Journal of Computing and Information Technology - CIT 16, 2008, 3, 183–195
doi:10.2498/cit.1001067

Development of a Speaker Diarization System for Speaker Tracking in Audio Broadcast News: a Case Study*

Janez Žibert, Boštjan Vesnicer and France Mihelič

Faculty of Electrical Engineering, University of Ljubljana, Slovenia

A system for speaker tracking in broadcast-news audio data is presented and the impacts of the main components of the system to the overall speaker-tracking performance are evaluated. The process of speaker tracking in continuous audio streams involves several processing tasks and is therefore treated as a multistage process. The main building blocks of such system include the components for audio segmentation, speech detection, speaker clustering and speaker identification. The aim of the first three processes is to find homogeneous regions in continuous audio streams that belong to one speaker and to join each region of the same speaker together. The task of organizing the audio data in this way is known as a speaker diarization and plays an important role in various speech-processing applications. In our case the impact of speaker diarization was assessed in a speaker-tracking system by performing a comparative study of how each of the component influenced the overall speaker-detection results. The evaluation experiments were performed on broadcast-news audio data with a speaker-tracking system, which was capable of detecting 41 target speakers. We implemented several different approaches in each component of the system and compared their performances by inspecting the final speaker-tracking results. The evaluation results indicate the importance of the audio-segmentation and speech-detection components, while no significant improvement of the overall results was achieved by additionally including a speaker-clustering component to the speaker-tracking system.

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

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 through and indexing 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).

The audio data of BN shows present a typical multi-speaker environment. The goal when searching and indexing target speakers in such an environment is to find and identify the regions in the audio streams that belong to the target speakers and produce an efficient way for accessing these regions in 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 *Rich Transcription* project in *'Who spoke when'* evaluations, [8]. The task of identifying the regions associated with particular speakers is known as a speaker-tracking task and was defined during a 1999 NIST Speaker Recognition evaluation, [16]. While diarization and tracking procedures serve for the detection of speakers in audio data, the purpose of speaker indexing is the organization of audio data according to detected speakers for efficient speaker-based audio-retrieval. In this paper we present approaches of speaker diarization and tracking in multi-speaker audio BN data and measure their influences to the overall speaker-tracking performance.

*This work was supported by the Slovenian Research Agency (ARRS), development project M2-0210 (C) entitled "AvID: Audiovisual speaker identification and emotion detection for secure communications."

The paper is organized as follows. In Section 2 we describe in more detail a system for speaker diarization, that is composed of several components, which include procedures for audio segmentation, speech detection, speaker clustering and speaker identification. In the following subsections we give an overview of all of the above procedures and provide more details of the approaches that were implemented to build a system for speaker tracking in BN shows. Each component of the system was separately tested and different approaches were compared. In Section 3 we present experiments and evaluate the results for the Slovenian audio BN database, where we explore the impact of each of the procedures on the overall speaker-tracking results. Finally, discussion of results and conclusions are given in the last sections.

2. Speaker Diarization in Continuous Audio Streams

Speaker diarization is the process of partitioning input audio data into homogeneous segments according to the speakers' 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 it is used together with speaker-identification systems, to provide the speaker's true identity. Such information is of interest for several speech- and audio-processing applications. For example, in automatic speech-recognition systems, the information can be used for unsupervised speaker adaptation [1, 17], which can significantly improve the performance of speech recognition in large-vocabulary continuous-speech-recognition (LVCSR) systems [11, 33, 4]. This information can be also 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, [15]. The outputs of a speaker-diarization system are also used in speaker-identification and speaker-tracking systems, [6, 22], which was also the case in our presented application.

Most speaker-diarization systems for the detection of speakers in continuous audio streams have a similar general architecture, [3, 31]. First, the signal is chopped into homogeneous segments. The segment boundaries are located by finding the acoustic changes in the signal, with each segment expected to contain speech from just one speaker. Those segments that do not represent speech data are additionally detected and discarded from any further processing. The resulting segments are then clustered according to speakers, i.e., all segments of one speaker are grouped in a cluster. During the final stage, each cluster is labeled with a corresponding speaker-identification name, or it is left unlabeled if the speech data in the cluster do not correspond to any of the previously enrolled target speakers. As such, speaker diarization in continuous audio streams is a multistage process made up of four main modules: audio segmentation, speech detection, speaker clustering and speaker identification.

The baseline speaker-indexing system architecture, that was followed in this study, is shown in Figure 1. First, the audio signal is processed in an *audio-segmentation* module, where timestamps are produced at the locations of the detected acoustic changes. Audio data are thus partitioned into small homogeneous segments labeled with the starting and ending times of each segment (segments: $[st_i, et_i]$ in Figure 1). It is expected that each such segment contains data from just one acoustic source, i.e., the speech from one speaker or non-speech data corresponding to music, silence or another non-speech source. Therefore, the obtained segments should be additionally divided into those containing either speech or non-speech data; this procedure requires a *speech-detection* module. The non-speech segments are marked as $[NS, st_i, et_i]$ in Figure 1 and are discarded from further processing. Only the speech segments are then passed through the *speaker-clustering* module. The aim of speaker clustering is to merge the speech segments from each speaker together. Here, a major problem is that the information about the speakers and the actual number of speakers are unknown *a priori* and need to be automatically determined. At this stage, only *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

