Automatic segmentation of speakers in broadcast audio material

Hyoung-Gook Kim, Thomas Sikora
Communication Systems Group, Technical University of Berlin
{kim, sikora}@nue.tu-berlin.de

ABSTRACT

In this paper, dimension-reduced, decorrelated spectral features for general sound recognition are applied to segment conversational speech of both broadcast news audio and panel discussion television programs. Without a priori information about number of speakers, the audio stream is segmented by a hybrid metric-based and model-based segmentation algorithm. For the measure of the performance we compare the segmentation results of the hybrid method versus metric-based segmentation with both the MPEG-7 standardized features and Mel-scale Frequency Cepstrum Coefficients (MFCC). Results show that the MFCC features yield better performance compared to MPEG-7 features. The hybrid approach significantly outperforms direct metric based segmentation.

Keywords: MPEG-7, Metric-Based and Model-Based Segmentation, Mel-scale Frequency Cepstrum Coefficients (MFCC)

1. INTRODUCTION

Our challenge is the design of automatic speaker segmentation algorithms that enable applications such as speech recognition or word spotting in TV broadcast environments.
In many applications, like indexing of broadcast news, the input data consists of a continuous, unsegmented audio stream. For such applications, effective methods able to segment continuous audio streams into homogeneous segments are required.

Previous research has focused on two directions. In [1][2][3], distance-based segmentation approaches are investigated. The audio stream is segmented at maxima of the distance between neighboring windows placed in evenly spaced time intervals. Then, segments belonging to the same speaker are clustered using a distance measure that measures the similarity of two segments. The advantage of this method is that it does not require any a priori information. However, since the clustering is based on distances between individual segments, accuracy suffers from segments that are too short to sufficiently describe the characteristics of a speaker. In [4][5][6], an alternative approach is investigated, which is model-based. For every speaker in the audio recording, a model is trained, and then an HMM segmentation is performed to find the best time-aligned speaker sequence. This method places the segmentation under a global maximum likelihood framework. However, most model-based approaches require a priori information to initialize the speaker models.
In [7][8], it is shown that the hybrid algorithm, which combines metric-based and model-based techniques, works significantly better than all other approaches. Therefore, our segmentation system focuses on hybrid metric-based and model-based segmentation using MPEG-7 sound recognizer on the broadcast news data and panel discussion television programs.

MPEG-7 [9] standard, formally named “Multimedia Content Description Interface”, focuses on describing the content for indexing, searching, and retrieval of the audiovisual information. Recently, a MPEG-7 audio standard group [10] [11] has adapted a feature extraction method based on the projection of a spectrum onto a low-dimensional subspace via reduced-rank spectral basis functions. The dimension-reduced decorrelated features, called Audio Spectrum Projection (ASP), are used to train hidden Markov models (HMM) in order to apply uniformly to diverse source classification tasks with accurate performance.

In [12], we implemented and analyzed the spectrum basis projection features based on the MPEG-7 standard for the purpose of speaker recognition system with 25 speakers (11 males and 14 females). The spectrum basis functions were computed from the Normalized Audio Spectrum Envelope (NASE) using a basis decomposition algorithm such as Principal Component Analysis (PCA) [13] and FastICA [14]. Our results showed that the system achieves around 93.6 % recognition accuracy in the speaker recognition and 100 % recognition accuracy in the gender recognition test.

In this paper, the ASP features for general sound recognition are applied to segment conversational speech of both broadcast news audio and panel discussion television programs. For the measure of the performance we compare the segmentation results of the hybrid method versus metric-based segmentation with both the MPEG-7 ASP features and Mel-scale Frequency Cepstrum Coefficients (MFCC).

2. SYSTEM OVERVIEW

Our hybrid segmentation using MPEG-7 ASP features is mainly composed of six modules: silence removal, feature extraction module using normalized audio spectrum envelope (NASE), speaker change detection, segment-level-clustering module, speaker model updating using MPEG-7 sound recognition and the HMM re-segmentation module. Figure 1 depicts the algorithm flow chart.

