Applied Clustering for Automatic Speaker-Based Segmentation of Audio Material

Olivier Pietquin Laurent Couvreur
Faculté Polytechnique de Mons
Parc Initialis – Av. Copernic, 1
7000 Mons
Belgium
E-mail: {pietquin,lcouv}@tcts.fpms.ac.be

Pierre Couvreur
Université de Mons-Hainaut
Place du Parc, 20
7000 Mons
Belgium
E-mail: pierre.couvreur@umh.ac.be

Abstract

In this paper, we introduce an algorithm dedicated to speaker-based segmentation of audio material. The algorithm consists in two distinct procedures namely splitting and merging. Its performance is assessed on broadcast news recordings provided by the British Broadcast Corporation (BBC). Results show that the splitting is performed with high accuracy and low missed detection rate while the merging procedure provides satisfying results.

1. Introduction

Speaker-based segmentation (also referred as speaker-tracking in the following) can be defined as splitting and labelling a spoken audio stream associated with an unknown number of unknown speakers into homogeneous regions according to speaker identity.

The algorithm presented in this paper has been developed in the framework of the THISL (THematic Indexing of Spoken Language) project [1]. The ultimate goal of this project was developing a system for indexation and retrieval of BBC Radio/TV recordings [2,3]. Material for indexation is obtained by automatic transcription of the recordings. In addition to the

recognition rate improvement by speaker-adaptation[1] [4,5], speaker-tracking also provides additional information about the audio database. That is, speaker identity is a valuable indexation key. Other applications of speaker-based segmentation are automatic active subtitling of movies or active help for ear impaired people. For example, speaker-tracking allows to change the subtitle colours dynamically according to speaker identity.

The proposed speaker-tracking algorithm can be summarized as follows. First, the audio stream is represented by a sequence of acoustic vectors. Next, speaker changes are found as steady state transitions along the sequence of acoustic vectors : the so-called splitting procedure. We assume that different speakers never speak simultaneously. Without any detection errors, homogeneous sequences of acoustic vectors (called *segments* in the following) are identified. Finally, a merging algorithm is applied to the segments to form homogeneous clusters according to speaker identity. The number of speakers is assumed to be known.

This paper is organized as follows : next section briefly describes the speech analysis technique. We then present the splitting and merging procedures. Finally, experimental results are reported and potential improvements are discussed.

2. Speech Signal Representation

In order to represent raw speech into a form suited for computer-based speaker-tracking, the speech signal is first sampled at discrete time instants (Fig. 1). Direct use of speech signal samples for speaker-tracking is too cumbersome and time-consuming. The sample sequence is instead processed to reduce the data stream and take advantage of speech redundancy. More especially, 24 *MEL-cepstrum* coefficients [6] are computed every 10ms from a 30ms analysis window (time while the speech signal can be considered as stationary). That is, two consecutive analysis frames overlap each other over 20ms.

[1] Automatic Speech Recognition (ASR) consists in decoding an acoustical speech signal into the text that has actually been uttered. Most speech recognizers use Hidden Markov Models (HMM) which have to be trained on large corpus of speech. Performances depend on the voice of the user which is most of time absent from the training database. To improve the recognition rate, we can either enlarge the training corpus or try to adapt the models to the user voice [4].

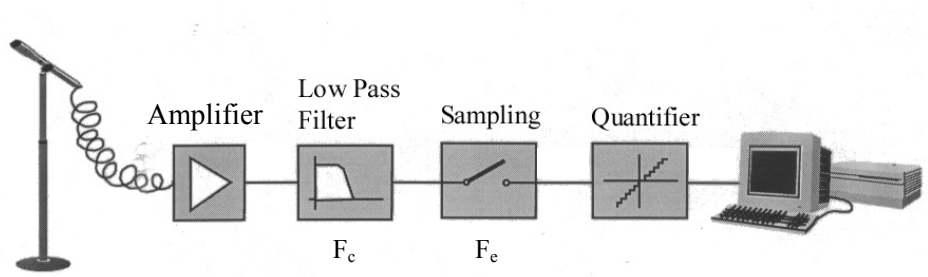

Fig. 1: Sampling channel. F_c is the cutting frequency of the guard filter and F_e is the sampling frequency.

