A System for Speaker Detection and Tracking in Audio Broadcast News

Janez Žibert, Boštjan Vesnicer and France Mihelič
University of Ljubljana, Faculty of Electrical Engineering
Tržaška 25, SI-1000, Ljubljana, Slovenia
E-mail: janez.zibert@fe.uni-lj.si

Keywords: speaker diarization, speech detection, speaker clustering, audio indexing, speaker recognition, speaker tracking

Received: May 22, 2007

A system for speaker-based audio-indexing and an application for speaker-tracking in broadcast news audio are presented. The process of producing an indexing information in continuous audio streams based on detected speakers is composed of several tasks and is therefore treated as a multistage process. The main building blocks of such an indexing system include components for an audio segmentation, a speech detection, a speaker clustering and a speaker identification. We give an overview of each component of the system with emphasis to the approaches that are followed in each stage of building of our speaker-diarization and tracking system. The proposed system is evaluated on the audio data from the broadcast news domain, whereas we test each of the system's component and measure their impacts to the overall system's performance. The evaluation results indicate the importance of an audio segmentation and a speech detection module to the reliable performance of the whole system. Based on an indexing information produced by our system we also developed an application for searching target speakers in broadcast news. The application is designed in a way to be user-friendly and can be easily integrated in various computer environments.

Povzetek: Predstavljen je sistem za indeksacijo zvočnih posnetkov glede na govorce in aplikacija tega sistema za iskanje govorcev v zvočnih posnetkih informativnih oddaj.

1 Introduction

With the increasing availability of audio data derived from various multimedia sources comes an increasing need for efficient and effective means for searching and indexing through this type of information. Searching or tagging speech based on who is speaking is one of the more basic components required for dealing with spoken documents collected in large audio data archives, such as recordings of broadcast news or recorded meetings. In this paper, we focus on the indexing and searching of speakers in audio broadcast news (BN).

Audio data of BN shows present a typical multispeaker environment. The goal of searching and indexing of target speakers in such an environment is to find and identify the regions in the audio streams that belong to target speakers and produce an efficient way for accessing this regions from the audio data archives. The task of finding such speaker-defined regions is known as a speaker diarization task and was first introduced in the NIST[1] project of *Rich Transcription* in *'Who spoke when'* evaluations, [7]. The task of identifying the regions according to given speakers is known as a speaker tracking task and was first defined in 1999 NIST Speaker Recognition evaluation, [14]. While diarization and tracking procedures serve for a detection of speakers in audio data, is the purpose of speaker indexing an organization of audio data according to detected speakers for efficient speaker-based information audio-retrieval. In this paper, we present the approaches of speaker diarization and tracking in multispeaker audio BN data.

The paper is organized as follows. In the first sections, we describe in more detail a system for speaker diarization, which serves for speaker-indexing of BN shows. A system is composed of several components, which include procedures for an audio segmentation, a speech detection, a speaker clustering and a speaker identification. The first two procedures aim in detecting speaker and acoustic changes in speech portions of audio streams and thus correspond to partitioning of audio data to the homogeneous segments. The procedures for speaker clustering and identification are employed to group together segments of the same speaker and to provide speaker names to each such portion of speech data. Hence, they are used for tagging target speakers in the audio data. In Section 2, we give an overview of all of the above procedures, which were implemented to build a system for speaker tracking in BN shows. In the following section we present experiments and the evaluation results on the Slovenian audio BN database, where we explore the impact of each of the procedure on the overall speaker-tracking results. At the end, an application for speaker detection and tracking, based on the proposed methods, is described.

[1]National Institute of Standards and Technology, http://www.nist.gov/speech/

2 Speaker diarization in continuous audio streams

Speaker diarization is the process of partitioning input audio data into homogeneous segments according to the speaker's identities. The aim of speaker diarization is to improve the readability of an automatic transcription by structuring the audio stream into speaker turns, and in cases when used together with speaker-identification systems by providing the speaker's true identity. Such information is of interest to several speech- and audio-processing applications. For example, in automatic speech-recognition systems the information can be used for unsupervised speaker adaptation [1, 15], which can significantly improve the performance of speech recognition in large vocabulary continuous speech recognition (LVCSR) systems [10, 28, 4]. This information can also be applied for the indexing of multimedia documents, where homogeneous speaker or acoustic segments usually represent the basic units for indexing and searching in large archives of spoken audio documents, [13]. The outputs of a speaker diarization system could also be used in speaker-identification and in speaker-tracking systems, [6, 20], which was also the case in our presented application.

Most speaker diarization systems for a detection of speakers in continuous audio streams have a similar general architecture, [3, 26]. First, the signal is chopped into homogeneous segments. The segment boundaries are located by finding acoustic changes in the signal and each segment is expected to contain speech from only one speaker. Those segments, which do not represent speech data, are additionally detected and discarded from a further processing. The resulting segments are then clustered so that each cluster corresponds to only one speaker. At the final stage, each cluster is labeled by a corresponding speaker identification name or is left unlabeled, if the speech data in cluster do not correspond to any of the previously enrolled target speakers. As such, speaker diarization in continuous audio streams is a multistage process comprised by four main modules: an audio segmentation, a speech detection, a speaker clustering and a speaker identification.

A baseline speaker-indexing system architecture, which was followed in this work, is shown in Figure 1. First, the audio signal is processed in an *audio segmentation* module, where time-stamps are produced at the locations of detected acoustic changes. Audio data are thus partitioned into small homogeneous segments labeled by starting and ending time of each segment (segments: $[st_i, et_i]$ in Figure 1). It is expected that each such segment should contain data from just one acoustic source, i.e. speech from one speaker or non-speech data corresponding to music, silence or other non-speech source. Therefore, the obtained segments should be additionally divided to those, which contain speech or non-speech data. This is done in a *speech detection* module. Non-speech segments are marked as $[NS, st_i, et_i]$ in Figure 1 and are discarded from further processing. Only speech segments are then passed through a *speaker clustering* module. The aim of a speaker clustering is to merge speech segments from each speaker together, a major issue being that the information of speakers and the actual number of speakers are unknown *a priori* and need to be automatically determined. At this stage, just *relative* speaker labels are produced and segments are marked with automatically derived cluster names (segments $[C_i, st_i, et_i]$ in Figure 1). The true identities of the speakers are obtained in a *speaker identification* module in the next stage. Here, a multiple speaker verification of each cluster is performed. Speaker identification module is capable to recognize just those speakers, who are presented in the repository of the target speakers and are previously enrolled into the system. Speech data from clusters, which do not correspond to any of the speakers from target group, should be marked as *unknown* speaker data and are discarded from further processing.

