UNSUPERVISED SPEAKER CHANGE DETECTION FOR BROADCAST NEWS SEGMENTATION

Kasper Jørgensen, Lasse Mølgaard, and Lars Kai Hansen

Informatics and Mathematical Modelling, Technical University of Denmark
Richard Petersens Plads, Building 321, DK-2800 Kongens Lyngby, Denmark
phone: +(45) 4525 3889, fax: +(45) 4587 2599, email: s001498,s001514,lkh@imm.dtu.dk,
web: http://isp.imm.dtu.dk

ABSTRACT

This paper presents a speaker change detection system for news broadcast segmentation based on a vector quantization (VQ) approach. The system does not make any assumption about the number of speakers or speaker identity. The system uses mel frequency cepstral coefficients and change detection is done using the VQ distortion measure and is evaluated against two other statistics, namely the symmetric Kullback-Leibler (KL2) distance and the so-called 'divergence shape distance'. First level alarms are further tested using the VQ distortion. We find that the false alarm rate can be reduced without significant losses in the detection of correct changes. We furthermore evaluate the generalizability of the approach by testing the complete system on an independent set of broadcasts, including a channel not present in the training set.

1. INTRODUCTION

The increasing amount of audio data available via the Internet emphasizes the need for automatic sound indexing. Broadcast news and other podcasts often include multiple speakers in widely different environments. Efficient indexing of such audio data will have many applications in search and information retrieval. Segmentation of sound streams is a significant challenge including segmentation of sequences of music and different speakers. Locating parts that contain the same speaker in the same environment can indicate story boundaries and may be used to improve automatic speech recognition performance. Indexing based on speaker recognition is a possibility but is hampered by the prevalence of unknown speakers, thus we have chosen to investigate unsupervised methods in this work in line with other recent systems, see e.g., [1]. Here we are interested in systems that are not too specialized to a given channel, hence, in both system design and in the evaluation procedure we will focus on the issue of robustness. In particular we show that a system can be tuned to a set of channels and not only generalize to other broadcasts from these channels, but also to a channel not present in the training set.

Speaker change detection approaches can roughly be divided into three classes: energy-based, metric-based and model-based methods. Energy-based methods rely on thresholds on the audio signal energy, placing changes at 'silence' events. In news broadcast the audio production can be quite aggressive with only little if any silence between speakers, making this approach less attractive.

Metric based methods basically measure the difference between two consecutive frames that are shifted along the audio signal. A number of distance measures have been investigated such as the symmetric Kullback-Leibler distance [2]. Parametric models corrected for finite samples using the Bayesian Information Criterion (BIC) are also widely used. Huang and Hansen [3] argued that BIC-based segmentation works well for longer segments, while BIC approach with a preprocessing step that uses a T^2-statistic to identify potential changes, was superior for short segments.

Nakagawa and Mori [4] compare different methods for change detection, including BIC, Generalized Likelihood Ratio, and a vector quantization (VQ) based distortion measure. The comparison indicates that the VQ method is superior to the other methods.

A simplification of the Kullback-Leibler distance, the so-called divergence shape distance (DSD), was presented in [1] for a real-time implementation. The system includes a method for removing false positives using "lightweight" GMM speaker models.

Model-based methods are based on recognizing specific known audio objects, e.g., speakers, and classify the audio stream accordingly. The model-based approach has been combined with the metric-based to obtain hybrid-methods that do not need prior data [5][6].

Our basic sound representation is the mel-weighted cepstral coefficients (MFCC), they have shown useful in a wide variety of audio application including speech recognition, speaker recognition [7] and music modelling, see e.g., [8].

Since we are interested in segmenting news with an unknown group of speakers we limit our investigation to metric based methods. To improve the performance we invoke a false alarm compensation step at relative low additional cost.

2. DISTANCE MEASURES

Metric based change detection is done by calculating a distance between two successive windows. The distance indicates the similarity between the two windows. Below we present three different distance measures that have been considered in this context.

2.1 Vector Quantization Distortion

The VQ approach is based on the generalized distance between two feature vectors sequences designated S^A and S^B.