Ultimately, the audio stream is replaced by a sequence of acoustic vectors. Each acoustic vector is computed as follows :

$$\{s(n)\}_{n=1,\dots,N} \xrightarrow{\text{FFT+MEL}} \{S(k)\}_{k=1,\dots,24} \xrightarrow{\log|\,|} \{\log|S(k)|\}_{k=1,\dots,24} \xrightarrow{\text{IDCT}} \{c(n)\}_{n=1,\dots,24}$$

where {s(n)} stands for the N speech samples gathered for each 30ms analysis frame, e.g. N = 240 for a 8 kHz sampling frequency. The *MEL-spectrum* vector {S(k)} is obtained by feeding a MEL filter bank with the magnitude of the spectrum computed over the analysis frame (Fig. 2). The spectrum can be efficiently computed by a Fast Fourier Transform (FFT). A logarithmic transform followed by an Inverse Discrete Cosine Transform (IDCT) finally provides the *MEL-cepstrum* coefficient vector {c(n)} for the current frame. The MEL analysis technique approximates the non-linear human hearing process and provides weakly correlated coefficients which are meaningful parameters for speech recognition and speaker-tracking. Fig.3 shows the speech signal and the corresponding MEL-cepsrum coefficients for a 2-speaker utterance.

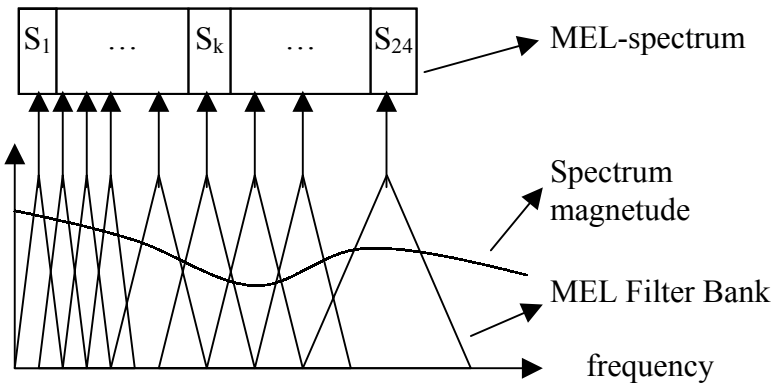

Fig. 2 : MEL Filter Bank

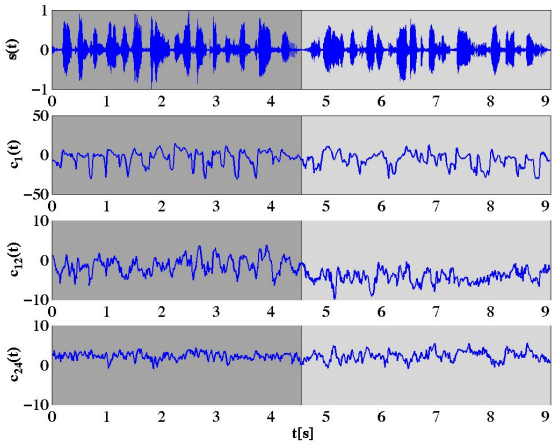

Fig. 3 : Speech signal and MEL-cepstrum coefficients (only 1st, 12th and 24th orders coefficients are represented).

3. Segmentation Algorithm

3.1 Splitting Procedure

Several methods exist to detect speaker changes in a speech signal [7]. In this work, a distance-based method has been applied. Such a method identifies speaker changes as maxima of a distance, the so-called splitting distance D_S^{lr}, computed between two contiguous sets of acoustic vectors (also referred as window in the following) sliding along the speech signal.

The following notation is used :

- ν_k = any acoustic vector;
- V_l = any left window of acoustic vectors;
- V_r = any right window of acoustic vectors;
- K = total length of audio recording in number of acoustic vectors.

