Unsupervised Speaker Segmentation using Autoassociative Neural Network

S. Jothilakshmi S. Palanivel V. Ramalingam

Department of Computer Science and Engineering
Annamalai University
Annamalainagar- India.

ABSTRACT

In this paper we propose an unsupervised approach to speaker segmentation using autoassociative neural network (AANN). Speaker segmentation aims at finding speaker change points in a speech signal which is an important preprocessing step to audio indexing, spoken document retrieval and multi speaker diarization. The method extracts the speaker specific information from the Mel frequency cepstral coefficients (MFCC). The speaker change points are detected using the distribution capturing ability of the AANN model. Experiments are carried out on different audio databases, and the method is capable of detecting speaker changes with short duration of speech in an unsupervised manner.

Categories and Subject Descriptors

Signal processing [Speech processing]: Neural networks

General Terms

Algorithms, Performance

Keywords

Audio indexing, Speaker segmentation, Mel frequency cepstral coefficients, Autoassociative neural networks.

1. INTRODUCTION

The objective of automatic speaker segmentation is to find the speaker change points in an audio stream. This is done by automatically partitioning an input stream into homogeneous segments and assigning these segments to the corresponding speakers.

The crucial problem in the design of audio databases is information retrieval. In audio, information retrieval is normally performed by indexing the audio databases, associating each audio document with a file describing its structure in term of retrieval keys [29].

In order to perform full indexing, an essential initial step is to determine the segmentation of the database with respect to different signals such as speech, music, noise etc. In many cases such as interviews or dialogues, the segmentation process consists of knowing which speaker is speaking at a given time. The speaker segmentation is also useful in speaker verification [6], speech recognition [22], broadcast news classification [25], phone voice classification [30], automatic transcription [12], [27] and spoken document retrieval [26].

In the literature, various speaker segmentation algorithms have been proposed. These algorithms can be categorized into the following categories: decoder based, model based, metric based and hybrid based segmentation algorithms.

In the decoder based approach, it is assumed that the sentences uttered by different speakers in a conversation are delimited by pauses [23]. As a consequence the segmentation relies on the accuracy of an inter speaker silence detector which usually works by measuring the energy or zero crossing rate of each segment and comparing it to a predefined or adaptively estimated threshold. The main drawback of this approach is no direct connection exists between a detected silence and an actual speaker change.

In the model based approach, a set of models is derived and trained for different speaker classes from a training corpus. It assumes that a speaker change is likely to occur at the time indexes where the model's identification decision changes from one speaker to another. As a result, prior knowledge is a prerequisite to initialize the speaker models. The models can be created by means of hidden Markov models (HMMs) [20], [14], [24], [1], Gaussian mixture models (GMM) [18], [12] or support vector machines (SVM) [21], [3], [16].

Other significant algorithms used in the segmentation of audio data records are the metric based segmentation approaches. In which two adjacent windows are selected from the speech stream, and their dissimilarities are assessed by a distance function of their contents. Then the system locates a changing mark in the point in which the dissimilarity is high. Depending on the application the analysis window may overlap or not. Metric based methods do not require any prior knowledge on the number of speakers, their identities, or signal characteristics. A wide variety of dissimilarity metrics have been proposed in the literature. Conventionally adopted metrics are generalised likelihood ratio (GLR) [5], [17], [13], Kullback-Leibler divergence [9], Bayesian information criterion (BIC) [7], Mahalanobis distance [9] and Bhattacharyya distance [9].

Hybrid based algorithms combine metric and model based techniques [15]. A set of speaker models are created by presegmenting the input audio signal using metric based approaches. Then the model based segmentation is applied to yield a more refined segmentation. In [14], HMMs are combined with BIC. Another hybrid system is introduced in [20] where two systems are combined namely LIA system, which is based on HMMs and the CLIPS system, which performs BIC based speaker segmentation followed by hierarchical clustering.

In this paper we consider the problem of segmenting speech containing two speakers. The problem consists of automatically marking the periods of time in which every speaker is talking. This work formulates a new speaker change detection algorithm, which can detect speaker changes with speech segments of short duration. Moreover this algorithm works without any prior knowledge of the identity of speakers, so it is unsupervised.