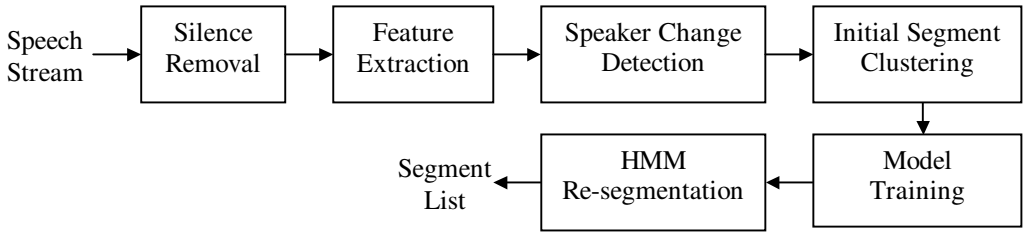


Figure 1: Segmentation procedures

The task of the system is to segment the audio recording, and to assign segments to clusters, where each cluster represents one speaker. First, a silence detection module detects silence segments in the input audio. The silence detection uses a simple energy-based algorithm. The detected silence segments are removed. The speech stream is divided into sub-segments by 3-second window with 2.5-second overlapping. That is, the temporal resolution of the segmentation is 0.5 second. The feature converts the speech conversation into some parameterized representation. The speaker change detection step is performed using a simple distance measure between segments and splits the conversation into smaller segments that are assumed to contain only one speaker. At this stage, the cluster number is kept larger than the actual speaker number, but every cluster gets enough observations to initialize a cluster model. The initial segment-clustering step merges all the segments until the desired N clusters remain. The speaker models are estimated from each cluster. Finally, the re-segmentation module performs a more refined segmentation using these speaker models based on the segments of clusters.

2.1 Feature Extraction Using Normalized Audio Spectrum Envelope

For feature extraction the normalized audio spectrum envelope (ASE) features, a member of the basic spectral descriptors class of the MPEG-7 standard, are used. Figure 2 shows the NASE feature extraction scheme.

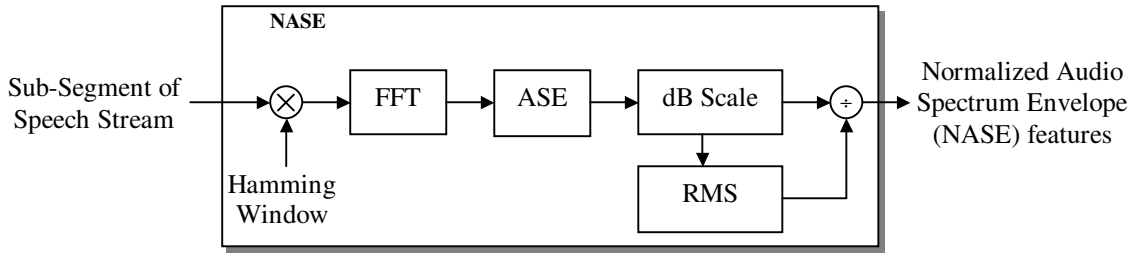


Figure 2: Feature extraction using normalized audio spectrum envelope (NASE)

The speech signal is divided into overlapping frames with a 50 % overlapping shift. Each analysis frame is multiplied by a Hamming window and transformed into the frequency domain using a fast Fourier transformation.

To extract reduced-rank spectral features, the spectral coefficients are grouped in logarithmic sub-bands. The output of the logarithmic frequency range is the sum of the power spectrum in each logarithmic sub-band. The spectrum according to a logarithmic frequency scale consists of one coefficient representing power between 0 Hz and "low edge", a series of coefficients representing power in logarithmically spaced bands between "low edge" and "high edge", and a coefficient representing power above "high edge". The resulting log-frequency power spectrum is converted to the decibel scale. Each decibel-scale spectral vector is normalized with the RMS (root mean square) energy envelope, thus yielding a normalized log-power version of the ASE called NASE. The full-rank features for each frame consist of both the RMS-norm gain value and the NASE vector.

2.2 Speaker Change Detection

In absence of prior information about speakers, an important step in speaker segmentation is to obtain initial estimates for training speaker models. For this, the next step is to build a metric-based segmentation of the NASE acoustic segments for speaker change detection module. The audio recording is segmented into regions spoken by a single speaker.