Algorithm (Fig. 4) :

a. Initialize : $k_0 = 1$

b. Collection of acoustic vectors in two neighbouring windows :

$$k_{l1} = k_0$$
$$k_{l2} = k_{l1} + \gamma_l \quad (\gamma_l \in Z \text{ and } \gamma_l > 0)$$
$$k_{r1} = k_{l2} - \gamma_{lr} \quad (\gamma_{lr} \in Z \text{ and } \gamma_{lr} \geq 0)$$
$$k_{r2} = k_{r1} + \gamma_r \quad (\gamma_r \in Z \text{ and } \gamma_r > 0)$$

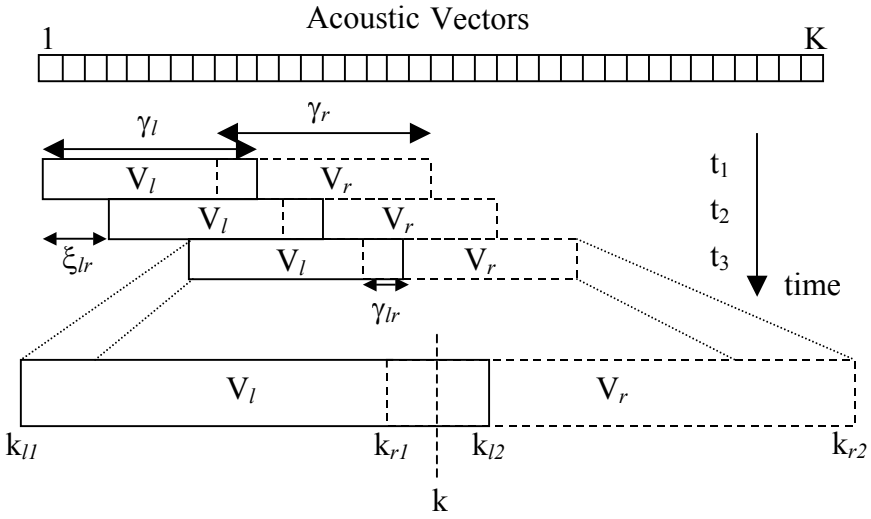

Fig. 4 : Windows V_l and V_r are shifted along the sequence of acoustic vectors.

$$V_l = \{\nu_k\},\ k = k_{l1},\ldots,k_{l2}$$
$$V_r = \{\nu_k\},\ k = k_{r1},\ldots,k_{r2}$$

where γ_l, γ_r and γ_{lr} are the length of the left and right windows and the window overlap, respectively[2].

c. Computation of the distance D_S^{lr} between V_l and V_r. It is associated with the discrete time instant $k = k_{l2}/2 + k_{r1}/2$.

d. Shift of the windows : $k_0 = k_0 + \xi_{lr}$ with ξ_{lr} being the shift size.

e. If $k_{r2} < K$, go to b.

The splitting distance D_S^{lr} is expected to be small when the left and the right windows contain acoustic vectors from a single speaker. It is expected to be high otherwise. Assume that the acoustic vectors from every window are drawn from a multidimensional Gaussian distribution. Measuring a distance between two windows, i.e. two sets of acoustic vectors, reduces to computing a statistical distance between two Gaussian distributions whose parameters have been estimated for each window. Many statistical distances may be proposed [8]. The Kullback-Leibler (KL) and the Bhattacharyya (BHA) distances have been tested for full covariance matrices while the Mahalanobis (MAH), the Euclidian

[2] All the duration parameters are expressed in number of acoustic vectors.

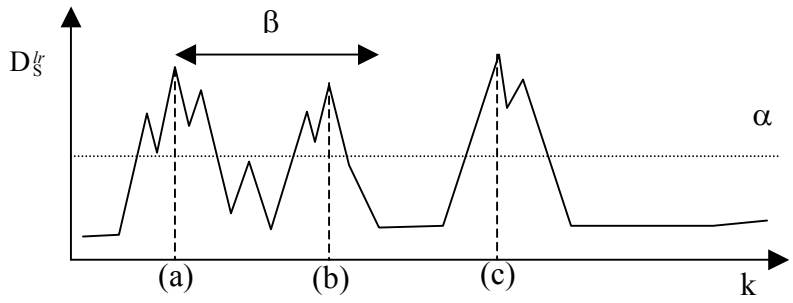

Fig. 5 : (a) Only the highest value over α is considered, (b) detected value is rejected because it is too close to the previous one and (c) a valid speaker change.