After obtaining the speech features for each frame of the given conversation, a block of frames are selected from the first frame. It is assumed that the speaker change occurs at the middle frame of the block. Autoassociative neural network (AANN) model is created to capture the distribution of left half of the block [LHB]. The feature vectors of the right half of the block [RHB] are used for testing the model. If speaker change occurs at the middle frame, (i.e., RHB and LHB will be from different speakers) all the feature vectors from the RHB may not fall into the distribution and the model gives low confidence (probability) score. Likewise, if the middle frame is not the true speaker change point and both LHB and RHB are from the same speaker then the confidence score of RHB will be high. The next possibility is either LHB or RHB may have the speech features from both the speakers. If this is the case, the confidence score of RHB will be in between the above two values. After obtaining the confidence score for this middle frame, the block is shifted by one frame to the right. Then the entire procedure is repeated for this new block and the confidence score is obtained by assuming the middle frame of this new block as speaker change point. Likewise the confidence scores are obtained until RHB reached the last frame of the speech frames. From the confidence scores, the local minima positions are the speaker change points and they are detected using a threshold.

The paper is organized as follows: Section 2 describes the method of extracting speaker specific information from the speech signal. Autoassociative neural network (AANN) model for capturing the distribution of acoustic feature vectors is given in Section 3. The proposed algorithm for speaker change detection is presented in Section 4. In Section 5 the various assessment measures used for speaker segmentation algorithms are discussed. Section 6 presents the experimental results and comparison of the proposed method with the existing methods. Section 7 concludes the paper.

2. FEATURE EXTRACTION FOR SPEAKER SEGMENTATION

Feature extraction is an essential step for speaker segmentation. Mel frequency cepstral coefficients have proven to be one of the most successful feature representations in speech related recognition tasks. The mel-cepstrum exploits auditory principles, as well as the decorrelating property of the cepstrum [10]. The DFT based cepstral coefficients are computed by computing IDFT of the log magnitude short time spectrum of the speech signal. The mel warped cepstrum is obtained by inserting an intermediate step of transforming the frequency scale to place less emphasis on higher frequency before computing IDFT.

In this work first 19 mel frequency cepstral coefficients, other than the zeroth value are used. Cepstral mean subtraction is performed to reduce the channel effects. The selected properties for the speech signals are: sampling rate of 8 kHz and 16 bit monophonic PCM format. We used a frame rate of 125 frames / sec, where each frame is 16 ms in duration with an overlap of 50 percent between adjacent frames.

3. AANN MODEL FOR CAPTURING THE DISTRIBUTION OF ACOUSTIC FEATURE VECTORS

Autoassociative neural network models are feed forward neural networks performing an identity mapping of the input space, and are used to capture the distribution of the input data [28].

A five layer autoassociative neural network model, as shown in Figure 1. is used to capture the distribution of the feature vectors in our study. The second and fourth layers of the network have more units than the input layer. The third layer has fewer units than the first or fifth. The processing units in the first and third hidden layers are nonlinear, and the units in the second compression/hidden layer can be linear or nonlinear.

The structure of the AANN model used in our study is 19*L* 38*N* 5*N* 38*N* 19*L*, where *L* denotes a linear unit and *N* denotes a nonlinear units. The nonlinear output function for each unit is *tanh(s),* where *s* is the activation value of the unit. The standard back propagation learning algorithm is used to adjust the weights of the network to minimize the mean square error for each feature vector. As the error between the actual and the desired output vectors is minimized, the cluster of points in the input space determines the shape of the hyper surface obtained by the projection onto the lower dimensional space. The AANN captures the distribution of the input data depending on the constraints imposed by the structure of the network, just as the number of mixtures and Gaussian functions do in the case of Gaussian mixture model.

