Light Weight Mobile Device Targeted Speaker Clustering Algorithm

Olli Vuorinen[#], Tommi Lahti[*], Satu-Marja Mäkelä[#], Johannes Peltola[#]

[#] *VTT Technical Research Centre of Finland*
P.O.Box 1100, FIN-90571, Oulu, Finland
email: firstname.surname@vtt.fi

[*] *Nokia Research Center, Tampere , Finland*
P.O.Box 100, FIN-33721, Tampere, Finland
email: firstname.surname@nokia.com

***Abstract*—In this paper we present a novel light weight speaker clustering algorithm based on the Bayesian Information Criterion (BIC). Algorithm utilises BIC profiles, which were earlier used for False Alarm Compensation (FAC) in our Speaker Change Detector (SCD). Proposed speaker segmentation followed by a light weight clustering is targeted to segment and label mobile device recordings directly in the device itself. Thus the main criterion in algorithm design was to maintain high detection accuracy while keeping computational costs in low level. Clustering algorithm gave *F*-score performance of 0.90 for speaker segmentation, which is 29% relative improvement compared to baseline [1] results. Speaker segment labelling performance was 88%, when the number of speakers was undetermined. The experimental results indicate that our unsupervised speaker clustering algorithm is sufficiently effective and efficient for speaker segmentation applications in mobile devices.**

I. Introduction

User generated content is getting more and more created using mobile phones and distributed through the social media services. Managing this content requires high quality metadata, thus more research effort on content analysis techniques such as speaker segmentation and clustering is required to ensure the development of high quality services for organising and distributing camera phone video clips.

The goal of the speaker segmentation and clustering is to find the boundaries for speaker segments and detect which segments are spoken by the same speaker. Typically speech from the same speaker may appear multiple times in an audio stream. In mobile device applications speaker metadata available from speaker clustering is useful for indexing and browsing of audio and video data. Clustering can also be used to accumulate longer speech segments for subsequent processing e.g. speaker adaptation, speaker identification, speaker emotion detection etc.

The usage environment of the mobile user terminals causes new challenges to speaker segmentation as well. The audio and video material may be recorded in any environment. This requires speaker segmentation algorithms to be robust against unforeseeable background noise.

Speaker clustering has been an active research field for many years. The fundamental problems are: how to form the clusters, how to find the correct number of clusters and, which measure should be used for merging and boundary decision. Ideally, the clustering should result in a single cluster for every speaker. Clustering of audio segments is often performed via hierarchical clustering [2]. Hierarchical clustering may be done divisively from top to down or using agglomerative bottom-up techniques. Forming the clusters can be divided into two groups: *model-based* and *non-parametric* cluster representations. The examples of former case can use GMMs, HMMs or HMNets. The later is represented by speaker grouping using e.g. estimated Vocal Tract (VT) parameters [3].

Speaker clustering methods can also be divided to supervised or unsupervised speaker clustering. In unsupervised clustering case prior knowledge of the number of speakers or speaker identities are not available. Traditional speaker clustering requires all the training data for expectation-maximization (EM) algorithm and is not suitable in mobile device application requirements because of its computational complexity and storage requirement. In [4] models are adapted online when the speech data is obtained from speech stream gradually, and the speaker model is established incrementally.

In this paper we present a new unsupervised light weight speaker clustering method, which is using BIC profiles [1] to form cluster representatives. Based on our earlier experiences BIC profiles were promising for false alarm compensation developed for SCD [1]. BIC profiles were especially promising in two speaker cases. In this paper we present methods which can help to handle also more than 2 speakers. It should be noted that the number of speakers in personal video recordings is usually rather modest. In our tests the recordings contained speech from 1, 2, 3 and 4 different speakers. The target of the proposed speaker clustering algorithm is to avoid computationally demanding solutions, while keeping the performance of SCD in high level. Speaker clustering must be also unsupervised and should be robust against the variations in audio data properties.

The rest of the paper is organized as follows. Section 2 describes speaker segmentation and clustering algorithms in details. Section 3 describes the evaluation set up and section 4 concludes this work.