(EUC) and the L2 distances have been considered in the case of diagonal matrices. That is, the distance d_x^{l2} between two Gaussian distributions $N(\bar{\mu}_1,\Sigma_1)$ and $N(\bar{\mu}_2,\Sigma_2)$[3] is given by one of the following equation:

$$d_{KL}^{12}=\frac{1}{2}(\bar{\mu}_2-\bar{\mu}_1)^T(\Sigma_2^{-1}+\Sigma_1^{-1})(\bar{\mu}_2-\bar{\mu}_1)+\frac{1}{2}tr(\Sigma_1^{-1}\Sigma_2+\Sigma_2^{-1}\Sigma_1-2I) \quad (1)$$

$$d_{BHA}^{12}=\frac{1}{4}(\bar{\mu}_2-\bar{\mu}_1)^T(\Sigma_2+\Sigma_1)^{-1}(\bar{\mu}_2-\bar{\mu}_1)+\frac{1}{2}log\frac{|\Sigma_1+\Sigma_2|}{2\sqrt{|\Sigma_1\Sigma_2|}} \quad (2)$$

$$d_{MAH}^{12}=\frac{1}{n}(\bar{\mu}_2-\bar{\mu}_1)^T(\Sigma_2\Sigma_1)^{-1}(\bar{\mu}_2-\bar{\mu}_1)=\frac{1}{n}\sum_{k=1}^{n}\frac{(\mu_{2k}-\mu_{1k})^2}{\sigma_{1k}\sigma_{2k}} \quad (3)$$

$$d_{EUC}^{12}=(\bar{\mu}_2-\bar{\mu}_1)^T(\bar{\mu}_2-\bar{\mu}_1)=\sum_{k=1}^{n}(\mu_{2k}-\mu_{1k})^2 \quad (4)$$

$$d_{L2}^{12}=\sqrt{\int_{\Re^n}\left[N(\bar{\mu}_2,\Sigma_2)-N(\bar{\mu}_1,\Sigma_1)\right]^2 d\bar{X}} \quad (5)$$

Once a distance has been computed for all window pairs, maxima are detected according to a simple criterion based on two thresholds (α,β) as shown on Fig. 5. The threshold α corresponds to the minimum value for detecting a peak and β guarantees a minimum delay between to consecutive

[3] $N(\bar{\mu},\Sigma)$ denotes a multidimensional distribution with a mean vector $\bar{\mu}$ and a covariance matrix Σ.

changes. It corresponds to the minimum time while a speaker is not interrupted.

In order to increase the effectiveness of the splitting procedure, we propose to apply first a K-means clustering [9] over each window. That is, the acoustic vectors of left and right windows V_l and V_r are classified into N_K classes. Members of each class are assumed to be drawn from a multidimensional Gaussian distribution. Let define $V_i^l \sim N(\bar{\mu}_i^l, \Sigma_i^l)$ as the Gaussian distribution associated with class 'i' for the left window. Likewise, $V_j^r \sim N(\bar{\mu}_j^r, \Sigma_j^r)$ is defined as the Gaussian distribution associated with class 'j' for the right window. The splitting distance D_S^{lr} between the left and the right windows is now computed as follows :

$$D_S^{lr} = \frac{d^{lr}}{d} \times \frac{max_{i,j}\, d_{ij}^{lr}}{min_{i,j}\, d_{ij}^{lr}} \qquad 1 \le i, j \le N_K \tag{6}$$

where d_{ij}^{lr} stands for the statistical distance between the Gaussian distribution corresponding to class 'i' in the left window and the one corresponding to class 'j' in the right window $\left(d\left[N(\bar{\mu}_i^l, \Sigma_i^l), N(\bar{\mu}_j^r, \Sigma_j^r)\right]\right)$, and d^{lr} denotes the statistical distance between the left and the right windows $\left(d\left[N(\bar{\mu}^l, \Sigma^l), N(\bar{\mu}^r, \Sigma^r)\right]\right)$ without any clustering, while d is defined as the mean d^{lr} over all window pairs. Those distances are computed using one of the previous definitions (1)-(5). Fig. 6 clearly shows that (6) is more

