A first step into speaker-based indexing

Perrine Delacourt * and Christian J.Wellekens

Institut EURECOM, 2229 route des Crêtes,
BP 193, 06904 Sophia Antipolis Cedex, France
{perrine.delacourt,christian.wellekens}@eurecom.fr
http://www.eurecom.fr/~delacour/speech/

Abstract. In this paper, we present the architecture of our speaker-based indexing system. The goal is to recognize from their voice the sequence of people engaged in a conversation. In our context, we make no assumptions about prior knowledge of the speaker characteristics (no speaker model, no speech model, no training phase). And the number of speakers is unknown. However, we assume that people do not speak simultaneously. For each stage of our speaker-based indexing system, we detail the constraints and propose or review some techniques according to these constraints. Finally, evaluation methods for each stage are examined.

1 Introduction

With the increasing number of TV or radio channels, a considerable amount of broadcast is stored either for archiving purposes or for later use in TV or radio on demand. In addition to the storage problem, the problem of fast and efficient retrieval arises. Therefore, automated data indexing and retrieval systems are more and more needed.

Among the possible indexing keys for audio data, we take a particular interest in speaker identity. This paper addresses the audio document indexing via the task of recognition of the sequence of speakers involved in a conversation from their voice. Since in applications training data is not always available and since we do not always know the number of speakers involved, we assume that we do not have any speaker model and not enough material to build some. Also the number of different speakers is unknown as well as the number of speaker occurrences. However, for the sake of simplicity and because it is a realistic hypothesis for most of TV or radio broadcasts, we assume that people do not speak simultaneously.

A speaker-based indexing system could be used for example to create a database where all utterances are indexed with respect to their author. At present, this type of task is done by hand, forcing the operator to listen to the whole audio document. Speaker-based indexing task can also be used as a preliminary step for transcription task [1], [2] or speaker tracking [3]. Indeed, speech recognition rates are improved when speech models are adapted to speakers. Therefore, audio data are first labeled according to speakers, then the pre-trained speech models are adapted with the data contained in the document. Finally, the speaker-adapted speech models are used for speech recognition. Concerning speaker tracking, the more data we have to take the identification decision, the more correct and reliable is the decision. In this way, the audio document is first indexed by speakers. Then, the decision whether the target speaker is speaking or not is taken on each segment rather than on few frames.

The speaker-based indexing system we will describe in this paper is first introduced section 2. The next sections detail the different steps of our system: the parameterization step is described in section 3, the segmentation step in section 4, the grouping step in section 5 and the modeling and recognition steps in section 6. Section 7 deals with the evaluation methods for each step of such a system and section 8 presents partial results. Finally, section 9 concludes and gives possible tracks for further work.

2 Description of the speaker-based indexing system

Our indexing system is organized in five steps, as described in figure 1. First, the audio data is parameterized to form the so-called acoustic vectors. The second step consists in segmenting the parameterized signal according to speakers involved in the conversation. In other words, the aim is to obtain speech segments as long as possible and containing utterances pronounced by a single speaker. The next step is concerned with the segment grouping: segments are clustered so that each cluster contains segments

* The financial support of this project from the Centre National d'Etudes des Télécommunications (CNET) under the grant n° 98 1B is gratefully acknowledged here.

related to a given speaker and all segments related to this speaker are situated in this cluster. A speaker model is built from each of the resulting clusters during the speaker modeling step and the models are finally used to possibly refine the segmentation and to recognize the sequence of speakers.