Our speaker-indexing system [35] was designed in such a way, that all the modules include the standard approaches from similar state-of-the-art systems. In the following subsections each of the integrated module is described in more details.

2.1 Audio segmentation module

In general, spoken audio documents derived from BN shows include data from multiple audio sources, which may contain speech of different speakers as well as music segments, commercials and various types of noises, that are present in the background of BN reports. Another characteristic of BN audio documents is, that the data are delivered in the form of continuous audio streams. In order to efficiently process and extract the required information from such documents the continuously derived audio data should be adequately chopped into smaller portions of data, which are suitable for further processing. In the case of speaker-tracking applications the process of breaking the continuous audio streams into the homogeneous regions based on speaker turns is done in an audio segmentation module.

The segmentation of the audio data was made using the acoustic-change detection procedure based on the Bayesian Information Criterion (BIC), which was first proposed for the audio segmentation in [5] and improved by Tritschler and Gopinath in [27]. The applied procedure processed the audio data in a single pass while searching for change points within a window using a penalized likelihood ratio test (BIC score) of whether the data in the window is better modeled by a single probability distribution or two different distributions. If the estimated BIC score was under the given threshold (meaning that the data from the current window are better modeled by two probability distributions), a change point was detected and searching was restarted in the next window. In the opposite case, the analyzed window was extended and searching was redone. The threshold, which was implicitly included in the penalty term of the BIC score, has to be given in advance and was in our case estimated from the training data. The output

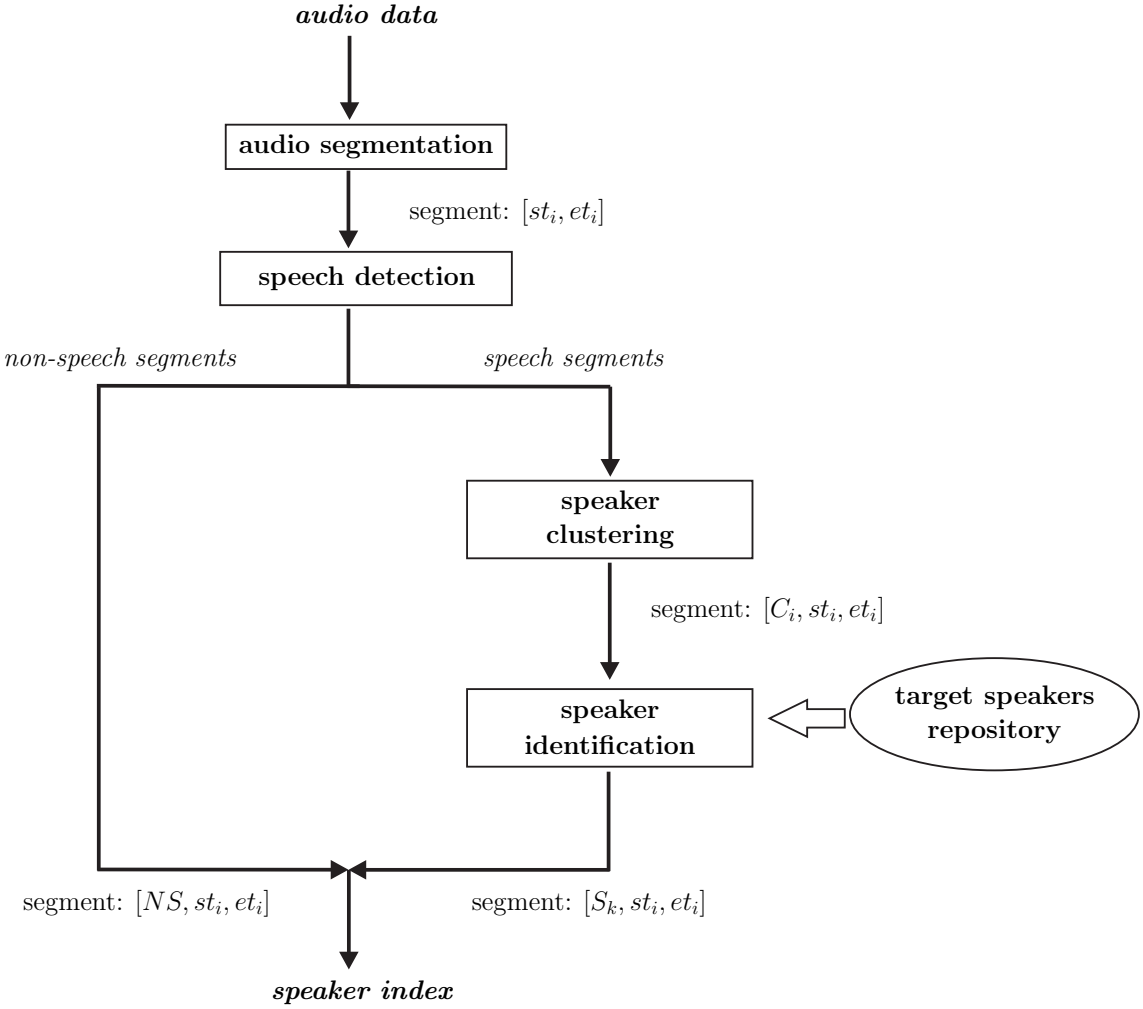

Figure 1: Main building blocks of a typical speaker-indexing system. Most systems have modules to perform speech detection, audio segmentation, speaker clustering and speaker identification, which may include component for gender detection.

of the audio segmentation module were acoustic change detection points, which defined basic audio segments for further processing.