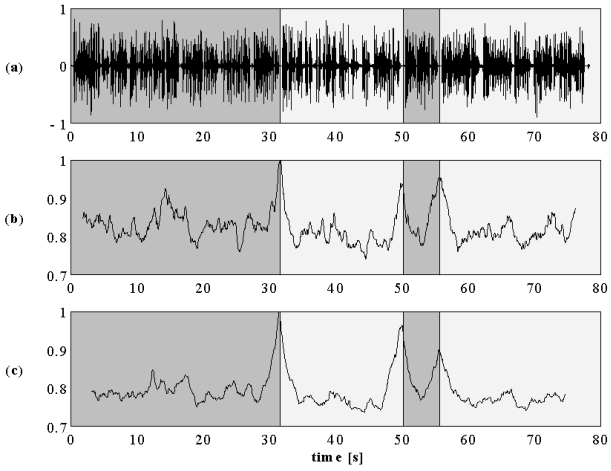

Fig. 5 : (a) Three speaker change utterance, (b) splitting distance without K-means clustering D_S^{lr} given by (2) and (c) splitting distance with K-means clustering (definition (6) is used in combination with (2)).

suited for detecting the speaker changes: actual peaks are strengthened while peaks at no-change locations are less disturbing for the peak detector. To reduce the computational burden, the number of clusters has to be chosen judiciously (typically three clusters are computed). Moreover, in order to fasten the clustering, the centroids of the current analysis window are initialized with the centroids of the previous one, so strong adaptation is needed only when there is a significant acoustic change.

3.2 Merging Procedure

Once the audio stream has been splitted into homogeneous segments, i.e. sets of acoustic vectors, according to speaker identity, an agglomerative hierarchical clustering is performed to group the segments uttered by identical speakers.

First, each segment is represented by a codebook consisting in a set of N_l centroids. This codebook is once again estimated by a K-means clustering over the acoustic vectors in the segment. Next, a bottom-to-up clustering is performed over the codebooks. At each step, the two nearest clusters of the current partition are merged to obtain a new partition. Several agglomerative schemes have been tested (Fig. 7) [10] : single linkage (SL), complete linkage (CL), average linkage between groups (ALBG) and average linkage within groups (ALWG). They involve computing the distance between two codebooks. The following merging distance $D_M^{l,2}$ is proposed [11]:

$$D_M^{l,2} = \frac{\sum_{i=1}^{N_l} \alpha_{1i} + \sum_{j=1}^{N_l} \beta_{2j}}{\sum_{i,j|\text{only for } \alpha_{1i}} o_{1i} o_{2j} + \sum_{i,j|\text{only for } \beta_{2i}} o_{2j} o_{1i}} \tag{7}$$

with

$$\alpha_{1i} = \min_j \; o_{1i} \; o_{2j} \; d_{ij} \tag{8}$$

$$\beta_{2j} = \min_i \; o_{2j} \; o_{1i} \; d_{ij} \tag{9}$$

where d_{ij} denotes the Euclidian distance between the centroid ‘i’ from codebook 1 and the centroid ‘j’ from codebook 2, o_{1i} is the number of acoustic vectors from segment 1 assigned to the centroid ‘i’ and o_{2j} is the number of acoustic vectors from segment 2 assigned to the centroid ‘j’.

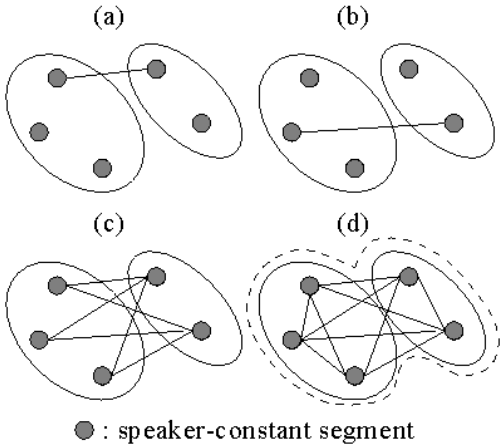

Fig. 6 : Different agglomerative schemes : (a) single linkage, (b) complete linkage, (c) average linkage between groups and (d) average linkage within groups.

4. Experimental Results

The algorithm described in the previous sections has been tested with broadcast news recordings provided by the BBC. These evaluation data consist of radio news bulletins about 30 minute long each, and have been hand-segmented according to speaker identity. In this paper, we report typical results for a 31 minute recording. The main speaker pronounced 17 segments while 20 speakers only pronounced 1 segment and the 28 remaining segments where uttered by 13 speakers.