In metric-based approaches, two neighbouring sub-segments of the NASE feature vectors are moved over the audio signal. The similarity between the contents of the two sub-segments is computed using a distance function.
The distance is determined by the covariance of two sub-segments called divergence shape distance [15][16], which is defined by

$$D = \frac{1}{2} T\left[\left(C_i - C_j\right)\left(C_j^{-1} - C_i^{-1}\right)\right] \qquad (1)$$

where C is the estimated covariance matrix.
At this step, only NASE divergence distance is used to detect potential speaker change point. Thus, a potential speaker change is found between i-th and (i+1)-th sub-segment, if the following conditions are satisfied:

$$\begin{aligned} &D(i,i+1) > D(i+1,i+2), \\ &D(i,i+1) > D(i-1,i) \\ &D(i,i+1) > T_i \end{aligned} \qquad (2)$$

where $D(i,j)$ is the distance between i-th sub-segment and j-th sub-segment, T_i is a threshold.
The threshold T_i is defined by

$$T_i = a \cdot \frac{1}{N} \sum_{n=0}^{N} D(i-n-1, i-n) \qquad (3)$$

where N is the number of previous distances used for predicting threshold, and a ($1 \le a \le 1.5$) is a coefficient as amplifier.
If the distance between two speech segments is larger than a threshold, these two segments could be considered as representing by different speakers.

2.3 Initial Segment Clustering

The segments created by the speaker change detection step are used to initialise the clustering algorithm. The goal of speaker clustering is to classify segmented speech into clusters such that each cluster contains speech from one speaker. Also speech from the same speaker is classified into the same cluster. For this, the next step consists of two components: building a distance measure of the speaker change segments; performing hierarchical clustering on the distance matrix with the prior assumption that consecutive segments should be more likely to come from the same speaker.

The clustering is a simple bottom-up algorithm. Initially, every segment is regarded as a cluster. The Generalized Likelihood Ratio (GLR) distance [17] is calculated:

$$R = \frac{L(z;\mu;\Sigma)}{L(x;\mu_1;\Sigma_1)L(y;\mu_2;\Sigma_2)} \qquad (4)$$

where $L(x;\mu_1;\Sigma_1)$ denotes the likelihood of the *x* sequence and $L(y;\mu_2;\Sigma_2)$ denote the likelihood of the *y*, L_1, of the two segments being generated by different speakers is given by, $L(x;\mu_1;\Sigma_1)L(y;\mu_2;\Sigma_2)$. Furthermore, the likelihood of the segments being generated by the same speaker is given by $L_0 = L(z;\mu;\Sigma)$.

For clustering, we use a hierarchical agglomerative method that computes the distance between every pair of clusters and merges two clusters with the minimum distance at every step. The clustering procedure takes that distance matrix as input and continues to aggregate clusters together until there is just one large cluster. The output from the procedure is a tree of clusters. A bottom up clustering of the frames using the distance function is performed.
At the end of the hierarchical classification algorithm a dendrogram is built in which each node corresponds to a cluster. The cutting of the dendrogram produces a partition composed of all the utterances.
Several techniques exist in the literature [18][19] for selecting a partition. These techniques consist in cutting the dendrogram at a given height or in pruning the dendrogram by selecting clusters at different heights.
In our experiments, we selected the partition by pruning the dendrogram. The method that we call is d^* described in [20]. It is based on the estimated purity nearest (neighbor purity estimator) $\hat{p}_i$ of the cluster *i*. At each stage, the best *score* is selected by

$$score_i = \hat{p}_i - \frac{Q}{n_i}, \qquad (5)$$

where n_i is the number of utterances in candidate cluster *i*, the variable *Q* is a system design parameter that controls the degree to which fewer, larger clusters are favored at the expense of decreased purity.
The $score_i$ is computed for each node *i* corresponding to the cluster *i* composed of n_i elements. The descendants and ancestors of *i* are then removed from the dendrogram.

2.4 Model Training based on MPEG-7 Audio Spectrum Projection

Bottom up (agglomerative) clustering is to form initial clusters from similar segments, which are used to train an initial set of speaker models from all segments of each respective cluster. This initial segment clustering is used to build models with the segments of each cluster.