This procedure is widely used in most of the current audio-segmentation systems [26, 7, 23, 30, 12, 33], and performed the best in comparison to alternative audio-segmentation approaches [26].

2.2 Speech detection module

The aim of this module in a speaker diarization system is to find the regions of speech in an audio stream. Since the audio stream was in our case already segmented into homogeneous regions of audio data based on acoustic changes, a speech detection module had to distinguish, which regions correspond to speech and non-speech data. The problem here represent non-speech data, which may consist of many acoustic phenomena such as silence, music, background noise or cross-talk.

The general approach used is a maximum likelihood classification with Gaussian Mixture Models (GMMs) estimated from acoustic representations of audio signals and trained on manually labeled training data [29, 19, 9, 23, 11, 24]. A speech detection based on such GMMs is performed either on pre-determined audio segments or by applying segmentation and detection together by using Viterbi decoding in the classification-network composed from trained GMMs. In both cases speech and non-speech data are usually modeled by several GMMs to cover various acoustic phenomena, which are expected in the processing audio data. To overcome this problem we proposed a new high-level representation of audio signals based on the phoneme recognition features, that are more suitable for speech/non-speech classification, [34, 16]. We developed four different measures based on consonant-vowel pairs and voiced-unvoiced regions obtained from phoneme speech recognizers and tested them in different segmentation-classification frameworks. The evaluation experiments on the BN au-

dio data, [34], proved that a combination of acoustic features – modeled by mel-frequency cepstral coefficients (MFCCs) – and our proposed phoneme-recognition features constituted the most powerful representation of audio data, which were robust enough and relatively unsensitive to different training and unforseen conditions. Hence, we also implemented this kind of fusion of acoustic and phoneme-recognition representations in our speech detection module. The speech detection was performed by using a standard maximum likelihood classification with just two GMMs (one model for speech and the other for non-speech data) on already segmented audio streams, which were obtained from the previously described audio segmentation module.

Detected speech segments were further passed to a speaker clustering module, while non-speech segments were discarded from further processing.

2.3 Speaker clustering module

The purpose of this stage is to associate or cluster segments from the same speaker together. The ideal clustering should produce one cluster for each speaker, which should include all segments of a given speaker.

The general clustering method, which was also followed in our speaker-indexing system, is to perform agglomerative clustering using bottom-up approach, [25]. The basic steps of the speaker-clustering algorithm based on this approach can be described in the following steps [35]:

1. *Initialization step:* Model each segment by a single Gaussian distribution.
2. *Merging step:* Use a BIC measure to estimate whether to join two clusters or not. The candidates for merging are those clusters, where the lowest BIC score is achieved.
3. *Stopping step:* Repeat the second step until some stopping criterion is not satisfied.

Since in our speaker-clustering approach a BIC measure was used for merging, clusters should be modeled by Gaussian distributions. In the initialization step each segment represents one cluster. In the merging step joining of clusters (segments) is performed by searching a minimum (or maximum, depending on BIC measure) BIC score among all possible pair-wise combinations of clusters. A BIC measure is usually the same as one used for audio segmentation and also possesses the same philosophy. It measures the difference when modeling the data from two separate clusters with two normal distributions and when modeling with just one. The low differences speak in favor of modeling the data with just one distribution, meaning that the data the most likely belong to just one audio source, i.e. one speaker in our case, while higher differences support hypothesis that the data from separate clusters correspond to different speakers. The merging process is generally stopped when the lowest BIC score is greater than a specified threshold, but there can be also applied other stopping criteria, [35]. The stopping criterion is critical for a good performance and depends on how the output to be used [26]. In our speaker-tracking system a stopping threshold was used, which was estimated from the development data to optimize the speaker clustering performance.

The output of the speaker clustering module contains segments with relative labels, which join speech segments of the same speaker together. Non-speech segments are treated in this stage as separate cluster. The task of such labeling of continuous audio streams is known as a speaker diarization task and can be used in various audio processing applications.

In this stage, several improvements can be made to increase a speaker diarization performance, like joint segmentation and clustering [17] and/or cluster recombination [31], but in the case of indexing information by speakers in our speaker-tracking system we found no additional gain in the performance when applying some of these methods.

2.4 Speaker identification module

Since speaker diarization systems only produce relative speaker labels (such as 'spk1'), additional modules for speaker identification has to be included into the system, when the true identities of the speakers are needed. This can be achieved in various ways. We decided to follow the standard approach of building speaker models for people who are likely to be in the news broadcasts (such as prominent politicians or main news anchors and reporters) and including these models in the last stage of the speaker-indexing system.

A speaker identification component was adopted from a speaker verification system, which was based on the state-of-the-art Gaussian Mixture Model Ű Universal Background model (GMM-UBM) approach, [22]. Such systems are in generally composed of an enrolment phase and a test phase. In the enrollment phase, a model of the client (target) speaker is built based on a client's speech data, while in the test phase, another speech data, which are in our case collected from speaker clusters, are tested against a hypothesized client model. As a result, a matching score is generated based on the likelihood ratio (LR) between the likelihood that the speech was produced by the claimed speaker and the likelihood that the speech was not provided by the claimed speaker. If the score is greater than a given threshold, the speaker is accepted (client trial), otherwise it is rejected (impostor trial). There have been many solutions proposed how to efficiently calculate the denominator of the LR, i.e. the likelihood that the given speech data were not uttered by the claimed speaker. The best results up to now are achieved when likelihoods are calculated by using UBMs, which are usually trained from pooled speech of a large number of different speakers [22]. These models also serve as a prior for deriving client speaker models by Bayesian technique called maximum aposteriori (MAP)

adaptation [8, 22], which was also applied in our speaker identification module.