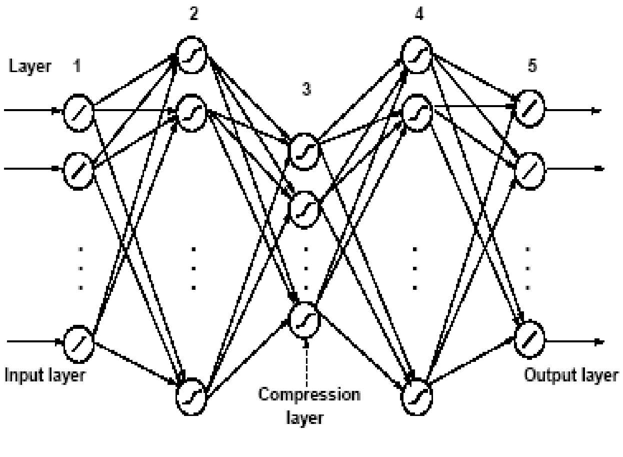


Figure. 1. A five layer AANN model

In order to visualize the distribution capturing ability, one can plot the error for each input data point in the form of some probability surface. The error e_i for the data point *i* in the input space is plotted as $p_i = exp(-e_i/\alpha)$, where α is a constant. Note that

p_i is not strictly a probability density function, but we call the resulting surface as probability surface. The plot of the probability surface shows large amplitude for smaller error e_i indicating better match of the network for that data point.

One can use the probability surface to study the characteristics of the distribution of the input data captured by the network. Ideally, one would like to achieve the best probability surface, best defined in terms of some measure corresponding to a low average error.

4. THE PROPOSED SPEAKER CHANGE DETECTION ALGORITHM

This paper proposes a novel speaker change detection algorithm using AANN. The basic concept of the proposed method is illustrated in Figure 2. We begin with the assumption that there is a speaker change located in the data stream at the center of the analysis window under consideration. If the speech signal of this analysis window comes from different speakers, all the feature vectors in the right half of the window may not fall into the distribution of the feature vectors from the left half window. On the contrary, if the speech signals of this analysis window comes from only one speaker then the feature vectors in the right half of the window falls into the distribution of feature vectors of the left half window.

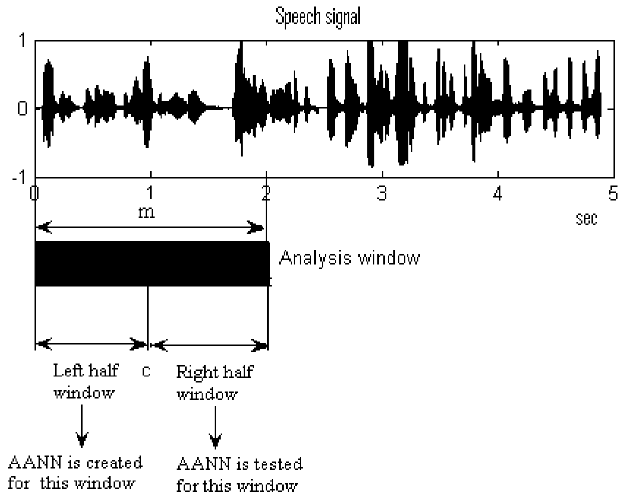
Given the speech feature vectors $S = s_i : i = 1, 2, . . , n$ where i is the frame index and n is the total number of feature vectors in the speech signal. The proposed algorithm for detecting speaker change is given below:

(1) m number of feature vectors ($m \bmod 2)=1$ are considered for k^{th} analysis window W_k and is given by

$$W_k = \{s_j\}, k \le j < m + k \qquad (1)$$

(2) It is assumed that the speaker change occurs at the middle feature vector *(c)* of the analysis window.

$$c = k + \left\lfloor \frac{m}{2} \right\rfloor \qquad (2)$$



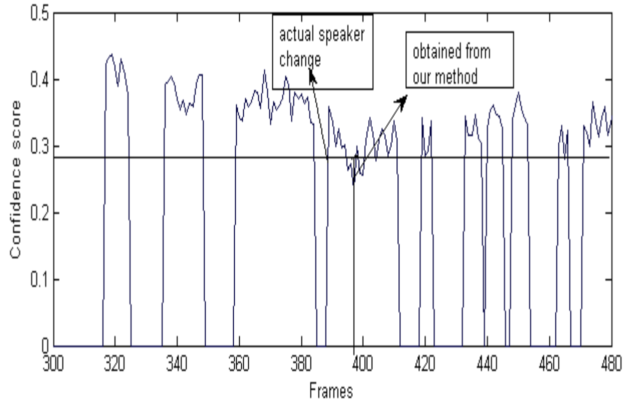


Figure 2. Basic concept of the proposed algorithm

(3) We consider all the feature vectors in the analysis window W_k that are located left of c as left half window (L_k).