The VQ-distortion measure VQD between S^B and the codebook C^A, created by clustering of the features in S^A, is defined as:

$$\mathrm{VQD}(C^A, S^B) = \frac{1}{T}\sum_{t=1}^{T} \arg\min_{1\le k\le K} \left\{ d\left(C_k^A, S_t^B\right) \right\},$$

where C_k^A denotes the k-th code-vector in C^A, $1 \leq k \leq K$. S_t^B denotes the t-th feature vector in the sequence S^B, $1 \leq t \leq T$, and d is the *Euclidean* distance function, see e.g., [1].

The codebook C^A is created by clustering the sequence of feature vectors S^A into K clusters, thus each cluster-center represents a code-vector.

2.2 Kullback-Leibler Distance

The symmetric Kullback-Leibler distance (KL2) has been used in speaker identification systems and applied to speaker change detection [9]. The symmetric Kullback-Leibler distance between two audio segments represented by their feature vector sequences S^A and S^B is defined as:

$$\mathrm{KL2}(S^A, S^B) = \int_{\boldsymbol{x}} [p_A(\boldsymbol{x}) - p_B(\boldsymbol{x})] \log \frac{p_A(\boldsymbol{x})}{p_B(\boldsymbol{x})} d\boldsymbol{x} \quad (1)$$

Assuming that the feature sequences S^A and S^B are n-variate Gaussian distributed, $p_A \sim \mathcal{N}(\boldsymbol{\mu}_A, \boldsymbol{\Sigma}_A)$, $p_B \sim \mathcal{N}(\boldsymbol{\mu}_B, \boldsymbol{\Sigma}_B)$, i.e.

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\Big\{ -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \Big\} \quad (2)$$

Combining equation (1) and (2) gives:

$$\begin{aligned} \mathrm{KL2}(S^A, S^B) &= \frac{1}{2}\mathrm{Tr}\Big[(\boldsymbol{\Sigma}_A - \boldsymbol{\Sigma}_B)(\boldsymbol{\Sigma}_B^{-1} - \boldsymbol{\Sigma}_A^{-1})\Big] \\ &\quad + \frac{1}{2}\mathrm{Tr}\Big[(\boldsymbol{\Sigma}_A^{-1} + \boldsymbol{\Sigma}_B^{-1})(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B) \\ &\quad (\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)^\top\Big] \end{aligned}$$

2.3 Divergence Shape Distance

The KL2 distance presented above is composed of two terms. The last term depends on the means of the features which can vary much depending on the environment [1]. Using only the first term should remove this dependency, so that only the difference between covariance contribute. This function is called the divergence shape distance (DSD).

$$\mathrm{DSD}(S^A, S^B) = \frac{1}{2}\mathrm{Tr}\Big[(\boldsymbol{\Sigma}_A - \boldsymbol{\Sigma}_B)(\boldsymbol{\Sigma}_B^{-1} - \boldsymbol{\Sigma}_A^{-1})\Big]$$

In all of the three presented distance measures a greater value means a greater difference in the two distributions.

3. SPEAKER CHANGE DETECTION

Based upon the distance metric the change detection algorithm determines whether or not a speaker change occurred.

Our algorithm works in two steps. The first step is the change-point detection part where candidate change-points are found. The second step is the false alarm compensation step.

3.1 Front-End Processing

MFCCs are chosen as the features for this work. The calculation of these features is preceded by transforming the audio streams to a common sampling and bitrate.

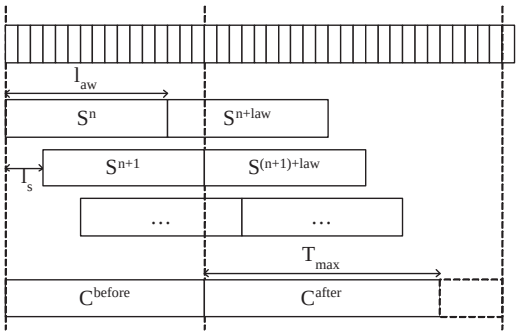

Figure 1: Illustration of windows used in the metric calculation. Speaker change-points are indicated with vertical dashed lines. The figure assumes that a change is found at time t_{n+1}, and false alarm compensation windows are shown at the bottom

3.2 Distance Metric Calculation

The audio is divided into analysis windows of length l_{aw} and with a shift of length l_s, see figure 1. Let S^n denote the sequence of feature vectors extracted from the analysis window with endtime t_n. Then, S^n and $S^{n+l_{aw}}$ are two succeeding and non-overlapping analysis windows.