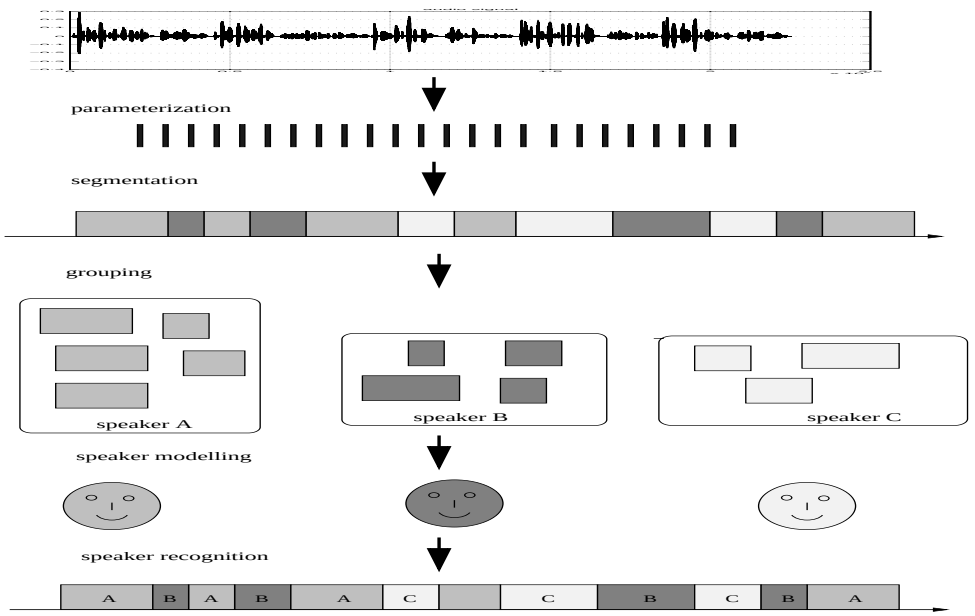

Fig. 1. Speaker-based indexing system

3 Parameterization

We briefly present the parameterization since it has been studied largely. For further details, see [4]. The speech signal whose sampling frequency is of 8 kHz is analyzed with frames of 32 ms, shifted by 10 ms. For each frame, relevant characteristics are extracted to form the acoustic vectors. We use Mel-cepstral coefficients since they prove to be efficient in speaker recognition. Concerning the dimension of the acoustic vectors, it may vary depending on the speaker-based indexing system stage. For instance, the use of the 12 first coefficients for the segmentation stage has proved its efficiency. By contrast, the addition of the first derivatives (Δ-coefficients) overloaded the computation, without improving performance.

4 Speaker-based segmentation

The aim is to detect the speaker changing points (scp). The segmentation technique we propose is organized in two passes. Our algorithm and its performance are described in detail in [5,6].

Distance-based segmentation The first pass of our segmentation technique relies on a distance-based segmentation. The measure function has to reflect how similar two adjacent segments are. A high value should indicate a change of speaker, whereas low values should signify that the two portions of signal correspond to a unique speaker. The Generalized Likelihood Ratio (GLR) presented in [7,8] proved to be the most efficient measure, showing high and narrow peaks at speaker change, and low variation in amplitude within single speaker segments.

Given $\mathcal{X} = \{x_1, ..., x_{N_\mathcal{X}}\}$ a sequence of $N_\mathcal{X}$ acoustic vectors, we assume that $\mathcal{X}$ is generated by a multi-Gaussian process $\mathcal{N}(\mu_\mathcal{X}, \Sigma_\mathcal{X})$ with mean $\mu_\mathcal{X}$ and full-covariance matrix $\Sigma_\mathcal{X}$, and consider the following hypothesis test for speaker change at time i:

- H_0: $(x_1, ..., x_{N_\mathcal{X}}) \sim \mathcal{N}(\mu_\mathcal{X}, \Sigma_\mathcal{X})$
- H_1: $(x_1, ..., x_i) \sim \mathcal{N}(\mu_{\mathcal{X}_1}, \Sigma_{\mathcal{X}_1})$ and $(x_{i+1}, ..., x_{N_\mathcal{X}}) \sim \mathcal{N}(\mu_{\mathcal{X}_2}, \Sigma_{\mathcal{X}_2})$

The likelihood ratio between the hypothesis H_0 and H_1 is defined by:

$$R = \frac{L(\mathcal{X}, \mathcal{N}(\mu_{\mathcal{X}}, \Sigma_{\mathcal{X}})}{L(\mathcal{X}_1, \mathcal{N}(\mu_{\mathcal{X}_1}, \Sigma_{\mathcal{X}_1})).L(\mathcal{X}_2, \mathcal{N}(\mu_{\mathcal{X}_2}, \Sigma_{\mathcal{X}_2}))} \quad (1)$$

where L denotes the likelihood. The GLR distance is computed from the logarithm of the previous expression: $d_{\text{GLR}} = -\log R$.