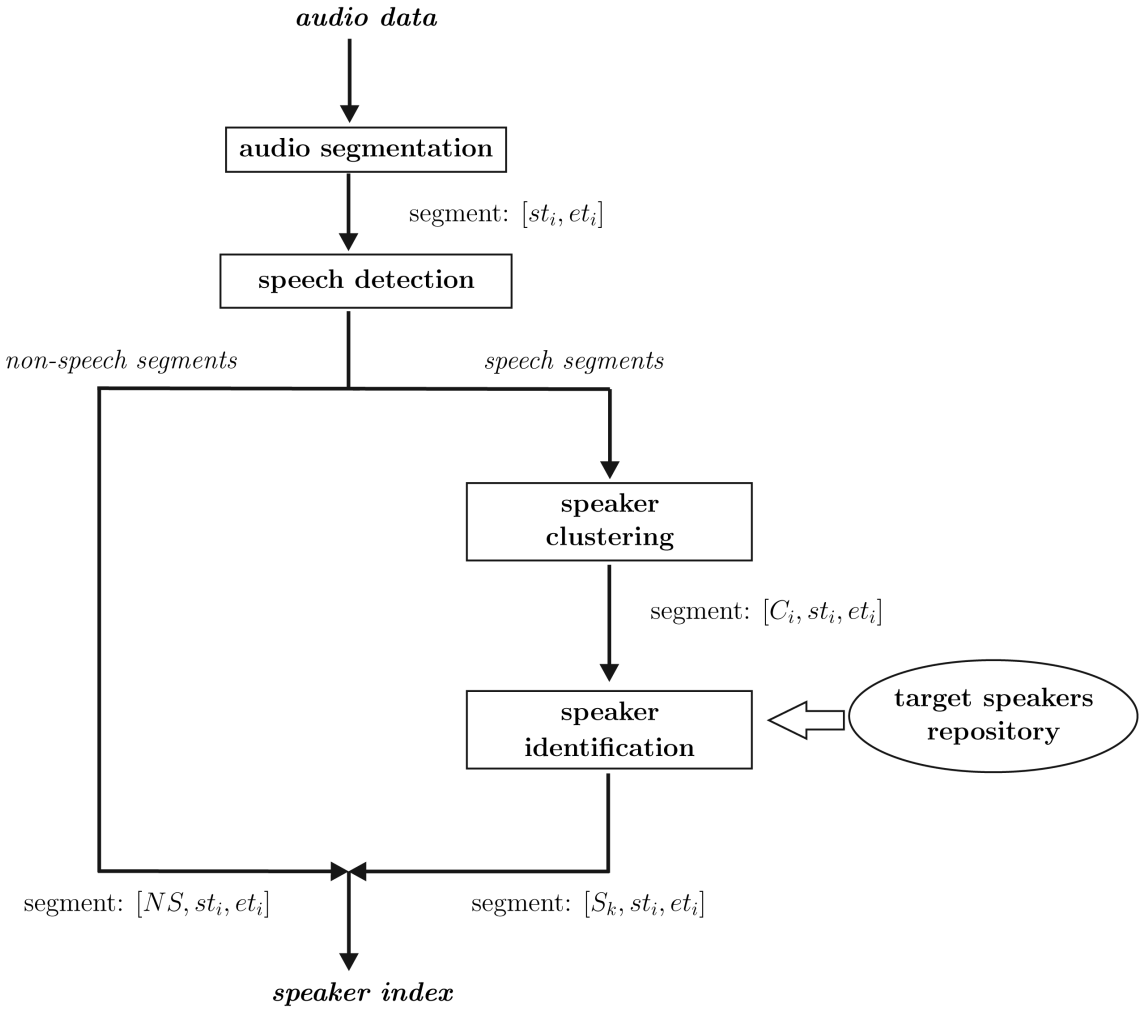

Figure 1. Main building blocks of a typical speaker-diarization system. Most systems have modules to perform speech detection, audio segmentation, speaker clustering and speaker identification, which may include a component for gender detection.

verification of each cluster is performed. A speaker-identification module is capable of recognizing just those speakers, who are present in the repository of target speakers and are previously enrolled in the system. The speech data from clusters that do not correspond to any of the speakers in the target group should be marked as *unknown-speaker* data. At the end, a speaker index is derived, which is used as a basis for searching and tracking speakers in the audio database.

In our speaker-based indexing system all of the components of the system were implemented in such a way that in each processing task different approaches could be applied. In the following subsections each component of the system is described in more details.

2.1. Audio Segmentation

We implemented two different audio-segmentation procedures, which both aimed to find time-stamps in audio streams at changes between different speakers or acoustic environments and were both based on the Bayesian Information Criterion (BIC), [5].

When the BIC is used as a model selection criterion for the audio segmentation, a problem of change detection is reformulated as a model selection task between two competing models and is defined as follows. If we assume that each acoustically homogeneous segment, which is represented by a sequence of frame-based acoustic feature vectors, i.e., $Z = x_1, \ldots x_t, \ldots, x_N$, can be modeled as one multivariate Gaussian process $Z \sim N(\mu, \Sigma_Z)$, the

detection of change points can be presented as a model selection problem between the following two nested models [5]:

$$\begin{aligned} M_1 &: \; Z = x_1, \ldots x_t, \ldots, x_N \sim N(\mu_Z, \Sigma_Z) \\ \text{and} \quad M_2 &: \; X = x_1, \ldots x_t \sim N(\mu_X, \Sigma_X); \\ &\quad\; Y = x_{t+1}, \ldots x_N \sim N(\mu_Y, \Sigma_Y). \end{aligned}$$