978-1-4244-2295-1/08/$25.00 © 2008 IEEE

II. Speaker Segmentation Algorithm

BIC is one of the most commonly used methods for speaker change detection. Use of BIC in this context was first proposed by Chen & Gopalakrishnan [2]. The BIC is a maximum likelihood criterion penalized by the complexity of model parameters. A one data segment has two hypotheses, it either consist speech of one speaker when there exists a single Gaussian model or it consists speech of two speakers with two multidimensional Gaussian models. The maximum likelihood ratio between the two hypotheses is then formulated as

$$R(i)=\frac{N_x}{2}\log|\Sigma_x| - \frac{N_{x1}}{2}\log|\Sigma_{x1}| - \frac{N_{x2}}{2}\log|\Sigma_{x2}|\,, \qquad (1)$$

where Σ is the corresponding covariance matrix and N is the number of acoustic vectors in the complete sequence. The variations between one speaker (one Gaussian) and two speakers (two different Gaussians) is given by

$$\Delta BIC\,(i) = - R(i) + \lambda P, \qquad (2)$$

where P is the penalty term $P = ½ (p + ½p (p+1)) * \log N_X$ and p is the dimension of the acoustic space and λ is the penalty factor. The negative value of BIC denotes the speaker turn change in the sequence.

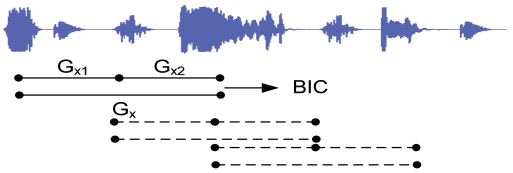

Fig. 1. Example of window sliding for 1st BIC.

BIC is achieved by comparing Gaussian distributions G_{x1} and G_{x2} calculated for two adjacent windows to Gaussian distribution G_x calculated for window including both smaller windows, see Fig. 1.

A. Speaker Segmentation

A block diagram of the speaker segmentation system is shown in Fig. 2. Initial speaker change detection consist of a Voice Activity Detector (VAD), audio feature extraction using Mel Frequency Cepstral Coefficients (MFCC), dissimilarity measurement using BIC calculated using Gaussian distribution model and decision logic, where information of detected silences is integrated with peak detection. After the initial speaker changes are detected they are validated or discarded using BIC based speaker clustering algorithm that is the main focus of this paper. Finally, speaker segmentation metadata is extracted from SCD.

In the first step input frames are classified to speech or silence. In our approach this step is executed by using combined Voice Activity Detector (VAD). Combined VAD utilizes conventional energy based VAD and Long Term Spectral Divergence (LTSD) VAD which is know to be especially noise robust [5]. The VAD that finds more silence frames is used to find silence positions for Silence Detection. Generally speaking LTSD VAD detects more silence in noisy environment. VAD selects speech frames to feature extraction , where 20 MFCCs are computed every 10 ms from a 30 ms analysis window with 20 ms overlap, which is commonly used setup in the literature [6], [7]. Gaussian distribution model is calculated using the MFCC feature vectors. We use diagonal covariance matrix, which is a good compromise between quality and model size.

One speaker test (OST) is done to detect if the recording contains speech only from one speaker. The test is based on the BIC-ratio, which is described below. If OST fails, the proposed Speaker clustering algorithm can still do the merging. More detailed description of the used SCD system is presented in [1], however, without the proposed speaker clustering algorithm.

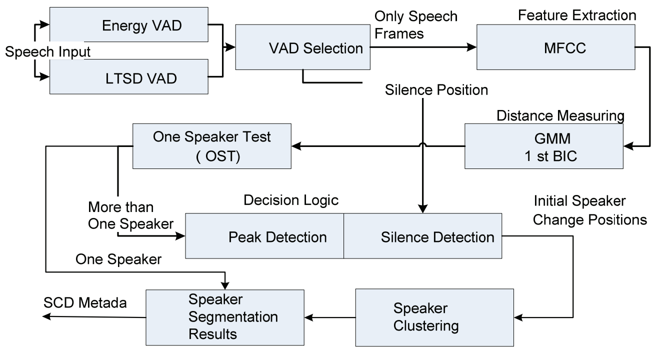

Fig. 2. Block diagram of speaker segmentation system.