The GLR distance is computed for a pair of adjacent windows of the same size (about 2s), and the windows are then shifted by a fixed step (about 0.1s) along the complete parameterized speech signal. All the distance values are stored to form at the end of the process a distance plot where the significant peaks are detected since they correspond to the speaker changing points.

BIC refinement Even with a fine tuning of the detection parameters, the number of insertion errors (a scp is detected although it does not exist) remains high after the distance-based segmentation. A second pass using the Bayesian Information Criterion (BIC), also known as Minimum Description Length (MDL) (see [9]), is required to merge the segments corresponding to the same speaker, and thereby to decrease the number of insertion errors. The BIC applied to segmentation has been proposed by S.Chen in [10].

The BIC is a likelihood criterion penalized by the model complexity. With the same notations as before, the BIC value is determined by: $\text{BIC}(M) = \log L(\mathcal{X}, M) - \lambda \frac{m}{2} \log N_{\mathcal{X}}$, where $L(\mathcal{X}, M)$ is the likelihood of $\mathcal{X}$ for the model M, m is the number of parameters of the model M and λ the penalty factor. The first term reflects how good the model fits the data and the second term corresponds to the complexity of the model. Thus, the modeling which maximized this criterion is favored. The variations of the BIC value between the two models (one Gaussian function versus two different Gaussian functions) is then given by: $\Delta\text{BIC}(i) = -R(i) + \lambda P$ where $R(i)$ denotes the maximum likelihood ratio between hypothesis H_0 (no speaker change) and H_1 (speaker change at time i) and the penalty is given by $P = \frac{1}{2}(d + \frac{1}{2}d(d+1)) \log N_{\mathcal{X}}$, d being the dimension of the acoustic space, and λ is the penalty factor. A negative value of Δ-BIC(i) indicates that the two multi-Gaussian models best fit the data $\mathcal{X}$, which means that a change of speaker occurred at time i.

For each pair of segments delimited by the speaker changing points previously found (during the first pass), a Δ-BIC value is computed. If the value is negative, the speaker changing point separating both segments is validated and then a Δ-BIC value is computed for the next pair of segments. If not, the speaker changing point separating both segments is discarded and then, both segments are merged to form one segment from the next pair of segments.

5 Segment grouping

Once data have been segmented, the goal is to group segments with respect to their corresponding speaker. Segments related to a given speaker should be grouped in the same cluster and this cluster should be the only one corresponding to this given speaker. Two types of grouping methods may be considered: hierarchical clustering and sequential clustering. We now propose a review of the existing techniques.

Hierarchical clustering Hierarchical clustering is fully detailed in [11]. We describe here bottom-up clustering technique. At the beginning, each segment forms a cluster. At each iteration, the two clusters best satisfying a merging criterion are grouped. This process is repeated until a stopping criterion is reached. Two parameters of hierarchical clustering have to be set: the merging and the stopping criteria.

An obvious stopping criterion could have been the total number of clusters desired. In our context, the number of speakers is unknown so that the final number of clusters is also unknown.

The merging criterion should reflect how similar two clusters are. Since clusters are composed of segments of a parameterized signal, a distance as used during the segmentation step is a suitable merging criterion. The GLR and the cross likelihood ratio are used respectively in [12] and [13] as merging criteria. Then, the hierarchical clustering consists of merging iteratively all the segments to form a unique cluster and storing at each iteration the pair of merged cluster (the two closest according to the given distance) to build a "merging tree". The algorithm is stopped when a unique cluster is obtained. The problem is now to choose an optimal partition of clusters from the resulting merging tree. D.A. Reynolds et al. [13] propose two methods to select the best partition of the candidate clusters.