The first model M_1 assumes that all data are derived from a single Gaussian process, while the model M_2 assumes that the data to point t are drawn from one Gaussian, while the last $N-t$ data samples are drawn from another Gaussian. One of the possible measures for choosing the model that better suits to the given data presents a difference in BIC values between these two models that is computed as:

$$\Delta_{BIC}^{X,Y}(t) = \frac{1}{2}\Big(N\log|\Sigma_Z| - t\log|\Sigma_X| - (N-t)\log|\Sigma_Y|\Big) - \frac{\lambda}{2}\Big(d+\frac{1}{2}d(d+1)\Big)\log N, \quad (1)$$

where Σ_X, Σ_Y and Σ_Z are full-covariance matrices of the Gaussian distributions, estimated from the data X, Y and Z, respectively. d is a dimension of the features vectors x_i. While the first term in (1) corresponds to a difference in log-likelihoods of models M_2 and M_1, the second term presents a difference in number of parameters of both models. The first term accounts for the quality of the match between the model and the data, while the second one is a penalty for the model complexity with λ allowing the tuning of the balance between the two terms.

In the first segmentation procedure a *standard approach* of finding acoustic-change detection points was followed, which was first proposed in [5] and improved in [32]. This procedure processed the audio data in a single pass while searching for change points within a window using the Δ_{BIC} measure, defined in (1). Candidates for segment boundaries were points t, where $\Delta_{BIC}^{X,Y}(t) > 0$, and among them a point with the highest $\Delta_{BIC}^{X,Y}$ score was selected as a change point. In this case, the window was moved to that position, while the length of the window was set to the initial size, and the computation of Δ_{BIC} continued within the new window. If there were no change points in the initial window ($\Delta_{BIC}^{X,Y}(t) < 0$ for all points t within the window), a window was increased by additional length, and the computation of Δ_{BIC} was redone on the extended window. These steps were repeated until there were no more data for processing. The threshold, which was implicitly included in the penalty term of the BIC score, had to be given in advance and was in our case estimated from the training data.
This procedure is widely used in most of the current audio-segmentation systems [31, 8, 27, 35, 13, 38].

The alternative procedure, which was also tested in our system, was based on a DISTBIC approach, [7]. A segmentation with this approach was done in two passes. In the first pass, candidate points for change detections were computed, while in the second pass a validation of these candidates was done. The segmentation in this case was performed in three main steps:

1. *a symmetric Kullback-Leibler (KL2) distance* [28] was computed for each point in an audio stream in the following way: the KL2 distance for one point was computed from two adjacent fixed-length analysis windows surrounding that point; and such calculation was performed for every frame in an audio stream. Additionally, a frame-skip was introduced to speed up the calculation process.
2. *segment-boundary candidates* were produced by finding the peaks in the distance function: segment boundaries were selected at time locations, where KL2 values exceeds a pre-determined threshold chosen on a development data. Additional smoothing of the distance function and elimination of the smaller neighboring peaks within a certain minimum duration was applied to prevent over-generation of segment candidates at true boundaries.
3. *validation of segment-boundary candidates* were performed by using the Δ_{BIC} measure: a candidate point at time t_i was accepted as a segment boundary, if $\Delta_{BIC}^{X,Y}(t_i) > 0$. The $\Delta_{BIC}^{X,Y}(t_i)$ was computed on $X = x_{t_{i-1}}, ..., x_{t_i}$ and $Y = x_{t_i+1}, ..., x_{t_{i+1}}$ by using the formula, defined in (1), where t_{i-1} and t_{i+1} corresponded to the times of previous and next candidate, respectively.

This method tends to be less independent on the average segment size and can greatly reduce computational time of a segmentation process due to less frequent usage of a computationally expensive BIC measure.

The outputs of the audio-segmentation modules in both cases were acoustic-change detection points, which defined basic audio segments for further processing.

2.2. Speech Detection

Since the audio stream was already segmented into homogeneous regions of audio data based on acoustic changes, the speech-detection module had to distinguish which segments correspond to either speech or non-speech data.

A general approach, that was also followed in our speaker-diarization system, is a maximum-likelihood classification with Gaussian Mixture Models (GMMs), which are trained on manually labeled training data [34, 21, 9, 27, 12, 29]. The main issue in such classification is how to adequately represent speech and non-speech data.

We implemented three different representations:

1. a *standard acoustic representation* of audio signals: mel-frequency cepstral coefficients (MFCCs) were computed in the same manner as they are produced in the standard speech-recognition systems. Several GMMs were estimated to represent various acoustic speech and non-speech events (normal/telephone speech, music, silences, background noises, etc.).
2. *phoneme-recognition features*, based on consonant-vowel pairs (CV) obtained from simple phoneme speech recognizers: this representation is more suitable for speech/non-speech classification as shown in [39, 19]. Here, just two GMMs were trained, one model for speech and the other for non-speech data.
3. *fusion of acoustic and phoneme-recognition features*: both representations were joined in a fusion classification system also with just two GMMs, [39].

The standard MFCC-based representations are very common in systems, where a speech detection is used as a front-end for further processing of speech data, e.g. in speech- and speaker-recognition systems, where the same set of features can be used in later processing stages. However, a modeling of speech and non-speech data with just acoustic representations causes less robust performances of such systems, since several models have to be built to cover various acoustic phenomena, that are expected when processing audio data. To overcome these limitations we implemented a high-level representation of audio signals, based on phoneme-recognition features, which was first proposed in [39] and extensively tested on BN audio data in [19, 40].