B. Speaker Clustering

We had two main targets for developing a new speaker clustering algorithm. First, to develop a real-time speaker clustering algorithm, which is able to cluster segments from speaker change detector and give them a proper speaker label. Secondly, to improve the false alarm compensation presented in [1]. Although the number of false alarms in [1] reduced efficiently, some real change points were also eliminated, decreasing the detection rate.

To improve the performance and robustness of our light weight speaker clustering algorithm, we have adopted following methods:

- BIC Profiles, which represent distances between all detected speaker segments. In speaker clustering BIC profile is used as a cluster representative
- RMS-measure, which evaluates the similarity between speakers
- BIC-ratio, which estimates dynamics of BIC distance time series
- Hierarchical top-down clustering tree

The use of these methods in speaker clustering is discussed more in detail in next subsections.

1) *BIC Profiles*

Typically BIC distance measure is applied to adjacent speech segments as in [8] for detecting or evaluating speaker change points. Problems in this approach include difficulties of setting proper thresholds and dealing with short data segments. Our approach is different and is based on the assumption that, if sequences belong to the same speaker, their BIC distance relations, which we call BIC profiles, to all other segments are mostly similar [1].

BIC distance matrix is calculated between all detected speech segments. Segments are composed based on initial speaker change positions from SCD (Fig. 1).

Each row represents, one segment BIC distances against all the segments, we call it as BIC profile. BIC matrix can be presented as:

$$BIC_{Matrix} = \begin{bmatrix} BIC\,(S_{1,1}) & BIC\,(S_{1,2}) & \cdots & BIC\,(S_{1,j-1}) & BIC\,(S_{1,j}) \\ BIC\,(S_{2,1}) & BIC\,(S_{2,2}) & \cdots & & BIC\,(S_{2,j}) \\ \vdots & & \ddots & & \vdots \\ BIC\,(S_{i-1,1}) & & & & BIC\,(S_{i-1,j}) \\ BIC\,(S_{i,1}) & BIC\,(S_{i,2}) & \cdots & BIC\,(S_{i,j-1}) & BIC\,(S_{i,j}) \end{bmatrix} \tag{3}$$

where BIC(S*i, j*) is a BIC value calculated between speech segments initially labelled as *i* and *j*. Segment indexes *i* and *j* get values from one to the number of segments.

In developed speaker clustering algorithm Gaussian distribution model is used in BIC calculation in a same way as in initial speaker change calculation, see formulas (1) and (2). Only difference is that BIC is calculated between two detected speaker segments, instead of data from two sliding window pairs [1].

To make speaker clustering more robust, BIC matrix in (3) is normalized. This helps to set thresholds for RMS-measure, which is discussed in more detail in the next subsection. In normalization all BIC values are scaled between [0, 1] and the scaled values are then subtracted from one. Hence, in opposite to regular BIC, small values present similarity and bigger values dissimilarity. In Fig. 3 are illustrated BIC profiles, including five speech segments from three different speakers. One speech segment is from speaker1, two segments from speaker2 and two segments are from speaker3. Speaker labels corresponding to segment index in Fig. 3 are: 1, 2, 3, 2 and 3.

