mathbf{m}_{i_1}$  and centroid  $m_j^{l_2}$  from codebook  $\mathbf{m}_{i_2}$ , number of feature vectors from segment  $s_{i_1}$  assigned to centroid  $m_i^{l_1}$ , and number of feature vectors from segment  $s_{i_2}$  for centroid  $m_j^{l_2}$ .

Once codebooks have been trained for every segment, the distance (8) is measured between every two segments and a distance matrix is build. Various ag-

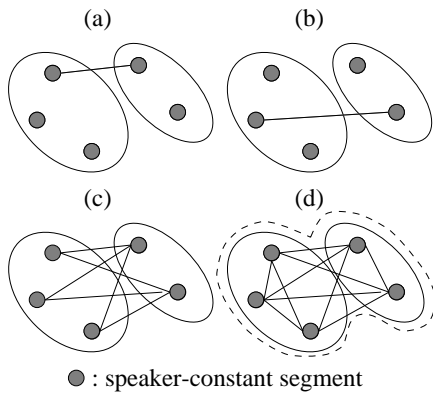


Figure 3: Various distances between groups used in neighborhood clustering schemes.

glomerative schemes [10] have been tested (see Figure 3):

**Single linkage** The distance between two clusters is defined as the distance between their two closest members (Figure 3.(a)).

**Complete linkage** The distance between two clusters is defined as the distance between their two farthest members (Figure 3.(b)).

**Average linkage between groups** The distance between two clusters is defined as the average of the distances between all pairs of members, one segment taken in each cluster (Figure 3.(c)).

**Average linkage within groups** The distance between two clusters is defined as the average of the distances between all pairs in the cluster which would result from combining the two clusters (Figure 3.(d)).

### 3. EXPERIMENTAL RESULTS

#### 3.1. Evaluation data

The speaker-based segmentation algorithm proposed in the previous section is tested with broadcast news programmes provided by BBC. These programmes are completely transcribed and hand-segmented according to speaker identity, background conditions and channel conditions changes. This evaluation data consists of radio news bulletins, about 30 minutes long each. There are about 65 (min = 57, max = 77) speaker changes along each programme. The mean number of speakers per programme is about 34 (min = 29, max = 39). As shown in Figure 4, the true segment length seems to be distributed according to an unimodal distribution with a mode around 20s. It is also interesting to mention that the segments are not allocated uniformly among the different speakers in a programme : there is generally one main news reader who speaks alternately with the rest of the speakers. For example, 20 segments

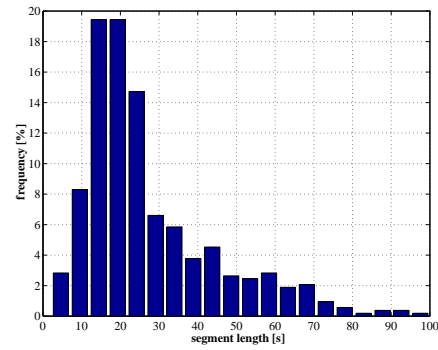


Figure 4: Observed distribution of the segment length (mean length  $\mu = 27.4s$  and standard deviation  $\sigma = 17.3s$ ).

were pronounced by the main news reader while the 30 other speakers shared the 41 remaining segments of a 61 segments programme (i.e., only 1 or 2 segments for every speaker except for the main news reader).

Even if a multiple sessions processing is possible, each programme is processed separately. Both procedures are based on 24-dimensional Mel-warped feature vectors computed from raw speech. These cepstral vectors are extracted every 10ms on 30ms analysis windows.

#### 3.2. Splitting Procedure

Different values for window length, window overlap and window shift were tested. Some proposed to use short windows with no overlap [5] in order to maximize discrimination, while others used longer windows with overlap [11]. We observed that increasing these parameters has a global smoothing effect on the signal distance, softening the peaks. Finally, window length, window overlap and window shift are respectively set to 5s, 1s and 50ms, as a trade-off between speed and accuracy. As already mentioned, the peak detection algorithm along the distance signal is based on two thresholds which are not set automatically and depend on the distance definition. These two thresholds are hand-tuned but stay constant for every programme.

Performances of the *Chop* procedure are expressed in terms of *Detection Rate* (percentage of true changes successfully detected), *False Alarm Rate* (percentage of changes erroneously detected) and *Mean Detection Error* (mean error between estimated and true changes times) as a function of *Tolerance* defined as the maximum time interval accepted between detected and true changes.