Given the NASE features of every cluster, the spectral basis is extracted by computing the PCA for dimension reduction and FastICA for information maximization. The resulting spectrum projection is the product of the NASE matrix, the dimension-reduced PCA basis functions and the FastICA transformation matrix. In order to obtain good results with the PCA and ICA algorithms, feature extraction parameters need to be selected with care. The spectrum projection features and RMS-norm gain values are input to the HMM [21] training process as depicted in Figure 3.

In order to train a statistical model on the basis projection features and RMS-norm gain value of each cluster, the MPEG-7 audio classification tools use hidden Markov models (HMMs), which consist of several states. During training, the parameters for each state of a speaker model are estimated by analyzing the feature vectors of a cluster. Each state represents a similar behaving portion of an observable sequence process. At each instant time, the observable symbol in each sequence either stays at the same state or moves to another state depending on a set of state transition probabilities. Different state transitions may be more important for modeling different kinds of data. Thus, HMM topologies are used to describe how the states are connected.

In television broadcasts, temporal structures of video sequences require the use of an ergodic topology, where each state can be reached from any other state and can be revisited after leaving. In our case, an ergodic HMM with 7 states is trained for each cluster.
The HMM parameters are then obtained using a maximum likelihood estimation procedure known as the Baum-Welch algorithm. These class HMMs are then combined into a speaker recognition classifier, which is itself a hidden Markov model.