In addition to that, we computed a new set of MFCC features, which were subjected to feature warping [21] to compensate different channel effects, and the log-likelihood scores normalization was performed at the end by applying ZT-normalization technique [2].

At the output of this stage the audio streams are equipped with the segment-time boundaries together with true speaker identification labels. Those clusters of segments, of which data do not belong to any of the enrolled speakers, get empty labels correspond to 'unknown' speakers. The output from this module also present the final results of the speaker-based audio-indexing and can be used for a detecting speakers in speaker-tracking applications. An application for speaker-tracking in BN shows, which was based on speaker's information obtained from our speaker-indexing system, is described in the last section.

3 Evaluation experiments

Evaluation of our speaker-based audio-indexing system was performed on the SiBN database [32], which consisted of 32 hours of BN shows in Slovenian language. 20 hours were used for an estimation of all the open parameters in all of the components of our indexing system, and the rest 12 hours served for the assessment of the system's performance.

The open parameters in the audio segmentation, the speech detection and the speaker clustering module corresponded to setting the thresholds to optimize the overall speaker diarization performance on the training data. In the audio segmentation module a threshold had to be estimated in the penalty factor of the BIC measure. It was set so to detect as many true change-detection points in the audio streams, while in the same time preserve low rate of miss-detected segment boundaries. The emphasis was put more on a detection of true segment boundaries, even if additional segment boundaries were falsely detected. In that case the over-segmented audio streams were produced, but they had almost no influence on the overall speaker diarization results while using them as inputs in speaker clustering module. In the case of under-segmented audio data it was found, that they could heavily degrade speaker-diarization and tracking performance. The same phenomena was explored in our speaker clustering module. Here, a threshold for stopping criteria of a merging process in a bottom-up clustering procedure had to be estimated. By setting a proper threshold we could optimize the speaker-diarization performance on the training data, but it was found out that this did not necessary reflect in the overall best performance of the speaker tracking system. The optimal performance was achieved in the case, when clusters did not contain speech from several speakers, i.e. a better performance was achieved in the under-clustering case, where speaker data were distributed over several clusters, rather than in the over-clustering case where too many contaminated clusters were produced containing speech from different speakers, which degraded a speaker-detection performance.

Another important issue was concerning a speech detection module. As was shown in [36] the impact of a speech detection in speaker diarization and tracking systems is direct and indirect. Since, non-speech data are treated as data from one of the speakers in the speaker-tracking system, a speech detection has a direct influence on the speaker-tracking results. On the other hand, an erroneous speech/non-speech classification of audio segments in the speaker-indexing system influences a speaker clustering and identification performance. Therefore, a good speech detection in continuous audio streams is a necessary pre-processing step for achieving a good speaker-diarization and tracking results. Since we decided to use a fusion of acoustic and phoneme-recognition features in a speech detection module, we had to apply a simplified version of a phoneme recognizer for deriving phoneme-recognition features. The recognizer was built on a standard way, using Hidden Markov models (HMMs) trained on Slovenian data. In addition, we had to estimate two GMMs for detecting speech and non-speech data, which were estimated from the training part of the SiBN database.

Since in a speaker identification module a true detection of speakers was carried out, GMM of each target speaker had to be provided. They were built from UBMs, which were trained on the speech data of the training part from the SiBN database. We were designed two UBMs corresponding to female and male speech data. All the models were constituted from 1024 Gaussian mixtures, which were estimated using Baum-Welch Expectation-Maximization (EM) algorithm. The GMM model for each target speaker was derived from corresponding UBM using MAP adaptation technique in a standard way, [22]. The evaluated system was capable to detect 41 target speakers, which were extracted from the training data in the enrollment phase. In the test phase, data from each cluster were compared against all of the models from target-speakers repository and LR score were produced. In the evaluation phase no additional score threshold was proposed, since we tried to evaluate the system in the whole range of all the possible operating points.

Note that gender-dependent UBMs were used for deriving speaker-dependent GMMs, meaning that in the test phase a gender classification was performed at first by using the same gender UBMs, which were also applied in the estimation of the target speaker models.

All modules in the tested system were built by using our own tools. The procedures for audio segmentation and speaker clustering were implemented in C/C++ programming environment, whereas the same component for the computation of the BIC measure was integrated in both modules. The fusion of acoustic and phoneme-recognition features was in the speech-detection module applied by

performing a Viterbi decoding on a classification network of speech and non-speech GMMs. The training of GMMs and decoding through the network was done by using *HTK Toolkit*, [37], while the acoustic and phoneme-recognition features were produced by our own tools. The same set of acoustic features was then used in the speaker identification module, where all the training and testing procedures were also implemented by our own tools.

3.1 Evaluation results

Since several modules were included in the speaker-based audio-indexing system of BN shows, series of experiments were performed to measure the impact of each module to the overall speaker-tracking results.

Overall results of the evaluated speaker-tracking systems are depicted in Figure 2. The results are presented in terms of false acceptance (FA) and false rejection (FR) rates (false alarm and miss probabilities in Figure 2), measured at different operating points in the form of Detection Error Trade-off (DET) curves, [14]. In our case, the evaluated speaker tracking systems were capable to detect 41 target speakers from the audio data, which include 551 different speakers. The performances of the evaluated systems were therefore assessed by including all 41 target speakers with the addition of non-speech segments and the results were produced by FA and FR rates measured at the time (frame) level.

Figure 2 presents the evaluation results from six tested speaker-tracking systems, whereas different versions of system's components were combined. Only the speaker identification module was the same in all the evaluations, while other components (audio segmentation, speech detection and speaker clustering module) were combined by applying manual or automatic version of each procedure. In addition to that, two versions of speaker-tracking system without speaker clustering were also tested. In this way we tried to estimate the gain of each component to the overall speaker-tracking results. In Figure 2, a procedure for speech detection is marked as *S/N* (referring to speech/non-speech detection), procedures for audio segmentation are marked as *S* and for speaker clustering *C*. Manual versions of each procedure are abbreviated as *man*, automatic versions as *aut*, and in systems, where one of the procedure was missing, an abbreviation *w/o* is used for that procedure. For example, a system where manual audio segmentation was used prior to automatic speech/non-speech detection and automatic speaker clustering is in Figure 2 marked as *(aut S/N man S aut C)*, a system, where everything was performed automatically, is marked as *(aut S/N aut S aut C)*, etc.