$$L_k = \{s_j\}, k \le j < c - 1 \qquad (3)$$

Similarly, all the feature vectors that are located right of c is in right half window (R_k).

$$R_k = \{s_j\}, c + 1 \le j < m + k \qquad (4)$$

(4) AANN is trained using the feature vectors in L_k and the model captures the distribution of this block of vectors. Then feature vectors in R_k are given as input to the AANN model and the output of the model is compared with the input to compute the normalized squared error e_k. The normalized squared error (e_k) for the feature vector y is given by

$$e_k = \frac{\|y - o\|^2}{\|y\|^2} \qquad (5)$$

where o is the output vector given by the model. The error e_k is transformed into a confidence score s using

$$s = \exp(-e_k) \qquad (6)$$

If true speaker change occurs at c, then L_k and R_k will be from different speakers and the confidence score s for this c will be low. Likewise, if c is not the true speaker change point and both L_k and R_k are from the same speaker then the confidence score s will be high. The next possibility is either L_k or R_k may have the speech feature vectors from both the speakers. If this is the case, the confidence score s will be in between the above two values.

(5) The value of k is incremented by one and the steps from 1 to 4 are repeated until $m + k$ reaches n.

It is not possible to obtain the same confidence score for all true speaker change points. The confidence score of speaker change point will be low when compared to the confidence scores of the frames on either side of the speaker change point. So the local minimum of the confidence score is considered instead of global minimum. To avoid the false alarms, the local minima which are less than the threshold value are considered. Hence, after obtaining the confidence score for the entire speech signal the hypothesized speaker change point is validated by using a threshold. The threshold (t) is calculated from the confidence score as

$$t = s_{\min} + as_{min} \qquad (7)$$

Where s_{min} is the global minimum confidence score and a is the adjustable parameter. The proposed method is unsupervised because it can detect the speaker changes without any knowledge of the identity of speakers and there is no need for training speaker models beforehand.

5. ASSESSMENT MEASURES

The performance of speaker segmentation is assessed in terms of two types of error related to speaker change detections namely false alarms and missed detections. A false alarm *(α)* of speaker change detection occurs when a detected speaker change is not a true one. A missed detection *(β)* occurs when a true speaker change cannot be detected. The false alarm rate *(α_r)* and missed detection rate *(β_r)* are defined as [11], [8].

$$\alpha_r = \frac{\text{Number of false alarmed speaker changes}}{\text{Number of detected speaker changes}} \qquad (8)$$

$$\beta_r = \frac{\text{Number of missed detection}}{\text{Number of true speaker changes}} \qquad (9)$$

Two other measures namely precision *(p)* and recall *(r*) can also be used, which are closely related to α_r , β_r [14], [2]. They are defined as

$$p = \frac{\text{Number of correctly found speaker changes}}{\text{Total number of changes found}} \qquad (10)$$

$$r = \frac{\text{Number of correctly found speaker changes}}{\text{Number of actual speaker changes}} \qquad (11)$$

In order to compare the performance of different systems, the f - measure is often used and is given by

$$f = 2\frac{pr}{p + r} \qquad (12)$$

The *f* -measure varies from 0 to 1, with a higher *f* -measure indicating better performance.

In the literature, the false alarms are treated as less cumbersome when compared to missed detections. Over segmentation caused by a high number of false alarms is easier to remedy than under segmentation, caused by high number of miss detection [11], [19], [4]. This means that the segmentation algorithms should yield a lower number of miss detections when compared to the false alarms. To compute these different metrics, it is necessary to take into account that the position of the speaker turns is not exactly defined, due to the presence of inter speaker silences or non speech sounds. Therefore, it is considered that a changing point is correctly located if it belongs to a time interval [t0 − Δt, t0 + Δt] in which t0 is the reference mark and Δt is the tolerance. In our work the tolerance is 0.25 sec.