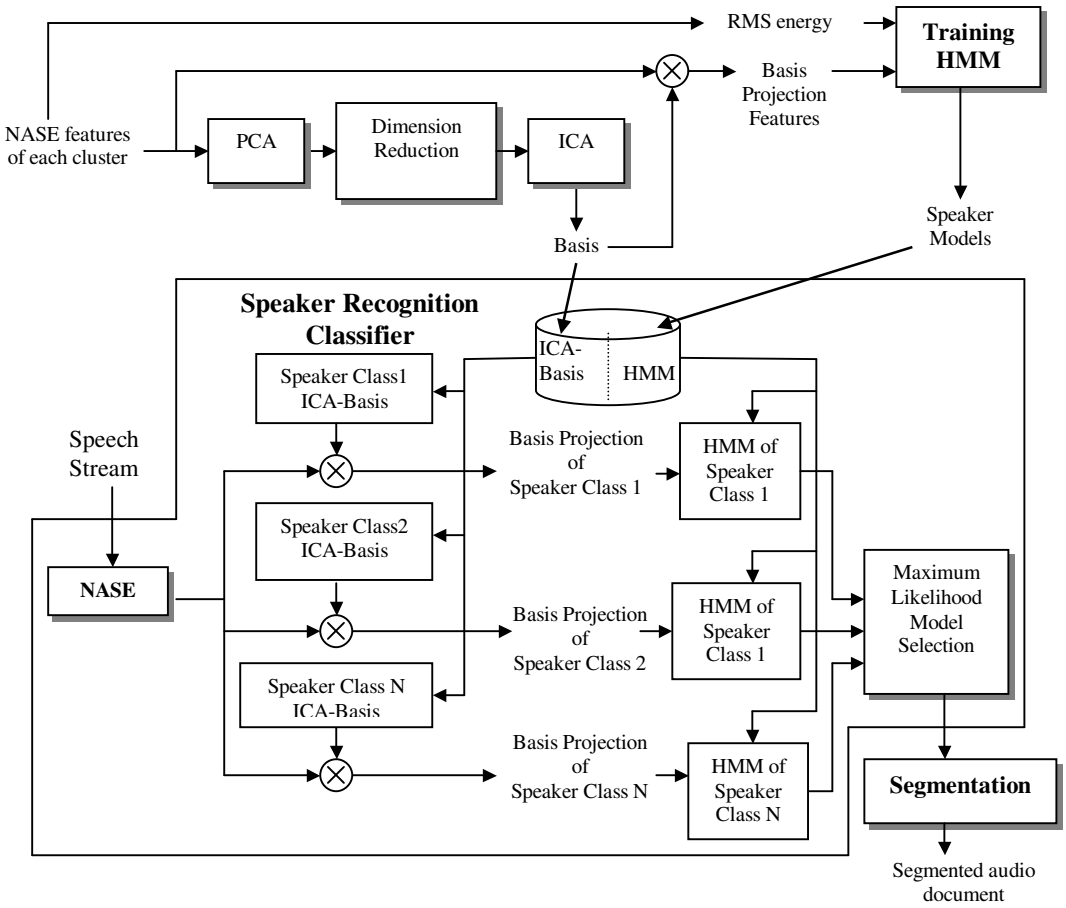


Figure 3: Block diagram of segmentation using a speaker recognition classifier

2.5 HMM Re-segmentation

Re-segmentation is achieved by using the Viterbi algorithm to determine the maximum likelihood state sequence through the speaker recognition classifier, given an observed sequence of feature vectors, and noting those times at which the state sequence passes between states associated with different classes. The trained speaker models are then used to resegment the speech stream. The re-segmentation can be performed in a single pass in real time, and if the models are well trained, this segmentation may be adequate.

For the re-segmentation, an audio track of a television panel discussion is used as input, but other kinds of audio input could be segmented as well. The track is cut into 3-second sub-segments with 2.5-second overlapping. That is, the "hop size" is 0.5 seconds. Overlapping increases the input data to be classified by 50% but yields more robust speaker segmentation results due to the filtering technique described below. We assume that there is no speaker change within each sub-segment. Therefore, speaker detection can be performed at the sub-segment level. Given a 3 second long sub-segment as input, the NASE features are extracted and projected against each sound model's set of basis functions. The Viterbi algorithm is applied to align each projection on its corresponding sound class HMM. The Viterbi algorithm finds the maximum likelihood sequence of states through the recognition classifier and returns the most likely classification label for the sub-segment. Invalid input, such as heated discussions with multiple people speaking at the same time, cause sub-segments to sometimes be classified incorrectly when there are no appropriate models for the input. As a result, the sub-segment labels need to be smoothed out. To this end, we use a low-pass filter to try to enable more robust segmentation by correcting errors. The filter waits for A adjacent sub-segments of the same label before declaring the beginning of a segment. Errors can be tolerated within a segment, but once B adjacent classifications of any other models are found, the segment is ended. For our data, the optimum values were $A = 1.5$ and $B = 1.5$. In a real-time system, this would imply at least a 4 second latency before detecting a new segment. The times between the segments are thus designated as the speaker change points and the segments themselves annotated with the appropriate labels.

3. EXPERIMENTAL PROCEDURE AND RESULTS

3.1 Database

The TV broadcast news are received from a television satellite and stored as MPEG-compressed files. The audio data is compressed with MPEG audio layer 2 compression at a data rate of 192 kbit/s and a sampling rate of 44.1 kHz. The recorded audio signal is sampled down to a 22.05 kHz/16 bit PCM format which was used throughout all our experiments. The set contains 5 test files, every file is about 5 minutes long containing speech of between 7 and 15 speakers. The video part of the signal was not used for the experiments described in this paper.

To evaluate the performance of the different segmenting approaches, we used two audio tracks from television talk-show programs for our purpose. "Talk Show 1" is approximately 15 minutes long and contains only four speakers. "Talk Show 2", which is 60 minutes long, is much more challenging. It contained 7 main speakers (5 male and 2 female), and an applause as the studio audience often responded to comments with applause. The speakers themselves are mostly German politicians arguing about tax reforms, so they interrupt each other frequently.

3.2 Feature Extraction

The input audio signal was divided into sub-segments of length 3 seconds and silence segments are removed. For each non-silence 3s segment, each segment is divided into overlapping frames of duration 30ms with 15ms overlapping for consecutive frames. Each frame was windowed using a Hamming window function. To extract Audio Spectrum Envelope (ASE) features, the observed audio signal was analyzed using a 512-point FFT. The power spectral coefficients were grouped in logarithmic sub-bands spaced in non-overlapping 7-octave bands spanning between low boundary (62.5 Hz) and high boundary (8 kHz). The resulting 30-dimensional ASE was converted to the decibel scale. Each decibel-scale spectral vector was normalized with the RMS (root mean square) energy envelope, thus yielding a normalized log-power version of the ASE called NASE. For each audio class, the spectral basis was extracted by computing the PCA for dimension reduction and FastICA for information maximization.