The evaluation results in Figure 2 are displayed in terms of DET curves. They are ranging from the best performance of a system, where all procedures, except speaker identification, were preformed manually, to a system, where all the procedures were performed automatically.

The impact of speaker clustering were explored in series of experiments with systems *(man S/N man S man C)*, *(man S/N man S aut C)*, and *(man S/N man S w/o C)*, where audio segmentation and speech detection were performed manually, and with systems *(aut S/N aut S aut C)* and *(aut S/N aut S w/o C)*, where *S* and *S/N* procedures were performed automatically. Results from Figure 2 reveal expected performances of these systems. The best results were obtained in a system where everything was carried out manually, next to them are results from the system, where just speaker clustering procedure was applied automatically, and at the end are systems where all three procedures were done automatically. A comparison of the system *(man S/N man S man C)* and the system *(man S/N man S aut C)* indicates that a proper speaker clustering can significantlly improve the overall performance of a speaker-tracking system. The same can be concluded for speech detection and audio segmentation tasks by expecting the performances of the systems *(man S/N man S aut C)* and *(aut S/N aut S aut C)*. When applying automatic versions of audio segmentation and speech detection into the speaker-tracking system (by using the same speaker clustering procedure), the overall results of the system dropped for around 3% in the whole range of the operating points. Another important issue can be observed by expecting the systems *(man S/N man S aut C)* and *(man S/N man S w/o C)*, and the systems *(aut S/N aut S aut C)* and *(aut S/N aut S w/o C)*. In this evaluations we investigated whether it is better to use a speaker clustering procedure in speaker-tracking systems or not. As can be seen from Figure 2 there is no so much difference in the performances of the systems, where clustering was applied, to those without clustering. Our tracking results with automatic clusterings show that just a marginal gain could be obtained. This indicates that in our case the speaker-tracking system could not benefit from speaker clustering. The same was shown in the study of speaker tracking of radio broadcast news in [18], where it was concluded that a speaker identification can even help to improve a speaker clustering performance and not a vice versa.

The influence of an audio segmentation to the overall speaker-tracking results was explored by the evaluation of the systems *(aut S/N man S aut C)* and *(aut S/N aut S aut C)*. In the first case the audio segmentation was performed manually, while in the second the audio segmentation procedure, described in Section 2.1, was applied. As can be seen from the results in Figure 2 a manual segmentation outperforms an automatic version by approximately 3% in the whole range of operating points. This means that an audio segmentation plays an important role in our evaluated speaker-tracking system. Since segmentation procedure is usually applied in the first steps of speaker-tracking systems, the errors from segmentation have impact on all subsequent procedures. In our case, the errors in detecting change points in continuous audio streams produced non-homogeneous segments, which caused unreliable detection of speech/non-speech regions and unreliable detection of target speakers as well. Consequently, both types of er-

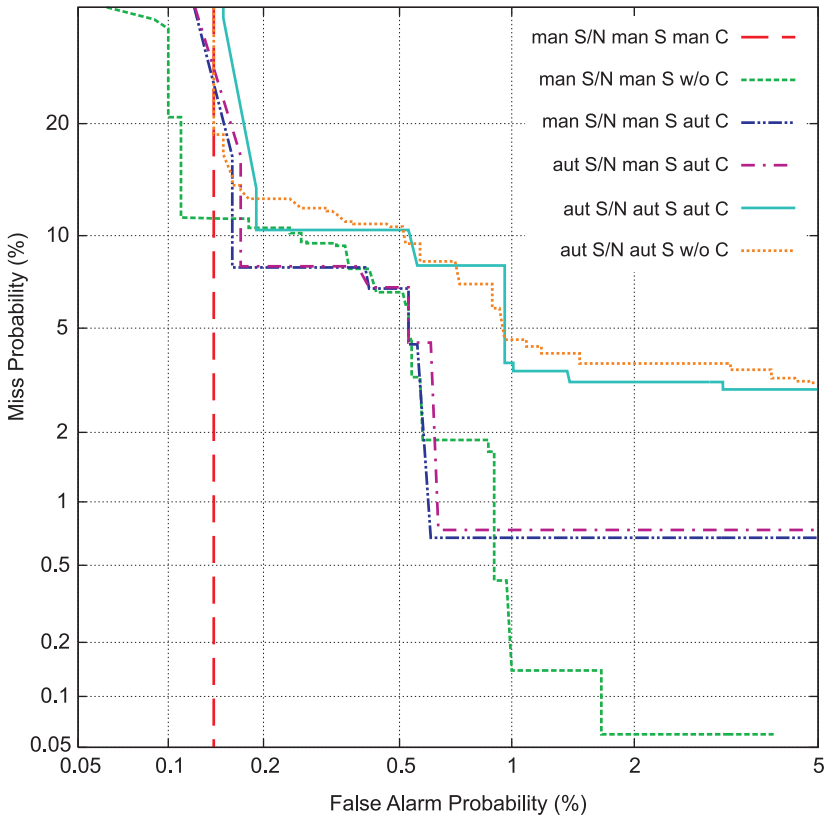

Figure 2: The overall speaker-tracking results of six evaluated systems. Lower DET values correspond to better performance.

rors were therefore integrated into the overall results of the evaluated system *(aut S/N aut S aut C)*.