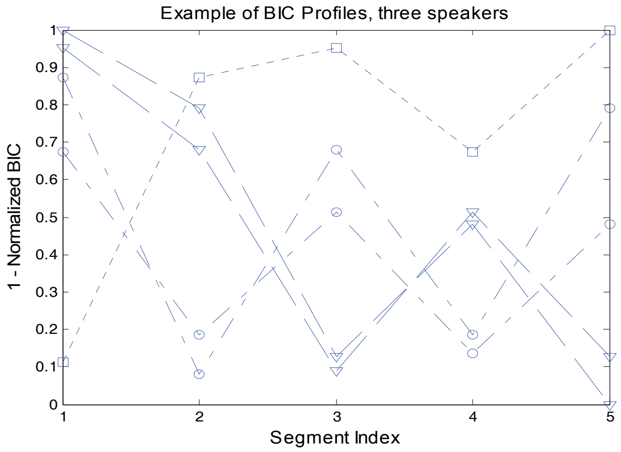

Fig. 3. Example of BIC profiles. (square=speaker1, circle=speaker2 and triangle=speaker3).

The clustering algorithm uses the BIC profile information for creating the clusters. The determination of the cluster is simply done by selecting one BIC profile to represent each speaker. We call this selected BIC profile here as Representative Speaker Cluster Profile (RSC profile). The selection of the RSC profile is explained soon below.

2) *RMS – measure*

To measure the closeness between already selected RSC profile and candidate profile, RMS-measure is used. It is calculated by subtracting candidate profile from RSC profile and calculating variance from the residual (Fig 4).

```
function RmsDiff(X, Y)
{
  MinRms=1;
  Rms=0;
     FOR every vector xi in X DO
     {
       // subtract CandidateProfile from each
       // Representative profile
          Residual = Y - Xi;

       //Calculate variance from residual
          Rms=var(Residual)

       // update minimum Rms
             if (MinRms > Rms)
              {
                 MinRms=Rms;
              }
     }
  // Return minimum distance
  return MinRms;
}
```

Fig. 4. Calculation of RMS-measure between candidate profile and RSC profiles.

New cluster can be created, if the minimum value of RMS-measures between candidate profile and existing clusters is greater than a threshold (experimentally about 0.05).

3) *BIC-ratio*

The clustering algorithm tests if the remaining segments contain speech from one or multiple speakers by using One Speaker Test (OST). For that test, measure named BIC-ratio is used. BIC-ratio measure estimates if the speech segments contain internal variation of one speaker or inter speaker variation. BIC-ratio is calculated by dividing the minimum value of the BIC matrix by the maximum value of the BIC matrix. This estimates the biggest variation between segments in a current situation. The measure is based on the natural assumption that there is less variation in BIC values if the segments do not contain speech from different speakers.

The probability to have only one speaker segments increases, when BIC-ratio gets bigger values. The maximum value is one, which indicates that compared segments are homogenous. The advantage of our approach is that the threshold setting is robust against variations in different data sets.

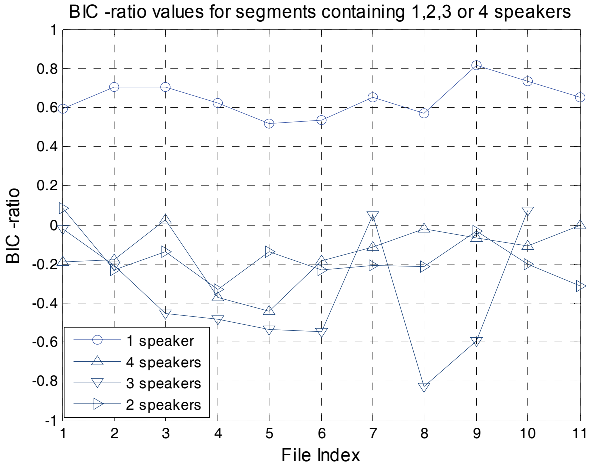

Fig. 5. Example of BIC-ratio values for different audio files containing segments from 1, 2, 3 or 4 speakers

In the upper area of figure 5 is BIC-ratio curve representing one speaker data. The internal variation of speech sets the BIC-ratio of one speaker segments near 0.6. The inter speaker variations of 2, 3 and 4 speaker data drop the BIC-ratio curves significantly lower. One speaker threshold is set experimentally to 0.5, which is used to separate one speaker segments from segments containing speech from several speakers.