As shown in Figure 5, the splitting algorithm can detect speaker changes with reasonable accuracy, whatever the cluster distance is. The number of peaks admitted as real speaker changes depends on the maximum acceptable delay (i.e., *tolerance*). Clearly, the mean detection error stays low even if you increase the

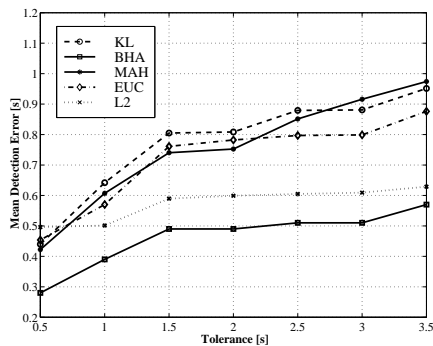


Figure 5: Mean Detection Error versus Tolerance for various cluster distances used in the Chop procedure.

| Cluster distance | DET rate [%] | FA rate [%] |
|------------------|--------------|-------------|
| KL               | 93.4         | 4.9         |
| BHA              | 96.6         | 6.1         |
| MAH              | 93.5         | 5.3         |
| EUC              | 94.2         | 5.4         |
| L2               | 95.8         | 5.9         |

Table 1: Comparison of detection rates (DET) and false alarm rates (FA) for various cluster distances used in the splitting procedure (Tolerance = 1s).

tolerance. This means that not many new changes are added for higher tolerance and they just can be seen as outliers. However, the major drawback of this method is the false alarm rate (see Table 1) : the thresholds of the peaks detection algorithm have to be small enough not to miss too much real changes. Consequently, the peaks detector is triggered off too often.

### 3.3. Merging Procedure

The merging procedure has been performed with true segments for each programme separately. The non-parametric representations for every segment  $s_i$  are  $W_1$ -clusters codebooks. The codebook size  $W_1$  must be chosen as a trade-off between computational burden and suitable segment representation (i.e., minimal admissible distortion should be guaranteed).  $W_1$  is set to 20 in our experiments. If a cluster label is assigned to the most represented speaker within this cluster, its purity may be defined as the ratio between the number of segments from this speaker to the total count of the cluster. We observe that our clustering algorithm can produce clusters with high purity. All agglomerative schemes seem to work equivalently (see Figure 6). Notice that most part of the 50%-purity clusters are grouping two segments from two different speakers, who generally spoken only one time, and are erroneously merged too early during the clustering procedure. Beyond these results, Table 2 shows classification rates. These results take into account one more constraint : there can be only one cluster for each speaker in the final partition (i.e., the cluster label is not necessary as-

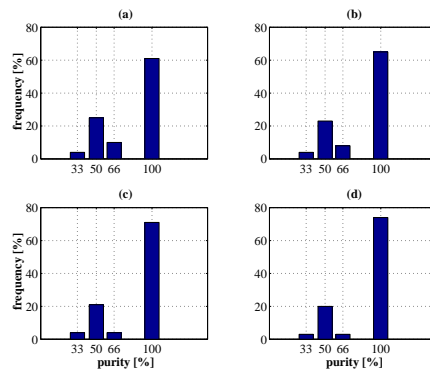


Figure 6: Cluster purities for 4 merging schemes : (a) single linkage, (b) complete linkage, (c) average linkage between groups, (d) average linkage within groups.

| Merging scheme         | Classification rate [%] |
|------------------------|-------------------------|
| Single linkage         | 69.2                    |
| Complete linkage       | 69.7                    |
| Average between groups | 72.3                    |
| Average within groups  | 73.1                    |

Table 2: Comparison of classification rates for 4 merging schemes.

signed to the most represented speaker). It is also observed that the most often misclassified segments are the shortest ones, maybe because their codebooks are not a sufficiently good representation of their speakers.

Notice that the difficulty of the classification task grows as the number of segments and the number of speakers increase. For example, a less complex 4-speakers/6-segments (i.e., 24 segments) problem is generally solved perfectly, producing 4 pure clusters.

### 3.4. Complete Procedure

Finally, the full algorithm is applied to the BBC recordings : the merging procedure is directly fed by the segment boundaries extracted by the splitting procedure. The recordings are still processed separately. The Bhattacharyya (3) cluster distance is used to chop the audio streams and the average linkage within groups scheme is chosen for reclustering the segments. Such algorithms are generally used to enhance the performances of unsupervised adaptation for speech recognition [7, 11, 6, 9]. Hence, performances are given in terms of error rates reduction. Since we are only concerned by the speaker-based segmentation problem, we do not produce such results. We just observe that the final segmentation is always worse than the one obtained with only true speaker changes. Because of the extra false changes, the number of segments is increased leading to a more complex task. As there are more shorter segments and the boundaries are not well defined (i.e., feature vectors from multiple speakers in the same segment), the codebooks are less discriminant

and the clustering process is more confused.