4.1 Splitting Procedure

Several values for the window length, the window overlap and the window shift have been tested. We observed that increasing those parameters has a smoothing effect on the distance curve and especially the window overlap. The following setup is chosen : the window length, the window overlap and the window shift are set to 3s ($\gamma_l = \gamma_r = 300$), 500ms ($\gamma_{lr} = 50$) and 50ms ($\xi_{lr} = 5$), respectively. As seen before, the peak detection algorithm is based on a two threshold criterion. Those thresholds depend on the distance definition and has to be hand-tuned but stay constant for each recording.

The performance of the splitting procedure is given in terms of *Detection Rate* (*DR*) and *False Alarm Rate* (*FAR*) :

$$DR[\%] = \frac{\#\text{Detected True Changes}}{\#\text{True Changes}} \times 100 \quad (10)$$

$$FAR[\%] = \frac{\#\text{Erroneously Detected Changes}}{\#\text{Detected Changes}} \times 100 \quad (11)$$

Those values have been calculated for each definition of the statistical distance d_X^{L2} (see Table 1). Of course, it is possible to increase DR by decreasing the threshold α. Actually, this leads to detect more peaks but it also increases FAR. The latter errors may be compensated by a reliable merging procedure. However, the shorter the segments are, the more badly the segment codebooks are estimated and the less reliable the merging procedure is. A trade-off between splitting and merging performances has to be made and missed detections have to be tolerated.

Distance	DR[%]	FAR[%]
BHA	97.01	7.46
KL	93.51	11.9
MAH	94.02	8.34
EUC	94.43	8.34
L2	96.61	8.65

Table 1: Comparison of DR, FAR and MDR for various cluster distances

4.2 Merging Procedure

The performances of the merging procedure are measured by computing the Rand Index value I_{RAND} [12] for the 34-cluster partition of 65 segments from the 34-speaker recording mentioned previously. It gives the number of pair of segments from the same speaker assigned to different clusters or segments from different speakers assigned to the same cluster. Ideally, I_{RAND} is equal to zero and is computed as follows :

$$I_{RAND} = \frac{1}{2}\left[\sum_{i=1}^{N_C} n_{i\bullet}^2 + \sum_{j=1}^{N_S} n_{\bullet j}^2\right] - \sum_{i=1}^{N_C}\sum_{j=1}^{N_S} n_{ij}^2 \quad (12)$$

were n_{ij} denotes the number of segments from speaker 'j' in cluster 'i', $n_{i\bullet} = \sum_j n_{ij}$, $n_{\bullet j} = \sum_i n_{ij}$, N_S is the number of speakers and N_C is number of clusters. Fig. 8 shows I_{RAND} as a function of the codebook size (i.e., the number of centroids used to model each segment) for different

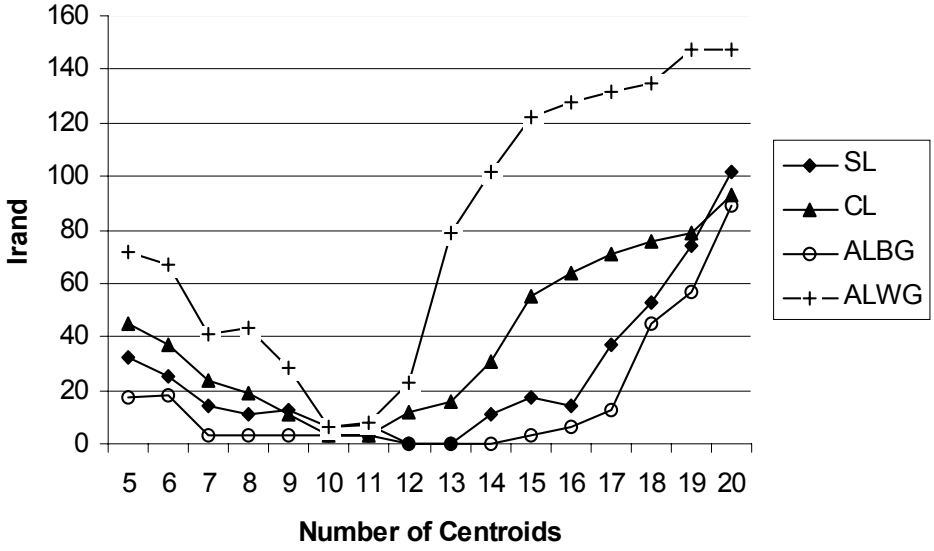

Fig. 6 : Clustering performances

agglomerative schemes. For a well-chosen codebook size, the merging procedure performs well or even perfectly. Below the optimal codebook size, the codebooks do not model well enough the inter-speaker variability. Over the optimal codebook size, the codebooks are not well designed because too many centroids have to be estimated and too few acoustic vectors are assigned to each centroid. In both cases, mismodeling of segments is performed, leading to classification errors during the merging procedure. Best results are obtained for the ALBG scheme.