Another possible merging criterion is a penalized distance. Two clusters are merged if they are the two closest among the set of clusters and if the penalized distance between them satisfies a criterion associated to its penalty. The complexity of the segment models and the number of segments and of

clusters are used as penalty factors respectively in [10] and [1]. Therefore, the stopping criterion is easily determined: the hierarchical clustering process stops when the distance between the two closest clusters does not satisfy its associated penalty criterion.

One can reproach hierarchical clustering algorithms not to take into account neighboring relationships between segments. Indeed, in some audio documents, the intra-speaker variability may be as high as in the phone conversations of SWITCHBOARD. However, this variability may be progressive: for example, at the beginning of the conversation, the speaker may be tense or restrained and as the conversation goes along, the person relaxes. In such a case, it is of interest to use sequential clustering as described in the next section.

Sequential clustering The principle behind sequential clustering is to use the neighboring relationships between segments. Rather than merging two segments of a given speaker respectively located at the beginning and at the end of the conversation, we might prefer to merge segments related to a given speaker which are separated by a small time interval only. Indeed, the intra-speaker variability between segments located at times far away from each other may be high. The sequential clustering acts as follows: all segments resulting from the segmentation step are ordered with respect to time. A first cluster is initialized with the first segment. Then, for each next segment, the following procedure is repeated: if the segment considered can be merged with a cluster according to a merging criterion, then it is added to the given cluster and the characteristics of the resulting cluster are updated. If the segment considered can not be added to any cluster, then a new cluster is created and initialized with this segment. The stopping criterion is obvious: the sequential clustering process stops when the last segment (with respect to time) has been processed. The merging criterion should be similar to a penalized distance: the two segments should be close enough to be merged.

M.Nishida and Y.Ariki [14] proposes a similar method although they perform segmentation and clustering concurrently. They model each cluster by a subspace. This speaker-subspace is deduced from the singular value decomposition (SVD) of the matrix formed by acoustic vectors in the cluster. If the projection of the acoustic vectors of a segment onto the speaker-subspace which maximises the norm of this projection is large enough, then the segment is added to this speaker-subspace. Otherwise, a new speaker is created from the acoustic vectors of this segment. By performing sequential clustering, the intra-speaker variability is taken progressively into account as the clustering process goes on.

Implementation and testing of hierarchical and sequential clustering techniques are in progress in our lab at the present time.

6 Speaker modeling and recognition

The aim of the final step is to build a speaker model for each resulting cluster (if it has not been already done during the clustering process) and to use these models to recognize the sequence of speakers involved in the conversation. As mentioned earlier, speaker models can be more sophisticated (like GMMs) since more data are available. Then, the speaker models are used to recognize the sequence of speakers. In parallel, a refinement of the segmentation is operated. For each segment resulting from the segmentation step, identification scores related to each speaker model are computed. A large score of a segment according to a speaker model signifies that the segment has been pronounced by the corresponding speaker. If the best score is large enough, then the segment is identified. Otherwise, there are two alternatives. First, the segment is contaminated: it contains two or more speakers. Then, the strategy to adopt consists in splitting the segment considered into smaller parts and then to repeat the identification process on all these small sub-parts. Thus, segmentation is refined. Second, the segment is related to one speaker only but no model corresponds to this speaker (an error occurred during the clustering step). Thus, even the smaller parts of the segments can not be identified and labeled. Then, as in a typical "open set" identification process (see [15]), the segment is "rejected": it is not assigned any label.

7 Evaluation methods

In this section, evaluation methods for each step of the complete indexing task are proposed or reviewed. We decide to evaluate the different stages separately for several reasons. Firstly, a global evaluation of the complete indexing system does not point out which module is involved when an error occurs. Secondly, the segmentation stage and the grouping stage can be considered as independent entities. Each of

them can be reused in other applications, as mentioned in the introduction. Therefore, they are assessed separately. And in our context, it allows to know the contribution of each stage. Finally, the evaluation of the final stage, the recognition stage, also provides an assessment of the complete system.