For each feature vector sequence S^n a codebook C^n is created by clustering the vector sequence into K clusters using the *k-means* clustering algorithm. Convergence of the *k-means* algorithm is sped up by exploiting the overlap of the analysis windows, which means that most samples are reused in subsequent analysis windows. The code-vectors of C^n are therefore computed using the code-vectors from C^{n-l_s} as initial cluster centers. This makes the *k-means* algorithm converge faster and minimizes the distance between two succeeding codebooks, resulting in less fluctuating distortion measures.

The conventional VQ-algorithm computes the distortion measure between two feature vector sequences S^A and S^B by computing $\mathrm{VQD}(C^A, S^B)$. By using the code-vectors of C^B instead of the whole sequence S^B, better results are obtained. Thus, we use $\mathrm{VQD}_n = \mathrm{VQD}(C^{S^n}, C^{S^{n+law}})$ as the VQ-distortion measure at time t_n.

The $\mathrm{KL2}_n$ and DSD_n at time t_n are given by $\mathrm{KL2}_n = \mathrm{KL2}(S^n, S^{n+l_{sw}})$ and $\mathrm{DSD}_n = \mathrm{DSD}(S^n, S^{n+l_{sw}})$

3.3 Change-Point Detection

The basic change-point detection evaluates the calculated distance metric M_n at every time step time (t_n). A change-point is found if M_n is larger than a threshold th_{cd} and M_n is the local peak within T_i seconds. The intention of this baseline approach is to detect as many true change-points as possible. The false alarms that occurs should then be rejected by our false alarm compensation described below.

3.4 False Alarm Compensation

When running the speaker change-point detection algorithm it is necessary to keep the analysis window relatively short in order to be able to detect short speaker turns. The short segments may lack data to make fully reliable segment models, which consequently may cause false alarms.

The baseline approach yields a number of potential change-points, dividing the audio stream into speaker seg-

ments. These speaker segments can then be used to make more accurate models between the potential change-points. Comparing these models can then accept or reject the potential change-point.

The false alarm compensation algorithm simply works by making two speaker VQ-codebooks, for the speaker segment before the change-point C^{before} and another after the change-point C^{after}.

The two VQ-distortion measures $VQD(C^{before}, C^{after})$ and $VQD(C^{after}, C^{before})$ are computed and the mean VQD_{mean} of these two measures is found. The change-point is then accepted if the measure is larger than the threshold th_{fac} and rejected if it is below. We found that using the mean of the two distortion measures is more stable than using just one of the measures.

If a real speaker change is missed during the initial change-point detection, the resulting speaker model would contain data from two speakers, meaning that the speaker codebook models both speakers. To counteract this problem only the T_{max} seconds nearest the change-point is used to make the speaker codebook.

3.5 Parameter Settings

The proposed change-point detection algorithm requires some parameters to be adjusted. The two thresholds th_{cd} and th_{fac} should be set according to the desired relation between recall and precision. As in [1] we use an automatic threshold setting method. We use $M_{n,mean}$ as the mean of the distance metric in a window of $2T_{max}$ around t_n:

$$M_{n,mean} = \frac{1}{2T_{max}+1}\sum_i M_{n+i},$$

with $-T_{max}/l_s < i < T_{max}/l_s$. The thresholds at time t_n are thereby set to:

$$\begin{aligned} th_{cd,n} &= \alpha_{cd} M_{n,mean} \\ th_{fac,n} &= \alpha_{fac} M_{n,mean} \end{aligned}$$

The two amplifiers α_{cd} and α_{fac} should be set in advance.

The timing parameters l_{aw}, T_i, and T_{max} should be set according to the expected distribution of speaker turn lengths. l_s defines the resolution of the detected change-points.

3.6 Example

An example of the change-point detection algorithm is shown in figure 2. The audio clip in this example is 113*s* long and contains speaker change-points at time $t = \{14.6, 29.3, 33.7, 43.8, 63.5, 78.9\}s$ indicated by the vertical lines. The upper part of the figure shows the VQ-distortion measure VQD_n as function of time. The dotted line indicate the threshold th_{cd} and the estimated change-points found by our change-point algorithm are shown with circles. It is seen that in addition to the true speaker change-points four false false alarms occur.

The lower part of the figure shows the VQ-distortion measure VQD_{mean} for the found change-points. Again, the dotted line indicate the threshold th_{fac} and the accepted change-points are shown by circles, and the rejected are shown by crosses.