The speech detection in all cases was performed by classifying each segment in an audio stream to speech or non-speech according to GMM that produced the highest likelihood from the given data, whereas in the phoneme-recognition and in the fusion case just two GMMs were used. The detected speech segments were subsequently passed to a speaker-clustering module, while the non-speech segments were discarded from any further processing.

2.3. Speaker Clustering

The purpose of this stage is to associate or cluster together segments from the same speaker. In ideal case, such clustering should be produced where all segments of each speaker are grouped in a single cluster.

The general method, that was also implemented in our system, is to perform agglomerative clustering using a bottom-up approach, [30], with the BIC measure as a merging criterion. Such clustering can be described in three main steps:

1. *initialization:*
   each segment C_i present one cluster;
   intial clustering is $\mathcal{C}_0 = \{C_i | i = 1, ..., N\}$
2. *merging procedure:*
   Repeat:
   - Among all possible pairs of clusters (C_r, C_s) in $\mathcal{C}_{t-1}$ find the one, say (C_i, C_j), such that
     $$\Delta_{BIC}(C_i, C_j) = \max \Delta_{BIC}(C_r, C_s) \quad (2)$$
   - Define $C_q = C_i \cup C_j$ and produce new clustering $\mathcal{C}_t = (\mathcal{C}_{t-1} - \{C_i, C_j\}) \cup \{C_q\}$
3. *stopping criterion:*
   The merging procedure is repeated until in $\mathcal{C}_t$ exists such pairs (C_r, C_s), for which
   $$\Delta_{BIC}(C_r, C_s) > 0.0. \quad (3)$$

In the *merging procedure* the joining of clusters was performed by searching for the maximum

BIC score among all the possible pair-wise combinations of clusters. The BIC measure was the same as the one used for the audio segmentation, that is defined in (1), but it needed to be reformulated in the following way: the Δ_{BIC} score was in the segmentation case computed from the Gaussians, estimated from the data X, Y, which presented the segment of data $X \cup Y$, divided at time t. In the clustering case, the data were constructed from the data of the current processing clusters C_r and C_s, i.e., $X = C_r$ and $Y = C_s$, where a dividing point at time t was obsolete. Therefore $\Delta_{BIC}(C_r, C_s)$ was defined as $\Delta_{BIC}(C_r, C_s) := \Delta_{BIC}^{C_r,C_s}(.)$ without the time t. The merging process was *stopped* when the highest BIC score was lower than a specified threshold, which was in our case set to 0.0.

The output of the speaker-clustering module produced relative segment labels (for example 'spk1'), which corresponded to speaker clusters.

At this stage several improvements can be made to increase the performance of the speaker diarization, like joint segmentation and clustering [18] and/or cluster re-combination [36], but in our speaker-tracking system we found no additional improvement in the performance when applying some of these methods. Note that this is also the final stage of the speaker-diarization process.

2.4. Speaker Identification

Since speaker-diarization systems only produce relative speaker labels, additional modules for speaker identification have to be included in the system, when the true identities of the speakers are needed. 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 our speaker-tracking system.

A speaker-identification component was adopted from a speaker-verification system that was originally designed for the detection of speakers in conversational telephone speech, [16]. The speaker-verification system was based on the standard Gaussian Mixture Model – Universal Background model (GMM-UBM) approach, [26].

The speaker models, which are needed in the speaker-identification component, were built in the *enrolment stage*, following the classical MAP adaptation approach [10, 26]. First, the UBM was trained from the pooled data of large number of different speakers, using the maximum likelihood criterion. After that, speech data from each target speaker was used to build the speaker models by adapting the means of all Gaussian components of the UBM, [26].

In the *speaker-identification stage* for each cluster C_i it had to be decided, to which speaker S_j, from the set of known speakers $\mathcal{S}$, the cluster C_i belongs. This decision was based on the maximum likelihood criterion:

$$S_j = \arg\max_{S_k \in \mathcal{S}} \log p(X_{C_i}|M_{S_k}) \qquad (4)$$

where X_{C_i} represents all the data (i.e., the set of acoustic feature vectors) from the cluster C_i and M_{S_k} is a GMM model of the hypothesized speaker S_k. To account for the possibility that the cluster C_i did not belong to any speaker from the set of known speakers, the final decision was made after comparing the likelihood of the winning model to the likelihood of the unknown-speaker model, which was represented by the UBM.

We additionally applied *feature warping* [23] and *t-norm* score normalization [2] techniques in all of our experiments to compensate for different channel effects. However, less efforts were taken to explicitly verify the effectiveness of these methods in the tested speaker-tracking system, since our research was more focused on measuring the impacts of speaker-diarization tasks to the speaker-identification process.

The results of this module were audio segments with the true speaker identification labels. Those segments that included the data which did not belong to any of the enrolled speakers got empty labels corresponding to 'unknown' speakers. These results presented the final outputs of our system. Audio streams equipped with such information can be further used for retrieving of speakers in various speaker-tracking applications.

3. Evaluation Experiments

Presented speaker-tracking system was evaluated on the SiBN database [37], which consisted of 33 hours of BN shows in Slovene. Twenty

hours were used for an estimation of all the open parameters in all the components of our diarization system, and the remaining 12 hours served for the assessment of the system's performance.

A tuning of the open parameters in audio-segmentation, speech-detection and speaker-clustering modules corresponded to optimizing the overall speaker-diarization performance on the training data.