Segmentation A good segmentation should provide the correct speaker changes and therefore segments containing a single speaker. We distinguish two types of errors related to speaker change detection. An *insertion error* or *false alarm* (FA) occurs when a speaker change is detected although it does not exist. A *deletion error* or *missed detection* (MD) occurs when the process does not detect an existing speaker change. In our context, a missed detection is more severe than a false alarm. Indeed, a missed detection may damage the grouping step: a "corrupted" segment (containing two or more speakers) will contaminate the cluster it is attached to. By contrast, false alarms may be resolved during the grouping step: if the utterances of a given speaker have been split in several segments, then they will be grouped in the same cluster during the grouping step. We can then define the false alarm rate (FAR) and the missed detection rate (MDR) where 'scp' denotes 'speaker changing point':

$$\text{FAR} = \frac{\text{number of FA}}{\text{number of actual scp} + \text{number of FA}} \text{ and MDR} = \frac{\text{number of MD}}{\text{number of actual scp}}$$

Grouping The partition resulting from a clustering process should meet two conditions to be efficient. First, the number of resulting clusters should be correct (when it is unknown as in our context). Second, each cluster should represent a single speaker and all the segments related to this given speaker should be in this cluster. In other words, the cluster should be pure (i.e. not contaminated).

The number of clusters should be equal to the number of actual speakers involved in the conversation. We can distinguish two types of errors. First, the cluster number is larger than the speaker number: a speaker may be represented by different clusters. Second, the cluster number is smaller than the speaker number: some (at least one) clusters contain segments related to different speakers.

A.Solomonoff et al. [12] proposes different measures to score the quality of a partition resulting from a clustering process. Especially, the authors define the purity of cluster i as: $p_i = \sum_j \frac{n_{ij}^2}{n_{i.}^2}$ where n_{ij} denotes the number of segments in cluster i that were spoken by speaker j and $n_{i.} = \sum_{j=1}^{N_s} n_{ij}$ is the size of cluster i (N_s is the total number of speakers). Purity is a quantity which describes to what extent all segments in the cluster come from the same speaker. If all segments from a cluster are generated by the same speaker, then $p_i = 1$. And the more speakers a cluster contains, the more purity decreases. More precisely, p_i represents the probability that two segments picked at random from cluster i are generated by the same speaker as explained in [12].

Recognition Concerning the recognition step, which is the final step, we choose to evaluate this step in terms of success rate defined by (where a frame denotes an acoustic vector):

$$\text{success rate} = \frac{\text{number of frames correctly attributed}}{\text{total number of frames}}$$

Since a frame may not be labeled, we define the rate of frames non labeled as the ratio between the number of frames non labeled and the total number of frames. It is of interest to also define these both rates for each speaker involved in the conversation.

Comments A reference segmentation is required for using this kind of error definitions. However, its accuracy, when the reference segmentation exists, may be low since the perception of speaker changes is sometimes subjective.

8 Results

We present only partial results since some experiments are still in progress in our lab.

8.1 Data

We worked on several types of data

- 2 conversations which are artificially created by concatenating sentences of 2 s on average from the TIMIT database (clean speech, short segments).
- 2 conversations created by concatenating sentences of 1 to 3 s from a French language database provided by CNET (Centre National d'Etudes des Télécommunnications) (clean speech, short segments).

- 3 TV news broadcasts extracted from the database provided by INA (Institut National de l'Audiovisuel) in French language (segments of any length).
- 3 phone conversations extracted from SWITCHBOARD ([16]) database (segments of any length, spontaneous speech).
- 4 French TV news broadcasts (referred to as *jt*) collected in our lab (segments of any length).