6. EXPERIMENTAL RESULTS

In order to test the performance of the proposed algorithm, several audio records from TV interviews are considered. A total dataset of 60 conversations is used in our studies. This includes 20 conversations for each male-male, male-female and female-female speaker conversations. The speaker change points are manually marked. The manual segmentation results are used as the reference for evaluation of the proposed speaker segmentation method. A total of 2,782 speaker segments are marked in the 60 conversations. Excluding the silence, the segment duration is mostly between 0.75 to 5 seconds.

The MFCC feature vectors are extracted for all the speech frames as described in Section 2. For each analysis window (W_k), the distribution of the feature vectors is captured using an AANN model as described in Section 3. The feature vectors of L_k are given as input to the AANN and the network is trained for 100 epochs. One epoch of training is a single presentation of all the training vectors to the network. The performance of the AANNs did not change, even if the number of epochs is increased. There is no significant change in the performance of the AANN, even though the number of epochs is increased to 1000. Hence the AANN models are trained for only 100 epochs.

The feature vectors of R_k are given as input to the AANN model and the average confidence score is calculated as described in Section 4. It is repeated for all analysis windows. Figure 3 shows the confidence score obtained for analysis window size of 65, 95, 125 and 140. The number of false alarms and miss detections are significantly low for the 125 frames window size when compared to analysis window size settings of 65, 95 and 140. So, in this work we used the analysis window size of 125 frames. Moreover the window size of 125 frames (1 sec.) is appropriate to detect speaker change for short duration speech segments.

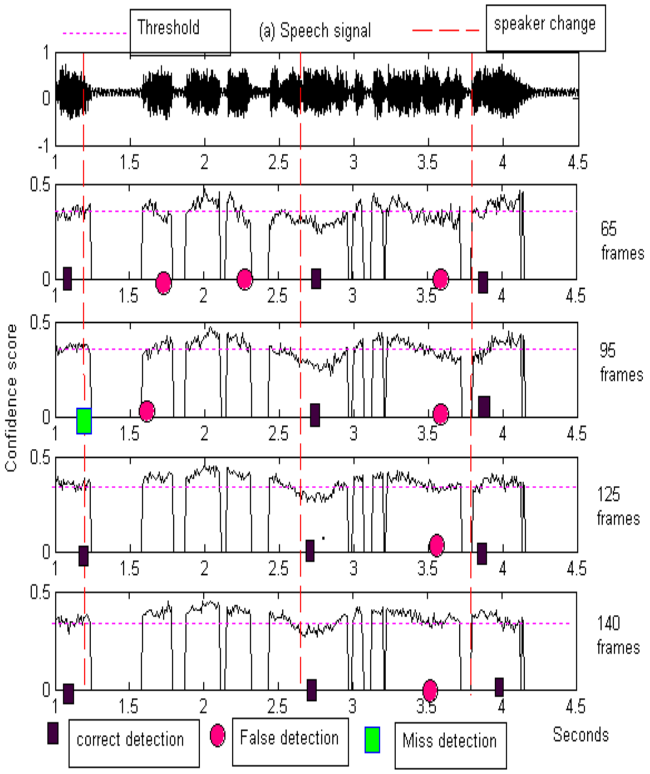


Figure 3. Performance of the algorithm for various analysis window sizes

The performance of the proposed method for speaker change detection for varying adjustable parameter is given in Figure. 4. It shows that nearly 5% miss detections and 5% false alarms are achieved for *a* =0.52. The proposed algorithm achieves 84.3%, 94.7% and 89.2% precision, recall and f -measure respectively for the TV interviews data. The performance of this algorithm is compared with support vector machine (SVM) and Gaussian mixture model (GMM) classifiers.