In the audio-segmentation module, we had to tune open parameters of both implemented segmentation procedures. In the first — *standard BIC* — approach a threshold in the BIC measure and the parameters of the analysis windows had to be estimated. Setting the parameters of analysis windows included determining an initial window length, an extension parameter and a maximum window duration. An initial window was set to 2.0 s, an extension parameter to 1.0 s and a maximum window size was 10.0 s. A threshold was determined by setting the penalty factor λ of the Δ_{BIC} measure in (1). It was set so as to detect as many true change-detection points in the audio streams, while, at the same time, to preserve a low rate of miss-detected segment boundaries. The emphasis was put more on the detection of true segment boundaries, even if additional segment boundaries were falsely detected. As a result, over-segmented audio streams were produced, but they had almost no influence on the overall speaker-diarization results when using them as inputs for the speaker-clustering module. In the case of under-segmented audio data, it was found that they could severely degrade the speaker-diarization and tracking performance.

In the case of the *DISTBIC* approach, we had to set several open parameters for finding change-detection candidates in the first and second steps of the approach and, in the last step, a penalty factor λ had to be determined in the Δ_{BIC} measure. A length of the adjacent windows in the first step was fixed to 2.0 s, while the frame-skip for calculating the KL2 measure was set to 0.1 s. In the second step, additional parameters needed to be assigned to find appropriate local-maximum points for the segmentation-boundary candidates. The minimum duration between two consecutive local-maxima was set to 1.0 s, while the threshold, above which the local-maximum points were appointed as segment-boundary candidates, was adjusted based on the development data. A tuning of the λ in the last step was done in the same manner as in the previous approach.

In the speech-detection module, a classification framework was based on the GMMs and three different audio signal representations were tested. In all the approaches, a set of GMMs were trained on speech and non-speech data from the training part of the SiBN database.
In the first approach the GMMs were trained based on the acoustic features, implemented by the first 12 MFCCs and a short-term energy with their first derivatives. By using such features several different GMMs for detecting speech and non-speech were produced trained on: broadband and telephone speech, silences, music, and noisy background data.
In the second approach, we tested our proposed representation based on the phoneme-recognition features. In this approach, a parametrization was made from four phoneme features [39], derived from detected consonant-vowels (CV) pairs. Therefore, we had to implement a simple phoneme recognizer to detect CV from audio signals. The recognizer was built in a standard way, using HMMs trained on Slovenian data. For this approach only two GMMs were trained, one for each class.
In the third approach we joined both representations in a fusion system. The fusion was achieved by using a state synchronous two-stream GMMs, [24].

In the bottom-up clustering approach in the speaker-clustering module just a penalty factor λ in the Δ_{BIC} measure from (2) was tuned. The same λ was also used in (3), which defined a threshold for stopping a merging process. The λ was set to optimize the overall speaker-tracking performance on the training data. Here, the same phenomena were explored as in the audio-segmentation module. By setting a proper threshold we could optimize the speaker-diarization performance on the training data, but it was found that this did not necessarily reflect in the overall best performance of the speaker-tracking system. The optimal performance was achieved in the case when the 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 clusters were produced containing speech from different speakers, which degraded the speaker-detection performance. Therefore, the λ was

chosen to optimize a speaker-tracking performance, rather than optimizing a speaker diarization.

In a speaker-identification module the true detection of speakers was carried out. Therefore, the GMM of each target speaker had to be provided. They were built from UBM, which were trained on the speech data of the training part from the SiBN database. All the models were constituted from 1024 Gaussian mixtures, which were estimated using the Baum-Welch Expectation-Maximization (EM) algorithm. The GMM model for each target speaker was derived from the UBM using the MAP adaptation technique in a standard way, [26]. The evaluated system was capable of detecting 41 target speakers extracted from the training data in the enrollment phase. In the test phase, the data from each cluster were compared against all of the models from the target-speakers repository by using criterion defined in (4). Note that no additional score threshold was proposed, since we tried to evaluate the system across the whole range of all possible operating points.

3.1. Evaluation Results

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

The evaluated speaker-tracking system was capable of detecting 41 target speakers from the audio data, which included 551 different speakers. The performance of the evaluated system was assessed by including all target speakers with the addition of non-speech segments.

Three groups of experiments were conducted. In each group the impact of one component of the system was explored by comparing the final speaker-tracking performance. The overall speaker-tracking results were produced in terms of the false-acceptance (FA) and false-rejection (FR) rates computed at different operating points and presented in the form of Detection Error Trade-off (DET) curves, [16]. This evaluation measure is commonly used for the assessment of speaker-recognition systems [25], where FA and FR rates are computed on the basis of speaker's utterances. In the case of speaker tracking in continuous audio streams, the FA and FR rates have to be derived on the audio segments. Thus, a generalization of the DET measure have to be applied, which computes the FA and FR rates at the time (frame) level.

Figures 2, 3 and 4 present speaker-tracking results from the evaluated system, where different versions of the system's components were combined. The speaker-identification module was the same for all the evaluations, while in the components for audio-segmentation, speech-detection and speaker-clustering different approaches were applied. The experiments were planned so, that the evaluation of each component took place in such order to follow the processing stages of a general speaker-tracking system. In this case the best approach of the current evaluated component could be used in the experiments of the next evaluated component.

In Figures 2, 3 and 4, a speech-detection module is marked as *SNS*, an audio segmentation is referred to as *S* and a speaker clustering to as *C*. Since the speaker-identification procedure was always the same, no legend names for that module are provided. In addition to that, the FA and FR rates in all figures correspond to false alarm probabilities and miss probabilities, respectively.