In this example all the true speaker changes are found, and false alarms are removed by the false alarm compensation step.

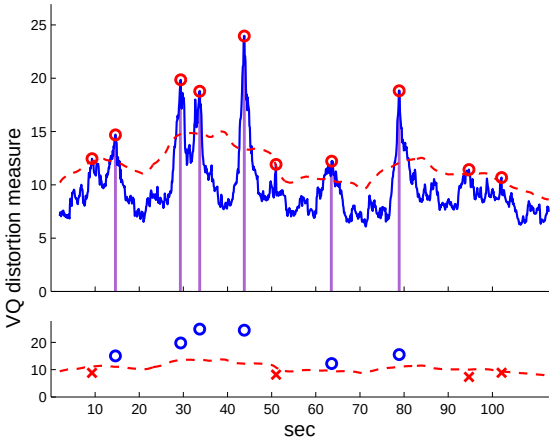

Figure 2: The upper part of the figure shows the VQ-distortion measure VQD_n for a sample file. The true speaker changes are indicated by vertical lines. The dotted line indicates the threshold th_{cd} and the estimated change-points found are shown with circles. In addition to the true speaker change-points four false change-points are found. The lower part of the figure shows the VQ-distortion VQD_{mean} for the found change-points. The threshold th_{fac} is indicated and the accepted change-points are shown by circles, and the rejected are shown by crosses.

4. EXPERIMENTS AND RESULTS

4.1 Speech Database

The speech data used was news-podcasts obtained from four different news/radio channels CNN, CBS, WNYC, and PRI.

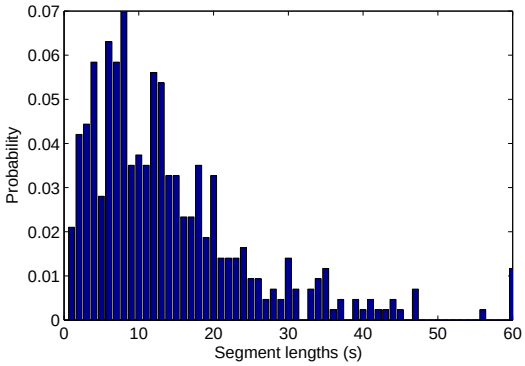

Figure 3: Histogram of the speaker segment lengths contained in the database.

The data consists of 103 min of broadcast news, which contains speech from numerous speakers, in different environments. Music has been removed as this is assumed to be done using a music/speech discriminator. The length of the segments range from 0.4*s* to 119*s* with a mean of approximately 14*s*. Figure 3 shows the distribution of the segment lengths. The number of speaker changes is 388, distributed over 47 files. The data was manually labelled into different speakers. The number of segments is 435, and 75 of these have a length less than 5*s*, which are segments considered relatively hard to detect [1, 3].

	Total length (min)	Avg. segment length (sec)	Speaker changes
CNN	38	17.0	134
CBS	20	9.9	121
WNYC	26	22.6	69
PRI	19	15.8	64
All	103	15.6	388

Table 1: Summary of evaluation data.

4.2 Feature Extraction

First all files have been down-sampled to 16kHz, 16bit mono channel. The MFCCs are extracted on a 20 ms Hamming filtered window. The windows overlap by 10 ms. The feature vector consists of 12 MFCCs. 'delta-MFCCs' or 'delta-delta-MFCCs' were not included because they worsened segmentation results. The features are not normalized.

4.3 Evaluation Measures

A change-point proposed by the algorithm may not be precisely aligned with the manual label. For example if the change occurs at a silence period or if speakers interrupt each other. To take this into account, a found change is counted as correct if it is within 1*s* of the manually labelled change-point, as in [3]. The *mismatch* is defined as the time between a correct found change-point point and the manually labelled one.

The evaluation measures frequently used are recall (RCL) and precision (PRC), that correspond to deletions and insertions respectively.

$$\text{RCL} = \frac{\text{no. of correctly found change-points}}{\text{no. of true change-points}}$$

$$\text{PRC} = \frac{\text{no. of correctly found change-points}}{\text{no. of hypothesized change-points}}$$

The F-measure combines RCL and PRC into one measure,

$$\text{F} = \frac{\text{RCL} \times \text{PRC}}{\alpha \times \text{RCL} + (1-\alpha)\text{PRC}}$$

with α as a weighting parameter that can be used to emphasize either of the two quantities. The results presented below use the equal weighting, with $\alpha = 0.5$.