To select suitable ICA basis parameters, we measured the classification rate of the 3 second long sub-segments from "Talk Show 2". Two minutes from each speaker's opening talk were used to train the speaker models and the last 40 minutes of the program were used for testing. In this case, we assumed that the ergodic topology with 7 states would suffice to measure the quality of the extracted data. The parameter with the most drastic impact turned out to be the horizontal dimension E of the PCA matrix. If E was too small, the PCA matrix reduced the data too much, and the hidden Markov models did not receive enough information. However, if E was too large, then the extra information extracted was not be very important and was better ignored. The recognition rate of the sub-segments versus E from the ICA method for "Talk Show 2" is depicted in Figure 4.

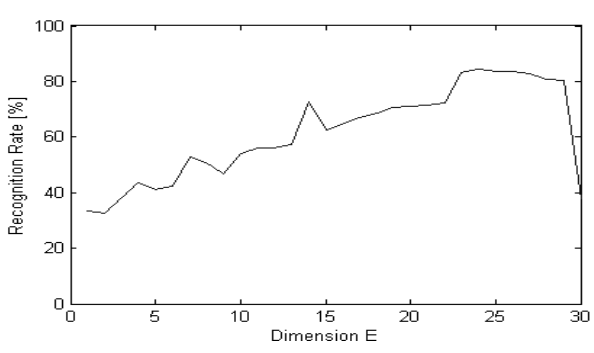


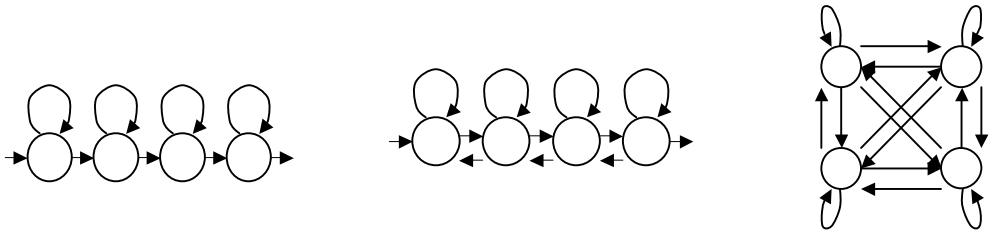
Figure 4: Comparison of recognition rates for different values of E of ICA

As can be seen above, the best value for E was 24 yielding a recognition rate of 84.3%. The NASE was then projected onto the first 24 PCA/ICA basis vectors of every class. The final output feature consisted of 24 basis projection features plus RMS energy. For the classification with these features, we tested the recognition rate for the different HMM topologies as described in section 3.3.

We compare the MPEG-7 ASP features with MFCC which have been widely used in speech recognition and audio classification. MFCCs are based on a short-term spectrum, where Fourier basis audio signals are decomposed into a superposition of a finite number of sinusoids. The power spectrum bins are grouped and smoothed according to the perceptually motivated Mel-frequency scaling. Then the spectrum is segmented into 23-critical bands by means of a filter bank that typically consists of overlapping triangular filters. Finally, a discrete cosine transform applied to the logarithm of the filter bank outputs results in vectors of decorrelated MFCC features.

3.3 HMM Topology

We investigate the effects of using different HMM topologies and differing numbers of states on the speaker recognition rates in the segmentation. Figure 5 shows a few common HMM topologies.



(a) left-right HMM (b) forward and backward HMM (c) ergodic HMM

Figure 5: Illustration of three HMM topologies with four emitting states

Figure 5 (a) shows a 4-state left-right HMM that is suitable for speech recognition. It is simple and works well for classifying short audio files. However, the addition of backwards transitions makes the models much more flexible and works better for classifying longer audio files that may include repetitions. The “forward and backward” HMM from Figure 5 (b) should thus be flexible enough to provide good results. However, the case of a long panel discussion requires to use ergodic HMM such as shown in Figure 5 (c). The transition probabilities themselves depend on the training data of each class and are learned using the Baum-Welch re-estimation training algorithm [20]. The procedure starts with random initial values for all of the model parameters and optimizes them by iterative re-estimation.