5. Discussion

Compared to previous results [13][14], our method provides satisfying results. However, some improvements may be suggested. First, the thresholds should be data-driven and automatically tuned. Next, model selection techniques such as *Bayesian Information Criterion* (BIC) can be used to improve the splitting procedure by reducing the FAR [7]. In the framework of THISL, news recordings are transcribed for indexing. So the transcriptions, i.e. the word sequences, are available and may be used for helping the splitting procedure. Since a speaker change can only occur between two different words, positioning the windows V_l and V_r at word boundaries improves the precision of the peak detection. Tests have shown that using this technique allows to reach a high accuracy in detection of speaker changes.

Besides, the assumption that the number N_S of speakers is a priori known should be relaxed. One should suggest to resort to blind clustering. That is, the merging procedure has to stop automatically when the partition counts as many classes as speakers without knowing the number N_S of speakers. For example minimizing I_{RAND} may be used as a criterion to pick up blindly the N_S-cluster partition. Other criterions based on maximizing some partition purity may be used as show in [7].

6. References

[1] THISL Web site : http://www.dcs.shef.ac.uk/research/groups/spandh/projects/thisl

[2] D. Abberley, S. Renals, G. Cook, 'Retrieval of Broadcast News with the THISL System', *Proc. of ICASSP, Vol. 6, pp. 3781-3784, 1998 Seattle USA.*

[3] D. Abberley, D. Kirby, S. Renals, T. Robinson, 'The THISL Broadcast News Retrieval System', *Proc. of ESCA ETRW Workshop on Accessing Information in Spoken Audio, Cambridge (UK), 1999.*

[4] L. Rabiner, B.H. Juang, 'Fundamentals of Speech Recognition', *Prentice Hall Ed., 1993.*

[5] H. Jin, F. Kubala, R. Schwartz, 'Automatic Speaker Clustering' *Proc. of DARPA Speech Recognition Workshop, 1997.*

[6] J.W. Picone, 'Signal Modeling Techniques in Speech Recognition', *Proc. of the IEEE, Vol 81, n. 9, pp 1215-1247, sep 1993.*

[7] P. Delacourt, 'La Segmentation et le Regroupement par Locuteurs pour l'Indexation Audio', *Institut Eurecom, PhD. Thesis.*

[8] M. Basseville, 'Distance Measures for Signal Processing and Pattern Recognition' *Signal Processing, Vol. 18(4), pp. 349-369, 1999.*

[9] H. Beigi, S. Maes, J. Sorenson, 'A Distance Measure between Collections of Distributions and its Application in Speaker Recognition', *Proc. of ICASSP, Vol. 2 pp. 753-776, 1998.*

[10] L. Lebart, A. Morineau, M. Piron, 'Statistique Exploratoire Multidimensionnelle', *Dunod, 1995.*

[11] L. Couvreur, P. Couvreur, 'Application de Méthodes de Classification pour la Segmentation Automatique de Programmes Radio/TV en fonction du locuteur' *Proc. of XXXIèmes Journées Françaises de Statistique, pp. 423-426, Grenoble (Fr), 1999.*

[12] L. Huber, P. Arabie, 'Comparing Partitions', *Journal of Classification, Vol. 2 pp. 193-218, 1985.*

[13] S. Chen, 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.*

[14] M. Siegler, U. Jain, B. Raj, M. Stern, 'Automatic Segmentation, Classification and Clustering of Broadcast News Audio', *Proc. of DARPA Speech Recognition Workshop, 1997.*

[15] L. Couvreur, J.M. Boite, 'Speaker Tracking in Broadcast Audio Material in the Framework of the THISL Project', *Proc. of ESCA ETRW Workshop on Accessing Information in Spoken Audio, Cambridge (UK), 1999.*