In the first evaluation experiments in Figure 2 the impact of the audio segmentation module to the overall speaker-tracking performance was assessed. Two different segmentation approaches were tested and compared with a manual segmentation. The experiments were conducted in a way to measure just the impact of the audio segmentation. This was achieved by applying the same procedures in all other system's components. For the speech detection a standard classification procedure with GMMs and CV features was applied (marked as *SNS:GMM+CVS* in Figure 2), which was presented in Section 2.2, while no speaker-clustering was performed (marked as *C:w/o* in Figure 2). We tested three different audio segmentation procedures. In the first case the segmentation was performed manually, while in the second and the third cases the automatic segmentation procedures described in Section 2.1 were applied. One was a segmentation procedure by using a standard one-pass approach with

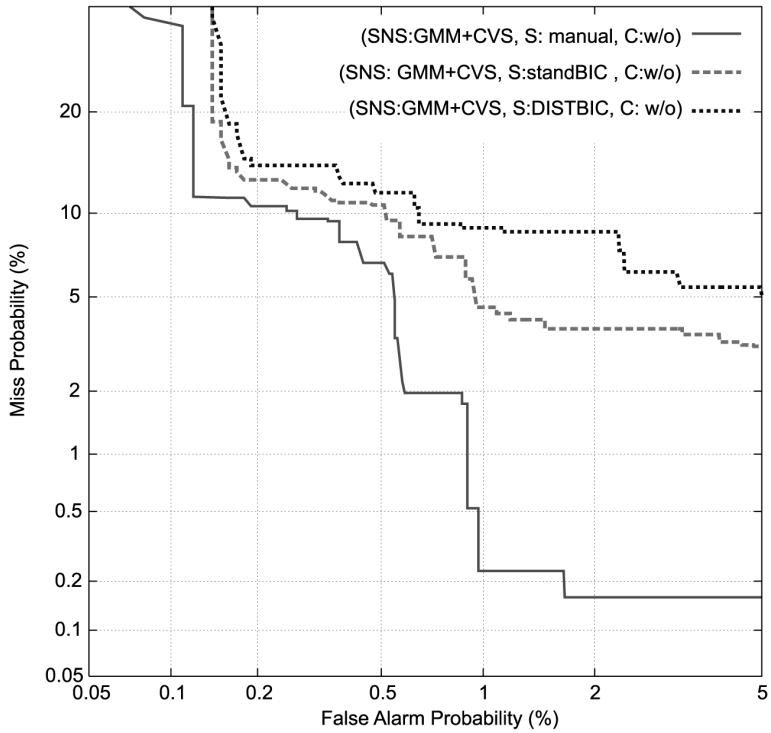

Figure 2. The overall speaker-tracking results of the evaluated system, where different audio-segmentation procedures were applied. Lower DET values correspond to better performance.

BIC measure (referred to *S:standBIC* in Figure 2). The other was the DISTBIC approach, which is marked as *S:DISTBIC* in Figure 2.

As can be seen from the results in Figure 2, the manual segmentation outperforms the automatic versions by more than 3% across the whole range of operating points. Since the segmentation procedure is usually applied in the first steps of speaker-tracking systems, the errors from the segmentation have an impact on all subsequent procedures. In our case, the errors in detecting change points in continuous audio streams produced non-homogeneous segments, which caused the unreliable detection of speech/non-speech regions and the unreliable detection of target speakers as well. Accordingly, both types of errors were, therefore, additionally integrated into the overall results of the evaluated systems.

The same phenomenon can be observed by inspecting both automatic versions of segmentation procedures. During the evaluation phases we also tested their performances in a segmentation task alone. The segmentation results obtained by using the *F-measure* [14] spoke in favor of the *standBIC* approach, where a segmentation accuracy of 74% was achieved, in comparison to *DISTBIC* approach, where a segmentation accuracy was 70%. Nearly the same difference can be observed in the overall speaker-tracking results in Figure 2. This confirms our previous observations that a good segmentation could greatly reduce the overall speaker-tracking error rates.

Another evaluation perspective present the results in Figure 3. Here, different speech-detection procedures were tested in a system where audio-segmentation and speaker-clustering modules stayed the same through all the evaluation experiments. We explored the impacts of three speech/non-speech segmentation approaches and compared them to the manually labeled segments. The manual speech/non-speech segmentation is referred to as the *SNS:manual* approach in Figure 3. A legend name *SNS:GMM+MFCC* presents the approach where MFCC features were used for the representation of audio signals. A legend name *SNS:GMM+CVS* refers to an approach, where CV features were applied for speech detection, while a legend name *SNS:GMM+fusion* corresponds to a fusion of both representations.