4.4 Results

This section will present the results obtained with our speaker change detection algorithm. The length of the analysis window is set to $\text{l}_{\text{aw}} = 3s$. T_{i} is set to 2*s* and T_{max} is set to 8*s*. The analysis windows are shifted with $\text{l}_{\text{s}} = 0.1s$.

Table 2 shows the results obtained using all the data from our database. α_{cd} and α_{fac} are set to maximize the F-measure after the false alarm compensation (FAC). The VQ-approach is evaluated using 24, 48, 56, and 64 clusters for both the change detection and in the false alarm compensation. In the KL2-FAC and DSD-FAC approaches, 56 clusters are used.

Comparing the results using the VQD measure the best performance is obtained using 56 clusters. In this case 80.1% of the true change-points are detected with a false alarm rate of 8.5 %. A relative improvement of 59,7% in precision with a relative loss of 7.2% in reduction is obtained with our false alarm compensation scheme.

By varying α_{cd} a recall-precision curve can be created. Figure 4 shows the recall-precision curve for the three metrics VQD-56, KL2, and DSD for the baseline algorithm. The curves for VQD-56 and KL2 are comparable, though VQD-56 gives better precision at lower recall. VQD-56 and KL2 is clearly better than DSD.

Figure 5 shows the recall-precision curves after the false alarm compensation. This curve is created by varying α_{cd} and keeping α_{fac} constant. Though, the baseline recall-precision curve for VQD and KL2 is very similar the VQD-FAC performs better than KL2-FAC. A reason for this could be that VQD and KL2 do not locate the same change-points and FAC then rejects more true change-point found by KL2 than found by VQD.

The change-points are found with a relatively small average mismatch of approximately 0.2*s*, which is acceptable for most applications.

An investigation reveals that approximately 62% of the missed change points are due to segments that are shorter than 5*s*.

Metric	F	RCL	PRC	Mismatch
VQD24	0.748	0.810	0.695	209ms
VQD24-FAC	0.829	0.740	0.943	206ms
VQD48	0.717	0.840	0.627	208ms
VQD48-FAC	0.839	0.766	0.928	206ms
VQD56	0.687	0.863	0.573	220ms
VQD56-FAC	0.854	0.801	0.915	202ms
VQD64	0.722	0.835	0.637	202ms
VQD64-FAC	0.837	0.789	0.892	215ms
KL2	0.763	0.833	0.704	212ms
KL2-FAC	0.823	0.789	0.860	212ms
DSD	0.623	0.766	0.526	308ms
DSD-FAC	0.732	0.665	0.814	288ms

Table 2: Results obtained with α_{cd} and α_{fac} adjusted to optimize the F measure after the false alarm compensation (FAC). Both the results before and after the FAC is shown.

4.5 Generalizability

To investigate the generalizability of our system, another test was set up where the database was divided into a training set and four test sets. The training set contains files randomly chosen from three of the channels, CNN, CBS, and WNYC. Four test sets were created, one for each of the channels, using the remaining files in the database.

The system was set up using the VQD measure with 56 clusters. The system parameters α_{cd} and α_{fac} were optimized for the training set and then evaluated on the test sets. Figure 6 shows the F-measure for this test. The results are compared with the system optimized for each of the specific test sets.

Generally our system performs better on the two test sets CNN and CBS compared to WNYC and PRI. This is most likely due to the fact that WNYC and PRI contain more short segments (<3*s*) than CNN and CBS. The analysis window length of 3*s* makes these segments hard to locate.

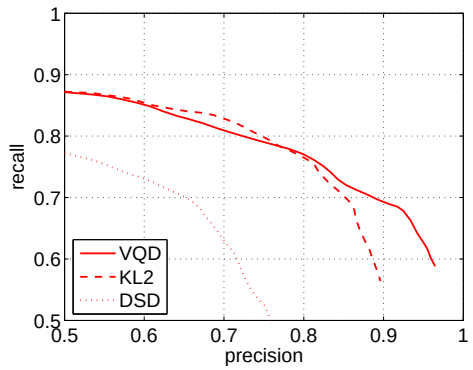

Figure 4: Recall-precision curve for baseline algorithm with the three distance metrics VQD, KL2, and DSD. The curve is created by varying α_{cd}. VQD and KL2 are superior to the DSD measure. VQD gives a better precision at lower recall rates.