4) Clustering process

The progress of the clustering process is presented in pseudo code below. At the start the data is treated as one cluster. In the process the data cluster is split into two sub-clusters if needed and the clustering process is repeated for each sub-cluster. Hence, it should be noted that even though the process for some sub-cluster is terminated, the overall clustering process may be still on going for some other branch(es) of the overall process.

The main purpose of the proposed clustering is to find one RSP profile for each speaker. When all representative profiles are found, segments are labelled with the same label as the closest RSC profile.

Steps of Speaker Clustering (level 0)**:**

```
I.   Initialization:
         Calculate the BIC matrix
         Calculate the BIC profiles
         Calculate the BIC-ratio
II.  If BIC-ratio > threshold
         Stop splitting the cluster
III. Candidate profile selection
         Find two profiles, which have biggest difference.
         Difference can be calculated from BIC-matrix or using
         RMS-measure between profiles.
IV.  Calculate the RMS-measure between the two candidates
     If RMS-measure < threshold
         Stop clustering the cluster
     Else
         Accept both candidate profiles as proper RSC-
         profiles
V.   Create two clusters out of one by clustering all remaining
     BIC-profiles in the original cluster to the closest RCS-
     profile according to the RMS-measure.
VI.  Repeat the process for each resulted sub-cluster in the
     next level of the clustering tree, see Fig. 6.
```

After all clusters (one for each speaker) are found, all segments are labelled with same label as closest cluster. If adjacent speaker segments contain same label the segment boundary is decided as being a false alarm and it is removed.

The structure of clustering tree is illustrated in figure 6. Comparing with basic hierarchical clustering algorithms, the additional part is BIC –ratio test, which tests if the remaining cluster contains only one speaker segments, when clustering is stopped in that branch.

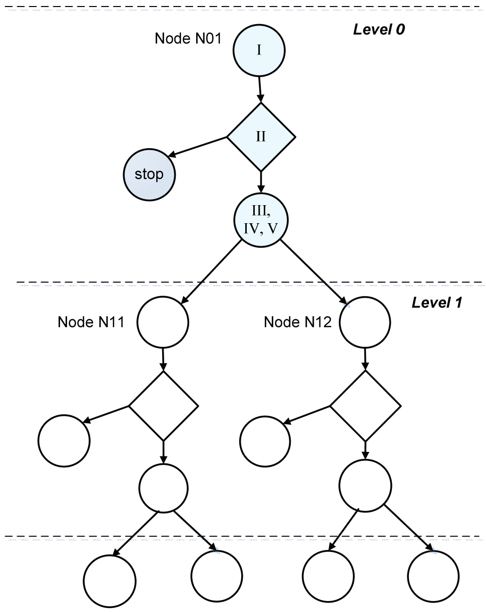

Fig. 6. Clustering tree structure

The advantage of splitting the clusters based on the biggest BIC difference is rather clear. In general, at least at the beginning of the clustering process, the biggest BIC difference is capable in capturing the clearest differences between speakers, for example to separate male speaker from female speaker. Then in sub clusters it is easier to separate for example male/male and female/female segments having smaller differences in their speech characteristic.

It should be noted that implementation variants are always possible and we have tested several off them. For example, in step III it is possible to select the maximum difference between profiles using RMS-measure between two BIC profiles, or finding maximum BIC-distance directly from BIC-matrix.

III. Experimentations

A. Database

Test database used in simulations is recorded with high quality in 44 - 48 kHz with 16 bits and down sampled to 16 kHz. Dataset contains 99 separate recordings, which contain 2, 3 or 4 speakers and each category has 33 recordings. Amount of different speakers in the dataset is in total 21 representing both genders. Average duration of the speaker segment is 11.6 seconds and the number of speaker change points in total is 630.