Table 1 depicts the classification results for different HMM topologies given the features with E = 24. The number of states includes two non-emitting states, so 7 states implies that only 5 non-emitting states were used.
From the Table 1, we know that the HMM classifier yields the best performance for our task when the number of states is 7 and topology is ergodic. The corresponding classification accuracy is 84.3%. 3 iterations were used to train the HMMs.

HMM topology	Number of states				
	4	5	6	7	8
Left-right HMM	77.3 %	75.9 %	78.1 %	78.8 %	77.5 %
Forward and backward HMM	61.8 %	78.1 %	73 %	76.7 %	75.9 %
Ergodic HMM	58.6 %	75.5 %	80.1 %	84.3 %	81.9 %

Table 1: Sub-segment recognition rate for different three HMMs.

3.4 Segmentation Results

The evaluation for the performance of the error correction using the low-frequency filter is composed of three measures: recall, precision, and F. We define recall (*RCL*) and precision (*PRC*) as follows:

$$RCL = \frac{ncf}{tn}, \quad (6)$$

$$PRC = \frac{ncf}{nhb}, \quad (7)$$

where *ncf* is the number of correctly found boundaries, *tn* is the total number of boundaries, and *nhb* is the number of hypothesized boundaries, meaning the total number of boundaries found by the segmentation module.

RCL errors occur if a true segment boundary has not been spotted by the segmentation (deletion). *PRC* errors occur if a found segment boundary does not correspond to a segment boundary in the reference (false alarm, or segment insertion). The F-measure *F* is a combination of the recall rate of correct boundaries and the precision rate of the detected boundaries. When *RCL* and *PRC* errors are weighted as being equally detractive to the quality of segmentation, *F* is defined as

$$F = \frac{2 \cdot PRC \cdot RCL}{PRC + RCL} \quad (8)$$

Like *RCL* and *PRC*, *F* is bounded between 0 and 1, where *F*=1 is a perfect segmentation result and *F*=0 implies segmentation to be completely wrong.

System	M	FD	Feature Extraction	Recognition Rate (%)	F (%)
Metric-based	News	30	NASE	78.5	75.3
		13	MFCC	82.3	79.9
		24	MFCC	89.7	88.1
	Talks 1	30	NASE	79.5	79.5
		13	MFCC	85.4	87.5
		24	MFCC	93.3	94.6
	Talks 2	30	NASE	66.3	49.9
		13	MFCC	81.6	65.5
		24	MFCC	87.5	73.7
Metric-based + HMM (hybrid)	News	13	MPEG-7 ASP	83.2	88.9
			MFCC	87.1	92.1
		24	MPEG-7 ASP	88.8	93.3
			MFCC	94.3	95.7
	Talks 1	13	MPEG-7 ASP	86.2	88.5
			MFCC	90.5	93.5
		24	MPEG-7 ASP	91.5	94.7
			MFCC	96.8	98.1
	Talks 2	13	MPEG-7 ASP	72.1	56.1
			MFCC	87.2	69.7
		24	MPEG-7 ASP	88.9	75.2
			MFCC	93.2	82.7

Table 2: Performance of the segmentation. M: TV materials, FD: feature dimension

We evaluated the effect of hybrid approach versus metric-based segmentation. Table 2 shows results for two systems. In the first, the segmentation result is the output of a metric-based initial segment clustering module. In the second system (marked as metric-based + HMM (hybrid)), the result comes from the output of the hybrid approach using the HMM re-segmentation.
The results show that the hybrid approach significantly outperforms direct metric-based segmentation, given the suitable initialization of speaker models.

The segmentation results for Talks 1 was quite good because there were only four speakers, and they rarely interrupted each other. The algorithm runs fast enough so that it was implemented on a real-time system. On the other hand, the results of the segmentation for Talks 2 was not as good, but still impressive in view of the numerous interruptions.