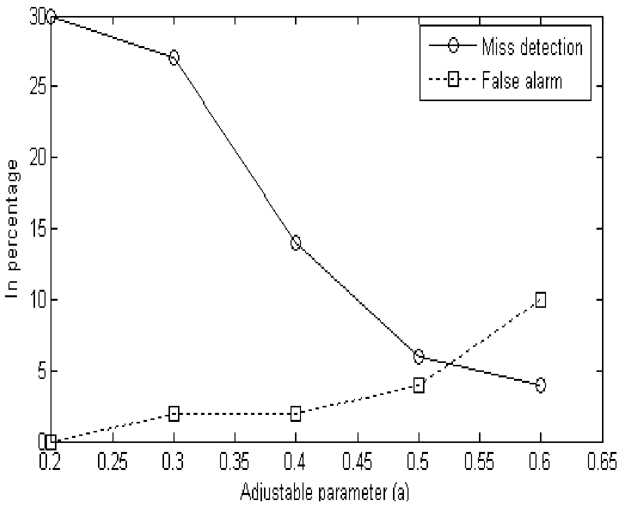


Figure 4. Effect of *a* on the performance of the algorithm

In [16], a window scanning approach was proposed using MFCC and SVM classifier. They claimed that SVM training misclassification rate (STMR) is superior to Gaussian training misclassification rate (GTMR), BIC and the commonly used KL2, Mahalanobis, Bhattacharyya and GLR distances. Hence we compared the performance of our algorithm with SVM classifier and GMM classifier. The feature vector used for AANN classifier is applied for both SVM classifier and GMM classifier. To compute STMR, the linear kernel function with C=1 (where C is the user specified positive parameter for the upper bound of the Lagrange multiplier) and STMR threshold of 0.075 are selected. For GMM classifier, we used four Gaussians with GTMR threshold of 0.04. To compute STMR and GTMR, the feature vectors of the analysis window of size m, the classifiers are trained and tested for m feature vectors. But in our algorithm the training and testing are done with only *m/2* feature vectors. The performance of the method is compared with STMR and GTMR, and the results are given in Table 1.

Table 1. Comparison of proposed algorithm with SVM and GMM classifiers

Classifier	α_r	β_r	Precision	Recall	f-measure
AANN	15.79%	4.68%	83.56%	95.31%	89.05%
SVM	27.01%	25.13%	66.66%	74.87%	70.85%
GMM	38.46%	37.50%	50.12%	62.50%	55.63%

7. CONCLUSION

In this paper we have presented an alternate method for speaker segmentation using MFCC features and autoassociative neural network. The standard methods for speaker change detection require large amounts of data for detecting speaker change points. The proposed algorithm can achieve effective unsupervised speaker segmentation with less speech data collection and it is capable of detecting speaker segments of shorter duration. The algorithm can be applied for real time applications and it does not require any prior knowledge about the speaker identity and their model. Moreover the time taken by the algorithm is less as the AANN model is created by using only one half of the analysis window feature vectors. The performance of this algorithm has been tested using several real conversations from TV interviews and also compared with existing algorithms. The present work was carried out for only two speaker conversations and by using clean speech signals. This work can be used for multispeaker diarization.

8. REFERENCES

[1] J. Ajmera, I. McCowan, H. Bourland. 2003. Speech/music segmentation using entropy and dynamism features in a HMM classification framework. Speech comm. 40, 3, 351–363.

[2] J. Ajmera, I. McCowan, H. Bourland. 2004. Robust speaker change detection. IEEE Signal Process. Lett. 11, 8, 649–651.

[3] J. A. Arias, J. Pinquier, R. Ande-Obrecht. 2005. Evaluation of classification techniques for audio indexing. In Proceedings of the 13th European conf. Sinal processing.

[4] C. Barras, X. Zhu, S. Meignier, J. L. Gauvain. 2006. Multistage speaker diarization of broadcast news. IEEE Trans. Audio, Speech, Lang. Process. 14, 5, 1505–1512.

[5] J. F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, C. Wellekens. 2000. A speaker tracking system based on speaker turn detection for NIST evaluation. In Proceedings of the IEEE International conference on Acoust., Speech, Signal Process.(ICASSP 00). 1177–1180.

[6] K. Chen. 2003. Towards better making a decision in speaker verification. Pattern Recognition. 36, 329–346.

[7] S. S. Chen, P. S. Gopalakrishnan. 1998. Speaker, environment and channel change detection and clustering via the Bayesian information criterion. In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop.

[8] S. Cheng, H. Wang. 2004. Metric SEQDAC: a hybrid approach for audio segmentation. In Proceedings of the 8th International conference on spoken language processing. 1617–1620.

[9] L. Couvreur, J. M. Boite. 1999. Speaker tracking in broadcast audio material in the frame work of the THISL project. In Proceedings of the Workshop on accessing information in spoken audio(ESCA-ETRW99). 84–89.