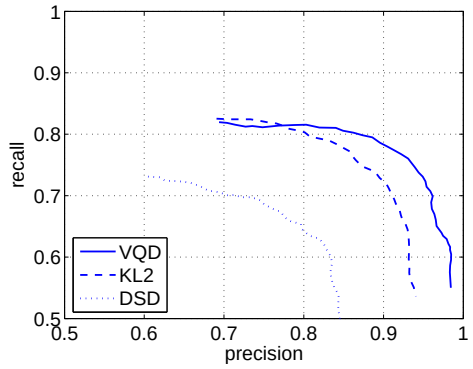

Figure 5: Recall-precision curve after the false alarm compensation with the three distance metrics VQD, KL2, and DSD. The curve is created by varying α_{cd} and keeping α_{fac} constant.

Only a minor reduction in the F-measure for all test sets is observed when using the training setting compared to the optimal settings for these test sets. Even the data from PRI that was not present in the training set show the same behavior. This demonstrates that the system is robust and lend support to the use in different media without need for further supervised tuning of parameters for new channels.

5. CONCLUSION

We have outlined an approach for robust segmentation of broadcast news. Fully implemented such a system could enable search in a broader media base than current web search engines. We have emphasized the need for an unsupervised approach because only a fraction of the speakers can be known a priori in realistic news cast. We obtained state-of-the-art performance using a vector quantization distance measure. The vector quantization approach showed better performance than systems based on the symmetric KL distance and the so-called 'divergence shape distance'. We showed that the choice of system parameters based on one data set generalized well to other independent data sets, including data from a different channel. We showed that the false alarm rate can be significantly reduced using a post-processing step on the alarms suggested by the vector quantizer.

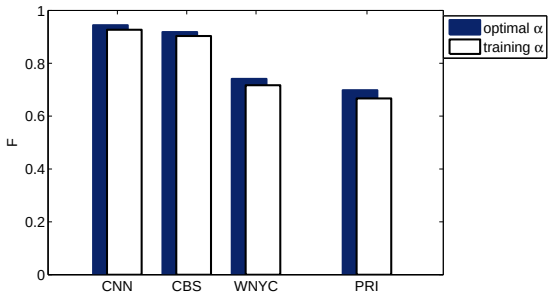

Figure 6: This figure shows the results obtained for different test sets. The system optimized for each of the tests are compared with a system optimized for a training set. The figure shows that a threshold chosen on a training set generalize reasonable well to other data sets.

Acknowledgments

This work is supported by the Danish Technical Research Council, through the framework project 'Intelligent Sound', www.intelligentsound.org (STVF No. 26-04-0092).

REFERENCES

[1] L. Lu and H. Zhang, "Unsupervised speaker segmentation and tracking in real-time audio content analysis," *Multimedia Systems*, vol. 10, no. Issue.4, pp. 332–343, 2005.

[2] M. Siegler, U. Jain, B. Raj, and R. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," *DARPA Speech Recognition Workshop*, pp. 97–99, 1997.

[3] R. Huang and J. H. Hansen, "Advances in unsupervised audio segmentation for the broadcast news and ngsw corpora," *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004*, vol. 1, pp. 741–744, May 2004.

[4] S. Nakagawa and K. Mori, "Speaker change detection and speaker clustering using vq distortion measure," *Systems and Computers in Japan*, vol. 34, no. 13, pp. 25–35, 2003.

[5] T. Kemp, M. Schmidt, M. Westphal, and A. Waibel, "Strategies for automatic segmentation of audio data," *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, vol. 3, pp. 1423–1426, 2000.

[6] H.-G. Kim, D. Ertelt, and T. Sikora, "Hybrid speaker-based segmentation system using model-level clustering," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, 2005.

[7] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various mfcc implementations on the speaker verification task," in *10th International Conference on Speech and Computer, SPECOM 2005*, vol. 1, (Patras, Greece), pp. 191–194, oct 2005.

[8] A. Meng, P. Ahrendt, and J. Larsen, "Improving music genre classification by short-time feature integration," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. V, pp. 497–500, mar 2005.

[9] H. Meinedo and J. Neto, "Audio segmentation, classification and clustering in a broadcast news task," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, vol. 2, pp. 5–8, IEEE, 2003.