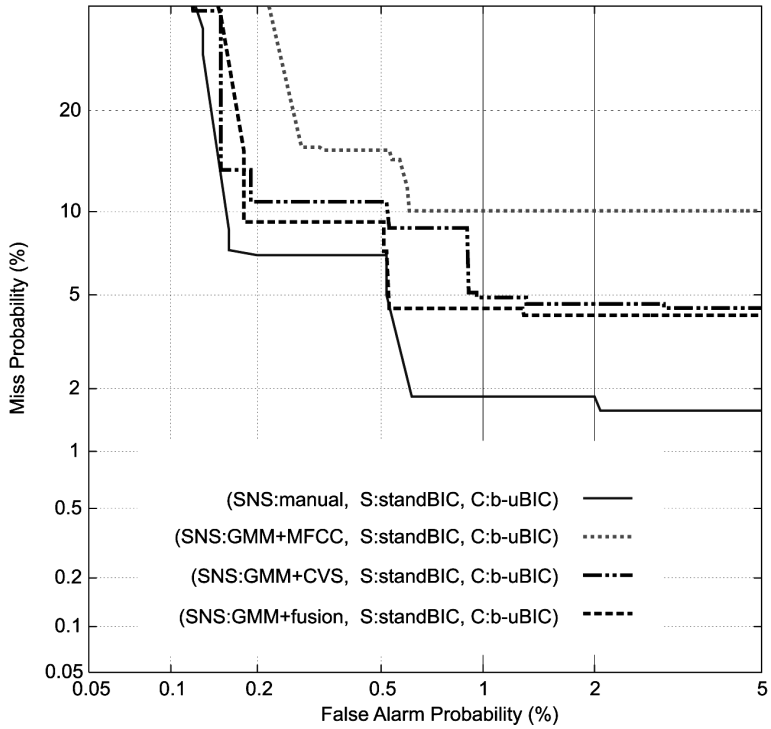

Figure 3. The overall speaker-tracking results of the evaluated system, where different speech-detection procedures were applied. Lower DET values correspond to better performance.

Here, the audio segmentation module was implemented by the *standBIC* approach (legend name *S:standBIC*), since it performed best among all tested segmentation methods from the previous evaluation. For speaker clustering we implemented an approach from Section 2.3, that is marked as *C:b-uBIC* in Figure 3).

By comparing the evaluation results of all tested systems we can draw the same conclusions as in the audio-segmentation case. The better speech/non-speech-segmentation procedure works, the lower speaker-tracking error is achieved. The best DET results correspond to manual speech detection, next are methods where phoneme-recognition features were introduced, and the last is a standard approach where just acoustic (MFCC) features were used. The same order of the performance was obtained by comparing these procedures in the speech/non-speech-segmentation task alone [19, 39]. Another important issue reveals these results, which was also observed in [40]. 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, speech detection errors directly influence the speaker-tracking results. On the other hand, an erroneous speech/non-speech classification of audio segments in the speaker-tracking system has an influence on the speaker-clustering and identification performance. Therefore, good speech detection in continuous audio streams is a necessary pre-processing step if we want to achieve good speaker-diarization and tracking results.

At the final evaluation, the impact of speaker clustering was explored in two experiments. While in audio-segmentation and speech-detection modules the best approaches from the previous evaluations were applied, we tested two speaker-tracking systems: one with and another without speaker clustering. In the first case a standard bottom-up speaker clustering, described in Section 2.3, was implemented. In Figure 4, this approach has a legend name *C:b-uBIC*, while a system where no clustering was implemented has a legend name *C:w/o*.

This evaluation aims to examine whether or not it is better to use a speaker clustering procedure in speaker-tracking systems. As can be seen from Figure 4, there is not so much difference in the performances of the systems, where clustering was applied, compared to those without

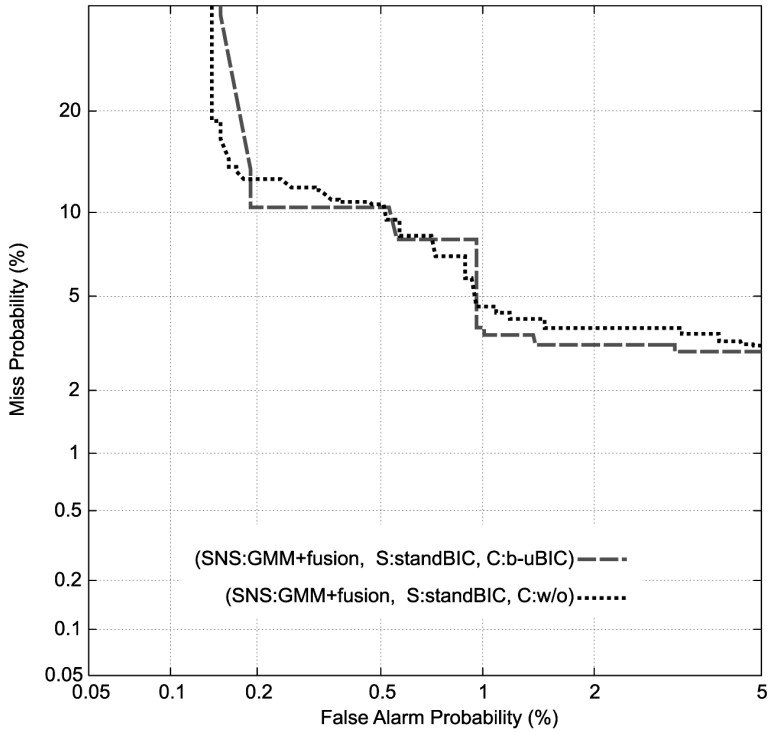

Figure 4. The overall speaker-tracking results of the evaluated system, where a speaker-clustering procedure was tested. Lower DET values correspond to better performance.

clustering. Our tracking results with automatic clustering 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 a study of speaker tracking for radio broadcast news in [20], where it was concluded that speaker identification could help to improve the speaker-clustering performance, and not vice versa.