Another evaluation perspective can be obtained by exploring systems *(man S/N man S aut C)* and *(aut S/N man S aut C)*. By comparing evaluation results of both systems we can estimate the gain of the speech detection procedure alone to the overall speaker-tracking results. As can be seen from the evaluation results in Figure 2, is the difference in the overall performances of systems, when using a manual and an automatic version of speech detection procedure, minimal. This marginal difference in the DET results was achieved due to the usage of the manual audio segmentation procedure in both systems. By applying a speech/non-speech detection procedure in a combination with manual segmentation (described in Section 2.2), a surprisingly high overall speech/non-speech accuracy of 99.38% was achieved, which resulted in the minimal difference of both evaluated systems. Note, that we used our own method for speech/non-speech detection, which proved to be a better choice for the speaker-diarisation and tracking tasks, as it was shown in a comparison study in [36].

To sum up, the comparison of the evaluation results of the different versions of the speaker-tracking system provides valuable insights of how the system works and which components of the system have greater impact on the overall performance. The overall results reveal an acceptable performance of the system, where all of the system's procedures were performed automatically. All other evaluated versions of the system serve for the estimation of the impact of each component to the overall speaker-tracking performance. It was found out that probably the most important component of the system is an audio segmentation module. If a segmentation procedure produces too many non-homogeneous segments due to improper detected change points in an audio stream, causes unreliable performances of a speech-detection and a speaker-identification module, and thus degrades the overall system's performance. As far as concerning speech detection module alone it was also shown, that we could gain some improvement in the overall system's performance by applying a good speech detection procedure. Since in our evaluated system a speech detection procedure, proposed in [34], was applied, almost no loss of the overall performance was obtained. Another important finding was concerning speaker clustering. The evaluation showed no important gain in the overall results, when speaker clustering was applied or not. This was in accordance with findings in [18].

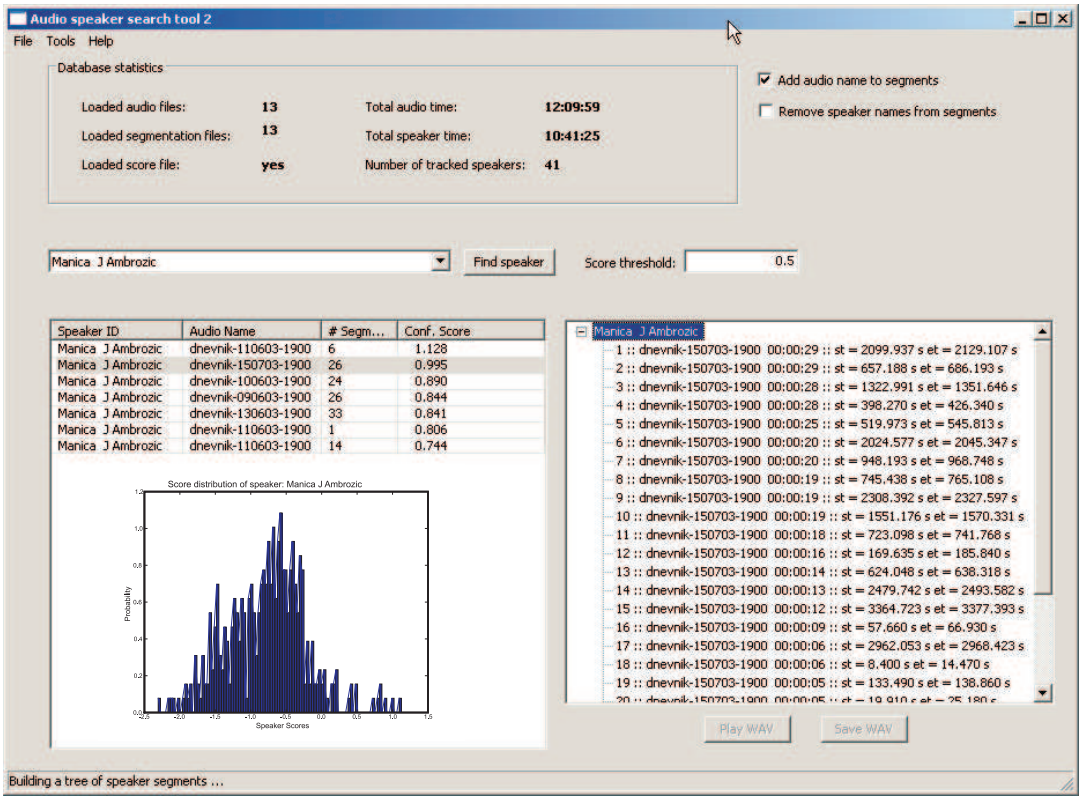

Figure 3: A graphical user interface of a speaker-tracking system.

4 Speaker tracking system

A derivation of indexing information by speakers is an important step in applications, which are used for searching speakers in large archives of audio data. In this section we present one such application for a detection of speakers in continuous audio streams of BN shows, which are based on the system for a speaker-based audio-indexing presented in previous sections.

The application was designed in a way to separate processes of audio-indexing and searching of target speakers. This is also a standard approach in search engines based on text data. The indexing process is usually done once in a while, i.e. when new data arrive and index has to be updated, while searching of information is done all the time.

In our application the process of audio-indexing was performed once on the BN data from the SiBN database. The output of this process were time boundaries of speech segments with target-speaker's scores. And in the searching process the audio segments corresponding to a given speaker have to be provided and properly displayed to a user. A graphical user interface of our searching application is shown in Figure 3.

In the top pane of the application in Figure 3 are displayed some base properties of an audio database, which is currently loaded into the system, i.e. the information of total audio data time, of speech data time, how many speakers can the system detect, etc. In the middle pane is the search form, which includes a list of all possible target speakers, which the system is capable to find, and the score threshold, that can be optionally set to return just those speech segments, where speaker-detection scores are above the given threshold. The two bottom panes display information of the speaker, who has to be found by hitting the *Find-speaker* button. The bottom-left pane are filled with cluster information corresponding to a searched speaker, which are in the case of BN data divided to each BN show. The clusters are by default sorted by a confidence score, but the application also provides other sort possibilities, i.e. sorting by BN show name, number of segments in cluster, speaker name, etc. At the bottom of this pane it is also showed a histogram of the LR scores of a given speaker from all possible clusters in the database. A speaker score-distribution displayed in a histogram can serve for estimating the optimal threshold for obtaining just speech data of the current speaker . In this way a user can control the amount of data that are displayed and can inspect how likely the current data correspond to a searched speaker. The right-bottom pane in Figure 3 displays a list of all segments of a target speaker's cluster, which is marked in the left-bottom pane. A change in cluster (in the left-bottom pane) cause a fill of a new list with segments of that marked cluster. A user can listen or save the audio data by clicking on one of the displayed segments.