The recognition accuracy and F-measure of the MFCC features are better than NASE and MPEG-7 ASP features in the case of both 13 und 23 feature dimensions for "Talk Show 1". For "Talk Show 2" the MFCC features show a remarkable improvement over the MPEG-7 ASP features. Recall that the recognition system identifies speakers as part of the segmentation task. Overall MFCC achieves best recognition rate and F-measure rate.

4. CONCLUSIONS

Different segmenting strategies for speech data have been compared on a common broadcast news data test set and panel discussion television program. A hybrid algorithm, which combines metric-based and model-based segmentation, is shown to outperform the distance metric-segmentation. A distance based segmentation is only used to create an initial set of speaker models. Starting with these, model-based segmentation is performed.
Additionally, we performed experiments with different feature dimensions of the different feature extraction methods. Our results show that the MFCC features yield better performance compared to MPEG-7 ASP in the audio segmentation. In the case of MFCC, the process of segmentation is simple and fast because there are no bases used. On the other hand, the extraction of the MPEG-7 ASP is time and memory consuming compared to MFCC.

5. REFERENCES

[1] M. A. Siegler, U. Jain, B. Raj, R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," Proceedings of Speech Recognition Workshop, 1997.
[2] S. S. Chen, P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," Proceedings of Speech Recognition Workshop, 1998.
[3] P. Delacourt, C. j. Welekens, "DISTBIC: A speaker-based segmentation for audio data indexing," Speech Communication 32, pp. 111-126, 2000.
[4] L. Wilcox, F. chen, d. Kimber, V. Balasubramanian, "Segmentation of speech using speaker Identification," Proceedings of ICASSP, 1994.
[5] J. L. Gauvain, L. Lamel, G. Adda, "Partitioning and transcription of broadcast news aata," Proceedings of ICSLP, 1998.
[6] K. Sommez, L. Heck, M. Weintraub, "Speaker tracking and detection with multiple speakers," Proceedings of EUROSPEECH, 1999.
[7] T. Kemp, M. Schmidt, M. Westphal, and A. Waibel, "Strategies for automatic segmentation of audio data," Proceedings of ICASSP, 2000.
[8] P. Yu, F. Seide, C. Ma, and E. Chang "An improved model-based speaker segmentation system," Proceedings of EUROSPEECH, 2003.
[9] B. S. Manjunath, P. Salembier, and T. Sikora, "Introduction to MPEG-7," Wiley, April, 2002.
[10] M. Casey, "MPEG-7 sound-recognition tools," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11. No.6, June 2001.
[11] ISO/IEC JTC 1/SC 29, "Information Technology Multimedia Content Description Interface-Part 4 : Audio," June, 2001.
[12] H.-G. Kim, E. Berdahl, N. Moreau, T. Sikora, "Speaker recognition using MPEG-7 descriptors," Proceedings of EUROSPEECH, September 2003.
[13] I. T. Jolliffe, "Principal components analysis," Springer-Verlag, 1986.
[14] A. Hyvarinen, E. Oja, "Independent component annlysis: algorithms and application," Neural Networks 13, pp. 411-430, 2000.

[15] J. P. Cambell, J. R. “Speaker recognition: A Tutorial,” Proceedings of the IEEE, Vol. 85, No. 9, pp.1437-1462, 1997.
[16] L. Lu, H.-J. Zhang, “Speaker change detection and tracking in real-time news broadcasting analysis,” Proceedings of 10th ACM international conference on multimedia, pp.602-610, France, December 1-6, 2002.
[17] H. Gish, M.-H. Siu, R. Rohlicek, “Segration of speaker for speech recognition and speaker identification,” Proceedings of ICASSP, pp. 873-876, 1991.
[18] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish, “Clustering speakers by their voices,” Proceedings of ICASSP, 1998.
[19] B. S. Everitt, “Cluster analysis,” Oxford University Press, New York, third edition, 1993.
[20] D. A. Reynolds, E. Singer, B. A. Carlson, J.J. McLaughlin, G.C. O’Leary, and M.A. Zissman, “Blind clustering of speech utterances based on speaker and language charchteristics,” Proceedings of ICASSP, 1998.
[21] L. Rabiner, B.-H. Huang, “Fundamentals of speech recognition,” Prentice Hall, N.J., 1993.