#### 4. DISCUSSION

Siegler *et al.* [11] proposed a segmentation algorithm based on a Kullback-Leibler metric as acoustic distance. Using also a threshold method to detect acoustic changes, they reported a 64.0% detection rate on the DARPA'96 HUB4 evaluation test set. Beigi *et al.* [5] used a similar splitting method but with a more discriminant distance and obtained a 70.0% detection rate with the DARPA'95 HUB4 data. Chen *et al.* [6] applied a maximum likelihood method based on a Bayesian information criterion to perform either splitting and clustering. For the DARPA'97 HUB4 data, their algorithm detected 91.5% of the true changes with a 2s tolerance and only a 4.1% false alarm rate. Tested on the DARPA'96 HUB4 data (824 segments for 28 speakers), their cluster algorithm produced automatically 31 clusters with high purity ( $\approx 92.0\%$ ). In all these works, only cepstral features were used.

Comparing to these previous results, our algorithm seems to work reasonably well. However some improvements are necessary. For example, the *splitting* procedure should be made more efficient and accurate. The thresholds of the peak detector should be data-driven and tuned automatically. Moreover, the *merging* procedure is not robust enough and unadapted to complex tasks. This last problem may be addressed by using better speaker models to represent the segments or more advanced clustering schemes. Up to now, the full process is very time consuming because of the computationally expensive *K-means* clustering which is performed in both procedures. Decreasing window size and increasing window shift in the *splitting* procedure, and decreasing the number of centroids in the segment codebooks in the *merging* procedure, are not the solution. Thus, it suggests once again that better (i.e., more speaker discriminant) acoustic representations should be used for the analysis windows and the segments.

#### 5. CONCLUSION

This paper presented a speaker-based segmentation of broadcast news recordings, which is developed in the framework of the THISL project. This speaker tracking system is based on *Chop-and-Recluster* algorithm. The splitting procedure rests on a *metric-based* technique detecting speaker changes as maxima of a distance signal measured between two windows shifted along the audio stream. Once the *Chop* procedure has been done, the extracted segments are *Reclustered* using an agglomerative clustering method based on neighborhood schemes. To do so, each audio segment is represented by a non-parametric model (i.e., a codebook trained with the feature vectors belonging to the segment). Many different acoustic distances for the

*splitting* procedure were tested and various hierarchical clustering schemes were applied for the *merging* procedure. Even with a good detection rate ( $\approx 95.0\%$ ) of the true speaker changes, a fair accuracy ( $\approx 0.7s$ ) on boundaries and a reasonable false alarm rate ( $\approx 5.5\%$ ), the full procedure lacks robustness to produce only high purity groups and high classification rate for the typical task (65 segments / 35 speakers for each broadcast news recording).

#### 6. REFERENCES

- [1] D. Abberley, S. Renals and G. Cook, "Retrieval of broadcast news with the THISL system", Proc. of ICASSP, Vol. 6, pp. 3781–3784, 1998.
- [2] D. Abberley, D. Kirby, S. Renals and T. Robinson, "The THISL broadcast news retrieval system", Proc. of ESCA ETRW Workshop on Accessing Information in Spoken Audio, Cambridge (UK), April 1999.
- [3] M. Basseville, "Distance measures for signal processing and pattern recognition", Signal Processing, Vol. 18(4), pp. 349–369, 1989.
- [4] H. Beigi, S. Maes and J. Sorensen, "A distance measure between collections of distributions and its application to speaker recognition", Proc. of ICASSP, Vol. 2, pp. 753–7756, 1998.
- [5] H. Beigi and S. Maes, "Speaker, channel and environment change detection", Proc. of the World Congress on Automation, 1998.
- [6] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion", Proc. of DARPA Broadcast News Transcription and Understanding Workshop, 1998.
- [7] H. Jin, F. Kubala and R. Schwartz, "Automatic speaker clustering", Proc. of DARPA Speech Recognition Workshop, 1997.
- [8] S. Johnson, "Speaker tracking", M.Phil Thesis, Cambridge University Engineering Dept., UK, 1997.
- [9] S. Johnson and P. Woodland, "Speaker clustering using direct maximisation of the MLLR-adaptation likelihood", Proc. of ICSLP, 1998.
- [10] L. Lebart, A. Morineau and M. Piron, "Statistique exploratoire mutlidimensionnelle", Dunod, 1995.
- [11] M. Siegler, U. Jain, B. Raj and M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio", In DARPA Proc. of Speech Recognition Workshop, pp. 97–99, 1997.