This application was developed by using Python programming language, while a graphical user interface of

the application was designed by using *wxPython cross-platform GUI Toolkit*[2]. The application was implemented to operate as a stand-alone process and currently works just for searching speakers in BN shows, but it can be easily extended for other types of audio documents. Since the application expects that the audio-indexing is done beforehand, it is also independent of the methods used in the audio-indexing system. As such, it can be integrated in various types of computer applications and environments.

5 Conclusion

A system for speaker-based audio-indexing and an application for speaker-tracking in BN audio data based on this system were presented. We gave an overview of four main building blocks of such audio-indexing system and provide an extensive evaluation of all of the system's components. While they were modules for an audio segmentation, a speaker clustering and an identification implemented by using the latest state-of-the-art approaches, was in a module for speech detection followed our own approach of incorporating phoneme-recognition features in a classification process. In the evaluation experiments the impact of each module to the overall speaker-tracking performance was measured. It was found out that the most critical component of such a system is an audio segmentation module, since it is usually applied in the first processing stages of such system and its poor performance causes unreliable performances of all other components. Nevertheless, the evaluation results demonstrate an acceptable performance of the system, where all of the procedures were performed automatically. This system were later applied for an audio-indexing of BN shows in a speaker-tracking application. An application was designed to serve as a search tool for speakers, who are likely to be in the news broadcasts, but it could be easily extended for other types of audio documents.

Acknowledgement

This work was supported by the two Slovenian Research Agency (ARRS), development projects: L2-6277 (C) entitled "Broadcast news processing system based on speech technologies" and M2-0210 (C) entitled "AvID: Audiovisual speaker identification and emotion detection for secure communications."

References

[1] T. Anastasakos, J. McDonough, R. Schwartz, J. Makhoul (1996) A Compact Model for Speaker-Adaptive Training, *Proceedings of International Conference on Spoken Language Processing (ICSLP1996)*, Philadelphia, PA, USA, 1996, pp. 1137-1140.

[2] http://www.wxpython.org/

[2] R. Auckenthaler, M. Carey, & H. Lloyd-Thomas (2000) Score normalization for text-independent speaker verification system, *Digital Signal Processing*, Vol. 10, No. 1, January 2000, pp. 42-54.

[3] C. Barras, X. Zhu, S. Meignier, & J.-L. Gauvain (2006) Multistage Speaker Diarization of Broadcast News, *IEEE Transactions on Speech, Audio and Language Processing, Special Issue on Rich Transcription*, Vol. 14, No. 5, September 2006, pp. 1505-1512.

[4] P. Beyerlein, X. Aubert, R. Haeb-Umbach, M. Harris, D. Klakow, A. Wendemuth, S. Molau, H. Ney, M. Pitz & A. Sixtus (2002) Large vocabulary continuous speech recognition of Broadcast News Ű The Philips/RWTH approach, *Speech Communications*, Vol. 37, No. 1-2, May 2002, pp. 109-131.

[5] S. S. Chen & P. S. Gopalakrishnan (1998) Speaker, environment and channel change detection and clustering via the Bayesian information criterion, *Proceedings of the DARPA Speech Recognition Workshop*, Lansdowne, Virginia, USA, February 1998, pp. 127-132.

[6] P. Delacourt, J. Bonastre, C. Fredouille, T. Merlin & C. Wellekens (2000) A Speaker Tracking System Based on Speaker Turn Detection for NIST Evaluation, *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP2000)*, Istanbul, Turkey, June, 2000.

[7] J. Fiscus, J. S. Garofolo, A. Le, A. F. Martin, D. S. Pallet, M. A. Przybocki & G. Sanders (2004) Results of the Fall 2004 STT and MDE Evaluation, *Proceeedings of the Fall 2004 Rich Transcription Workshop*, Palisades, NY, USA, November 2004.

[8] J. L. Gauvain & C.-H. Lee (1994) Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, *IEEE Trans. Speech, and Audio Processing*, Vol. 2, No. 2, April 1994, pp. 291-298.

[9] J.-L. Gauvain, L. Lamel & G. Adda (1998) Partitioning and Transcription of Broadcast News Data, *Proceeedings of the International Conference on Spoken Language Processing (ICSLP98)*, Sydney, Australia, December 1998, pp. 1335-1338.

[10] J. L. Gauvain, L. Lamel & G. Adda (2002) The LIMSI Broadcast News transcription system, *Speech Communications*, Vol. 37, No. 1-2, May 2002, pp. 89-108.

[11] T. Hain, S. E. Johnson, A. Tuerk, P. C. Woodland & S. J. Young (1998) Segment Generation and Clustering in the HTK Broadcast News Transcription System, *Proceeedings of the 1998 DARPA Broadcast News Transcription System*, Lansdowne, VA, USA, 1998, pp. 133-137.

[12] D. Istrate, N. Scheffer, C. Fredouille & J.-F. Bonastre (2005) Broadcast News Speaker Tracking for ESTER 2005 Campaign, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005, pp. 2445-2448.

[13] J. Makhoul, F. Kubala, T. Leek, D. Liu, L. Nguyen, R. Schwartz & A. Srivastava (2000) Speech and language technologies for audio indexing and retrieval, *Proceedings of the IEEE*, Vol. 88, No. 8, 2000, pp. 1338-1353.

[14] A. Martin, M. Przybocki, G. Doddington & D. Reynolds (2000) The NIST speaker recognition evaluation - overview, methodology, systems, results, perspectives, *Speech Communications*, Vol. 31, No. 2-3, June 2000, pp. 225-254.