B. Evaluation methods

For comparing target and hypothesized changes, we are using the precision *PRC*, recall *RCL* and the *F*-score measures. The precision and recall measures are defined as

$$PRC = \frac{\text{numberof correctlyfound changes}}{\text{total numberof changesfound}} \quad (4a)$$

and

$$RCL = \frac{\text{numberof correctlyfound changes}}{\text{total numberof correct changes}}. \quad (4b)$$

The evaluation of the segmentation quality is made in terms of *F*-score, a combined measure of *PRC* and *RCL* of change detection. In the literature *F*-score is often defined as

$$F - score = \frac{2.0 * \mathrm{PRC} * \mathrm{RCL}}{\mathrm{PRC} + \mathrm{RCL}}. \quad (5)$$

The *F*-measure values vary from 0 to 1, with a higher *F*-measure indicating better performance.

We use relative improvement (*RI*) measure to compare baseline results to proposed speaker segmentation results.

$$RI = 100 * \frac{F_X - F_B}{(1 - F_B)}, \quad (6)$$

where F_X denotes the *F*-score value of the proposed speaker segmentation method to be compared against F_B, the *F*-score value of the baseline SCD [1].

C. Simulation Results

It is assumed that a speaker change point is true if the bias from the hand-labelled break point is less than 1 s. In Fig. 7 is presented the histogram, which shows the bias from the hand-labelled change point. It can be seen that most segmentation decisions are located closer that 0.2 s from the true change point.

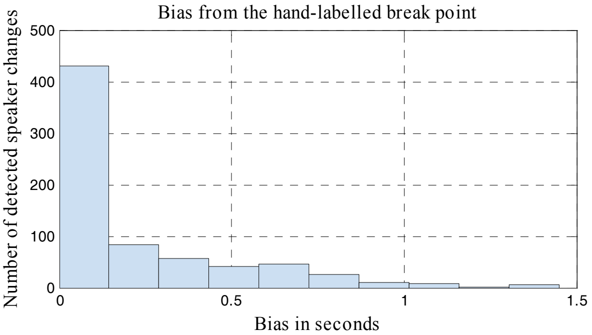

Fig. 7. Bias values from the hand-labelled change points

Speaker segmentation results are presented in Table I. The results are compared to baseline, which is our earlier implementation of the speaker segmentation that uses BIC profiles based false alarm compensation [1].

TABLE I
Speaker Segmentation Performance

Method	*F*-score	Recall	Precision
Baseline	0.86	0.92	0.82
Proposed	0.90	0.92	0.89

Results show that proposed Speaker Clustering algorithm merges efficiently segments from the same speaker, while keeping the number of correctly detected changes in a high level. The relative improvement (6) of *F*-score in segmentation results against the baseline is 29.1%.

In Table II, are presented speaker segmentation results, for each number of speakers in used test set. The number of speakers is unknown. It can be noted that *F*-score stays nearly at the same level.

TABLE II
Speaker Segmentation Performance, the Number of Speakers is Unknown

# Speakers	*F*-score	Recall	Precision
2	0.92	0.91	0.92
3	0.89	0.90	0.90
4	0.88	0.95	0.84
All	0.90	0.92	0.89

In Table III, are presented speaker segmentation results, when the number of speakers is supervised. Often in practice it very difficult to get the information on the number of speaker in before hand. In some cases user might be willing to offer the information. Fortunately, based on the results, it can be seen that in both cases the performance is about the same.

TABLE III
Speaker Segmentation Performance, The Number of Speaker is Supervised

# Speakers	*F*-score	Recall	Precision
2	0.92	0.93	0.91
3	0.89	0.90	0.89
4	0.89	0.91	0.87
All	0.90	0.92	0.89

Speaker clustering results are evaluated by comparing the manual annotated speaker label with speaker label from speaker clustering algorithm. We calculated segmentation results using script, which allows the reference and hypothesis speaker segments to have different labels, as mentioned in [9]. This may occur e.g. in situation when labelling detects falsely an extra speaker between speakers one and two. Speaker two becomes then speaker three, and all other segments from this speaker should be labelled as three even if the ground truth uses label two.

In table IV are presented the results for speaker clustering. Test was executed in two different ways. In a supervised manner the number of speakers was forced to be correct. In the unsupervised case, the number of speakers was unknown. Max of speakers was then set to greater number than the real maximum number of speakers.