[10] S. B. Davis, P. Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoust., Speech, Signal Processing. 28, 357–366.

[11] P. Delacourt, C. J. Wellekens. 2000. DISTBIC: a speaker based segmentation for audio data indexing. Speech comm. 32, 111–126.

[12] J. Gauvain, L. Lamel, G. Adda. 2002. The LIMSI broadcast news transcription system. Speech comm. 37, 89–108.

[13] T. Kemp, M. Schmidt, M. Westphal, A. Waibel. 2000. Acoustics, strategies for automatic segmentation of audio data, In Proceedings of the IEEE International conference on Acoust., Speech, Signal Processing.(ICASSP 00). 1423–1426.

[14] H. Kim, D. Elter, T. Sikora. 2005. Hybrid speaker based segmentation system using model level clustering. In Proceedings of the IEEE International conference on Acoust. Speech, Signal Processing (ICASSP 05). 745–748.

[15] M. Kotti, V. Moschou, C. Kotropoulas. 2007. Speaker segmentation and clustering. Signal processing. 88, 1091–1124.

[16] P. Lin, J. Wang, J. Wang, H. Sung. Unsupervised speaker change detection using SVM misclassification rate. IEEE Trans. Computers. 56.

[17] D. Liu, F. Kubala. 1999. Fast speaker change detection for broadcast news transcription and indexing. In Proceedings of the European Conf. Speech comm. and technology (EUROSPEECH '99). 1031–1034.

[18] L. Lu, H. J. Zhang. 2002. Speaker change detection and tracking in real time news broadcasting analysis. In Proceedings of the 10th ACM Int'l conf. Multimedia. 602–610.

[19] L. Lu, H. Zhang. 2005. Unsupervised speaker segmentation and tracking in real time audio content analysis. Multimedia system. 10, 4, 332–343.

[20] S. Meignier, D. Moraru, C. Fredouille, J. F. Bonastre, L. Besacier. 2006. Step by step and integrated approaches in broadcast news speaker diarization. Computer, Speech and Language. 20, 303–330.

[21] S. Mesgarani, S. Shamma, M. Slaney. 2004. Speech discrimination based on multiscale spectro-temporal modulations. In Proceedings of the IEEE Int'l conf. Acoustics, speech, signal processing (ICASSP '04). 601–604.

[22] K. Mori, S. Nakagawa. 2001. Speaker change detection and speaker clustering using VQ distortion for broadcast news speech recognition. In Proceedings of the IEEE International conference on Acoust., Speech, Signal Process.(ICASSP 01). 413–416.

[23] M. Nishida, Y. Ariki. 1997. Speaker indexing for news articles debates and drama in broadcasted TV programs In Proceedings of the Speech Recognition Workshop. 67–72.

[24] B. L. Pellom, J. H. L. Hansen. 1998. Automatic segmentation of speech recorded in unknown noisy channel characteristics. Speech comm. 25 ,1-3, 97–116.

[25] M. Siegler, U. Jain, B. Raj, R. Stern. 1997. Automatic segmentation, classification and clustering of broadcast news audio. In Proceedings of the DARPA Speech Recognition Workshop. 97–99.

[26] M. Viswanathan, H. S. M. Beigi, S. Dharanipragada, A. Tritschler. 1999. Retrieval from spoken documents using content and speaker information. In Proceedings of the Fifth Int'l conf. Document Analysis and Recognition (ICDAR '99). 567–572.

[27] S. Wegmann, P. Zhan, L. Gillick. 1999. Progress in broadcast news transcription at dragon systems. In Proceedings of the IEEE Int'l conf. Acoustics, speech, signal processing (ICASSP '99). 33–36.

[28] B. Yegnanarayana, S. P. Kishore. 2002. AANN: An alternative to GMM for pattern recognition. Neural Networks. 15, 459–469.

[29] T. Zhang, J. Kuo. 2001. Audio content analysis for online audiovisual data segmentation and classification. IEEE Trans. Speech and Audio Processing. 9, 4, 441–457.

[30] X. Zhong, M. Clements, S. Lim. 2003. Acoustic change detection and segment clustering of two way telephone conversation. In Proceedings of the European conf. speech comm. technology (EUROSPEECH '03). 2925–2928.