8.2 Segmentation stage

Results of the segmentation stage are reported in table 1 for the different types of data. The false alarm rate (FAR) and the missed detection rate (MDR) are given for the first pass, the distance-based segmentation and for the second pass, the BIC refinement. In [6], we compare our segmentation technique to an other one proposed by S.Chen ([10]). Applied on conversations containing segments of any length, both techniques provide comparable results. However, our technique gives better results with conversations containing short segments (1 to 3 seconds). A qualitative study of error occurrences can also be found in [6].

	1^{rst} pass		2^{nd} pass	
	FAR	MDR	FAR	MDR
TIMIT	40.3	14.3	28.2	15.6
CNET	18.2	16.7	16.9	21.4
INA	37.4	9.03	18.5	13.5
SWITCHBOARD	39.0	29.1	25.9	29.1
JT	59.0	8.9	23.7	9.4

Table 1. FAR and MDR respectively with the first and the second pass of the segmentation stage

The problem of speech over music or over noise may be of less importance than for transcribing tasks. It seems to be easier to recognize all the utterances of a given person speaking over music than to recognize what this person says. But, the problem still remains if this person speaks sometimes over music, sometimes without background noise. It will likely result in a false alarm during the segmentation stage and also in an error during the grouping stage.

8.3 Grouping stage

First experiments on sequential clustering (method proposed by Y.Ariki [14]) applied to conversations containing short segments (1 to 3 seconds) do not give expected results. Further experiments are required to come up with a robust algorithm.

9 Conclusion and further work

In this paper, we propose an architecture for a speaker-based indexing system. For each stage, we detail the aim, the constraints and we review possible techniques to perform the associated task. We also propose assessment methods for each stage. Some stages have already been implemented in our lab and some are in progress. Our further work will clearly consist in implementing and tying all the stages to form the complete indexing system and to test its performances on large databases.

References

1. Jean-Luc Gauvain, Lori Lamel, and Gilles Adda. Partitioning and transcription of broadcast news data. In *ICSLP*, 1998.
2. P.C. Woodland and al. The development of the 1996 HTK broadcast news transcription system. In *DARPA speech recognition workshop*, 1997.
3. Aaron E. Rosenberg and al. Speaker detection in broadcast speech databases. In *ICSLP*, 1998.
4. L.R. Rabiner and R.W. Schafer. *Digital processing of speech signals*. Prentince-Hall, 1978.
5. Perrine Delacourt, David Kryze, and Christian J. Wellekens. Speaker-based segmentation for audio data indexing. In *ESCA workshop: accessing information in audio data*, 1999.

6. P. Delacourt, D. Kryze, and C.J. Wellekens. Detection of speaker changes in an audio document. In *EUROSPEECH*, 1999.
7. Herbert Gish, Man-Hung Siu, and Robin Rohlicek. Segregation of speakers for speech recognition and speaker identification. In *ICASSP*, pages 873–876, 1991.
8. Herbert Gish and N. Schmidt. Text-independent speaker identification. *IEEE signal processing magazine*, oct. 1994.
9. Jorma Rissanen. *Stochastic complexity in statistical inquiry*, volume 15 of *Series in Computer Science*, chapter 3. World Scientific, 1989.
10. S.S. Chen and P.S. Gopalakrishnan. Speaker, environment and channel change detection and clustering via the Bayesian information criterion. In *DARPA speech recognition workshop*, 1998.
11. R.O. Duda and P.E. Hart. *Pattern classification and scene analysis*. John Wiley and Sons, Inc., 1973.
12. A. Solomonoff and al. Clustering speakers by their voices. In *ICASSP*, 1998.
13. D.A. Reynolds and al. Blind clustering of speech utterances based on speaker and language characteristics. In *ICSLP*, 1998.
14. M. Nishida and Y. Ariki. Real time speaker indexing based on subspace method: applications to TV news articles and debate. In *ICSLP*, 1998.
15. Sadaoki Furui. An overview of speaker recognition technology. In *ESCA workshop on automatic speaker recognition, identification and verification*, 1995.
16. J.J Godfrey and al. SWITCHBOARD: telephone speech corpus for research and development. In *ICASSP*, 1992.