TABLE IV
SPEAKER CLUSTERING RESULTS IN TERMS OF CORRECT SPEAKER LABEL PERCENTAGES

# Speakers	**Unsupervised**	**Supervised**
2	94.73	97.61
3	86.22	89.29
4	84.61	87.90
All	88.52	91.60

In the case of unsupervised speaker clustering, the mean for correctly labelled speakers is 88.52%. Correspondingly, mean segmentation error is 11.48% for all test set.

IV. CONCLUSIONS

Proposed speaker clustering method gave clear improvement to false alarm compensation comparing with SCD [1] results for test set used in this work. The relative improvement against the baseline result is 29.1 %. In addition the clustering provides estimation about the amount of speakers in the audio track and is also capable of labelling segments that contain same speakers.

Mean value of the correct speaker label percentage is 88.52 %, when the number of speakers is not known in before hand and in the supervised case, when the number of speakers is fixed, corresponding result is 91.6 %. The mean difference between them is only 3.77%, which indicates the good performance for the unsupervised speaker clustering, given that the number of speakers is not determined forehand.

The proposed speaker clustering method, utilizing BIC profiles, fulfils the given requirements, since it enhances the false alarm compensation performance and can also estimate the amount of speakers and their labels with a relatively good performance even for shorter segments of data. The computational requirements remains low since the algorithm does not need e.g. costly estimation of GMM or HMM parameters. Indicative tests show that proposed light weight speaker clustering algorithm increase computational costs only about 1-3% compared to SCD [1].

Although developed lightweight clustering was applied first to the clustering of speakers, the future interest is to test how it can be applied generally to the detection of acoustical changes and clustering them to homogenous regions according to environmental condition.

ACKNOWLEDGEMENTS

This work is partly funded by Nokia Research Center, Finland and partly by National Technology Agency of Finland (TEKES) in ITEA project Cantata.

REFERENCES

[1] O. Vuorinen, J. Peltola, S.-M. Mäkelä, “Unsupervised Speaker Change Detection for Mobile Device Recorded Speech”, in Proc. IEEE ICASSP’07, pp. 757-760, Honolulu, USA 2007.

[2] Scott Shaobing Chen, P.S. Gopalakrishnan, “Speaker, environment and channel change detection and clustering via the bayesian information criterion”, 1998 DARPA Broadcast News Transcription & Understanding Workshop.

[3] M. Naito, L. Deng and Y. Sagisaka, “Speaker Clustering for Speech Recognition Using. Vocal-Tract Parameters”. Speech Communication, vol. 36, no. 3, pp. 305-315, 2002

[4] TingYao Wu, Lie Lu, Hong-Jiang Zhang. "UBM-based Real-time Speaker Segmentation for Broadcasting News", Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP03), Vol. II, pp. 193-196, Hong Kong, April 4-10, 2003

[5] J. Ramírez, J. C. Segura, C. Benítez, A. de la Torre, A. Rubio, “Efficient voice activity detection algorithms using long-term speech information”, Speech communication, vol. 42, pp. 271-287, 2004.

[6] O. Pietquin, L. Couvreur, P. Couvreur, "Applied Clustering for Automatic Speaker-Based Segmentation of Audio Materials", Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL), Special Issue Operations Research and Statistics in the Universities of Mons, volume 41, no. 1-2, pp. 69-81, 2001.

[7] Jitendra Ajmera, Iain MCCowan and Hervé Bourlard, “Robust Speaker Change Detection”, IEEE Signal Processing Letters, Vol. 11, No.8, August 2004.

[8] R. Huang, J.H.L. Hansen, ”Unsupervised Audio Segmentation and Classification for Robust Spoken Document Retrieval”, IEEE ICASSP-2004, volume 1, pp. 741-744, May 2004.

[9] Xavier Anguera, Javier Hernando, “Evolutive Speaker Segmentation using a Repository System”, Proceedings of International Conference on Speech and Language Processing , ICSLP 2004 . Jeju Island, Korea . October 2004