[15] S. Matsoukas, R. Schwartz, H. Jin & L. Nguyen (1997) Practical Implementations of Speaker-Adaptive Training, *Proceedings of the 1997 DARPA Speech Recognition Workshop*, Chantilly VA, USA, February 1997, pp. 1137-1140.

[16] F. Mihelic & J. Zibert (2006) Robust speech detection based on phoneme recognition features, *Proceedings of Text, speech and dialogue (TSD 2006)*, Brno, Czech Republic, September 2006, pp. 455-462.

[17] S. Meignier, J.-F. Bonastre, C. Fredouille & T. Merlin (2000) Evolutive HMM for Multi-Speaker Tracking System, *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2000)*, Istanbul, Turkey, June 2000.

[18] D. Moraru, M. Ben & G. Gravier (2005) Experiments on speaker tracking and segmentation in radio broadcast news, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005.

[19] P. Nguyen, L. Rigazio, Y. Moh & J. C. Junqua (2002) Rich Transcription 2002 Site Report. Panasonic Speech Technology Laboratory (PSTL), *Proceeedings of the 2002 Rich Transcription Workshop*, Vienna, VA, USA, May 2002.

[20] B.Nedic, G. Gravier, J. Kharroubi, G. Chollet, D. Petrovska, G. Durou, F. Bimbot, R. Blouet, M. Seck, J.-F. Bonastre, C. Fredouille, T. Merlin, I. Magrin-Chagnolleau, S. Pigeon, P. Verlinde, & J. Cernocky (1999) The Elisa'99 Speaker Recognition and Tracking Systems, *Proceedings of IEEE Workshop on Automatic Advanced Technologies*, 1999.

[21] J. Pelecanos & S. Sridharan (2001) Feature warping for robust speaker verification, *Proceedings of the Speaker Odyssey Workshop*, Crete, Greece, June 2001, pp. 213Ű218.

[22] D. A. Reynolds, T. F. Quatieri, & R. B. Dunn (2000) Speaker verification using adapted Gaussian mixture models, *Digital Signal Processing*, Vol. 10, No. 1, January 2000, pp. 19-41.

[23] D. A. Reynolds & P. Torres-Carrasquillo (2004) The MIT Lincoln Laboratory RT-04F Diarization Systems: Applications to Broadcast Audio and Telephone Conversations, *Proceeedings of the Fall 2004 Rich Transcription Workshop*, Palisades, NY, USA, November 2004.

[24] R. Sinha, S. E. tranter, M. J. F. Gales & P. J. Woodland (2005) The Cambridge University March 2005 speaker diarisation system, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005, pp. 2437-2440.

[25] S. Theodoridis & K. Koutroumbas (2003) *Pattern Recognition (2nd edition)*, Academic Press, 2003.

[26] S. Tranter & D. Reynolds (2006) An Overview of Automatic Speaker Diarisation Systems, *IEEE Transactions on Speech, Audio and Language Processing, Special Issue on Rich Transcription*, Vol. 14, No. 5, September 2006, pp. 1557-1565.

[27] A. Tritschler & R. Gopinath (1999) Improved speaker segmentation and segments clustering using the Bayesian information criterion, *Proceedings of EUROSPEECH 99*, Budapest, Hungary, September 1999, pp. 679-682.

[28] P. C. Woodland (2002) The development of the HTK Broadcast News transcription system: An overview, *Speech Communications*, Vol. 37, No. 1-2, May 2002, pp. 47-67.

[29] C. Wooters, J. Fung, B. Peskin & X. Anguera (2004) Towards Robust Speaker Segmentation: The ICSI-SRI Fall 2004 Diarization System, *Proceeedings of the Fall 2004 Rich Transcription Workshop*, Palisades, NY, USA, November 2004.

[30] B. Zhou & J. Hansen, (2000) Unsupervised Audio Stream Segmentation and Clustering via the Bayesian Information Criterion, *Proceedings of International Conference on Spoken Language Processing (ICSLP 2000)*, Beijing, China, October 2000, pp. 714-717.

[31] X. Zhu, C. Barras, S. Meignier & J.-L. Gauvain (2005) Combining Speaker Identification and BIC for Speaker Diarization, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005, pp. 2437-2440.

[32] J. Žibert & F. Mihelič (2004) Development of Slovenian Broadcast News Speech Database, *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, May 2004, pp. 2095-2098.

[33] J. Žibert, F. Mihelič, J.-P. Martens, H. Meinedo, J. Neto, L. Docio, C. Garcia-Mateo, P. David, J. Zdansky, M. Pleva, A. Cizmar, A. Žgank, Z. Kačič, C. Teleki & K. Vicsi (2005) The COST278 Broadcast News Segmentation and Speaker Clustering Evaluation - Overview, Methodology, Systems, Results, *Proceedings of Interspeech 2005 - Eurospeech*, Lisbon, Portugal, September 2005, pp. 629-632.

[34] J. Žibert, N. Pavešić & F. Mihelič (2006) Speech/Non-Speech Segmentation Based on Phoneme Recognition Features, *EURASIP Journal on Applied Signal Processing*, Vol. 2006, No. 6, Article ID 90495, pp. 1-13.

[35] J. Žibert (2006) *Obdelava in analiza zvočnih posnetkov informativnih oddaj z uporabo govornih tehologij*, PhD thesis, Faculty of electrical engineering, University of Ljubljana, 2006.

[36] J. Žibert, B. Vesnicer & F. Mihelič (2007) Novel Approaches to Speech Detection in the Processing of Continuous Audio Streams, *Robust Speech Recognition and Understanding*, M. Grimm, K. Kroschel (Eds.), I-Tech Education and Publishing, 2007, pp. 23-48.

[37] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev & P. Woodland (2004) *The HTK Book (for HTK Version 3.2)*, Cambridge University Engineering Department, Cambridge, United Kingdom, 2004.