4. Conclusion

A system for speaker tracking in BN audio data was presented. We gave an overview of the four main building blocks of such a system and provided an extensive evaluation of the impacts of each of the system's components to the overall speaker-tracking performance. We implemented different approaches of audio segmentation, speech detection and speaker clustering, and measured their impacts to the overall speaker-tracking results. The comparison of the evaluation results of different versions of the speaker-tracking system provides valuable insights into how the system works and which components of the system have greater impact on the overall performance. It was found that the most critical component of the system is the audio-segmentation module. If the segmentation procedure produces too many non-homogeneous segments due to improperly detected change points in an audio stream, this causes unreliable performance of the speech-detection and the speaker-identification modules, and thus degrades the overall performance of the system. The same can be concluded for the speech-detection module, where an erroneous speech/non-speech classification caused explicit errors in the evaluated speaker-tracking system and had further influence on the speaker-clustering and identification performance. As far as the speaker clustering is concerned, it was shown that we could not gain any improvement in the overall performance of the system when using clustering or not.

Even though our system was built by implementing the most recently published speaker-diarization methods and the evaluation results demonstrated an acceptable speaker-tracking performance, we believe that there is still a room for an improvement. In our future work we will

try to increase the robustness of the critical system's components in the same manner as we did in the speech-detection module, by introducing phoneme-recognition features for representations of audio signals, which proved to be more discriminative and less sensitive to different training and unseen conditions.

References

[1] T. ANASTASAKOS, ET AL., A Compact Model for Speaker-adaptive Training. Presented in the *Proceedings of International Conference on Spoken Language Processing (ICSLP1996)*, (1996) Philadelphia, PA, USA.

[2] R. AUCKENTHALER, ET AL., Score normalization for text-independent speaker verification system. *Digital Signal Processing*, 10 (2000), pp. 42–54.

[3] C. BARRAS, ET AL., Multistage Speaker Diarization of Broadcast News. *IEEE Transactions on Speech, Audio and Language Processing, Special Issue on Rich Transcription*, 14 (2006)), pp. 1505–1512.

[4] P. BEYERLEIN, ET AL., Large vocabulary continuous speech recognition of Broadcast News. The Philips/RWTH approach. *Speech Communications*, 37 (2002), pp. 109–131.

[5] S. S. CHEN, P. S. GOPALAKRISHNAN, Speaker, environment and channel change detection and clustering via the Bayesian information criterion. Presented in the *Proceedings of the DARPA Speech Recognition Workshop*, (1998) Lansdowne, Virginia, USA.

[6] P. DELACOURT, ET AL., A Speaker Tracking System Based on Speaker Turn Detection for NIST Evaluation. Presented in the *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP2000)*, (2000) Istanbul, Turkey.

[7] P. DELACOURT, C. J. WELLEKENS, DISTBIC: A speaker-based segmentation for audio data indexing. *Speech Communication*, 32 (2000), pp. 111–126.

[8] J. FISCUS, ET AL., Results of the Fall 2004 STT and MDE Evaluation. Presented in the *Proceeedings of the Fall 2004 Rich Transcription Workshop*, (2004) Palisades, NY, USA.

[9] J.-L. GAUVAIN, ET AL., Partitioning and Transcription of Broadcast News Data. Presented in the *Proceeedings of the International Conference on Spoken Language Processing (ICSLP98)*, (1998) Sydney, Australia, pp. 1335–1338.

[10] J. L. GAUVAIN C.-H. LEE, Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. Speech, and Audio Processing*, 2 (1994)), pp. 291–298.

[11] J. L. GAUVAIN, ET AL., The LIMSI Broadcast News transcription system. *Speech Communications*, 37 (2002)), pp. 89–108.

[12] T. HAIN, ET AL., Segment Generation and Clustering in the HTK Broadcast News Transcription System. Presented in the *Proceeedings of the 1998 DARPA Broadcast News Transcription System*, (1998) Lansdowne, VA, USA.

[13] D. ISTRATE, ET AL., Broadcast News Speaker Tracking for ESTER 2005 Campaign. Presented in the *Proceedings of Interspeech 2005 — Eurospeech*, (2005) Lisbon, Portugal.

[14] T. KEMP, ET AL., Strategies for Automatic Segmentation of Audio Data. Presented in the *Proceedings of the IEEE International Conference on Acoustic Signal and Speech Processing, ICASSP 2000*, (2000) Istanbul, Turkey.

[15] J. MAKHOUL, ET AL., Speech and language technologies for audio indexing and retrieval. *Proceedings of the IEEE*, 88 (2000), pp. 1338–1353.

[16] A. MARTIN, ET AL., The NIST speaker recognition evaluation — overview, methodology, systems, results, perspectives. *Speech Communications*, 31 (2000), pp. 225–254.

[17] S. MATSOUKAS, ET AL., Practical Implementations of Speaker-adaptive Training. Presented in the *Proceedings of the 1997 DARPA Speech Recognition Workshop*, (1997) Chantilly VA, USA.

[18] S. MEIGNIER, ET AL., Evolutive HMM for Multispeaker Tracking System. Presented in the *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2000)*, (2000) Istanbul, Turkey.

[19] F. MIHELIČ, J. ŽIBERT, Robust speech detection based on phoneme recognition features. Presented in the *Proceedings of Text, Speech and Dialogue (TSD 2006)*, (2006) Brno, Czech Republic.

[20] D. MORARU, ET AL., Experiments on speaker tracking and segmentation in radio broadcast news. Presented in the *Proceedings of Interspeech 2005 — Eurospeech*, (2005) Lisbon, Portugal.

[21] P. NGUYEN, ET AL., Rich Transcription 2002 Site Report. Panasonic Speech Technology Laboratory (PSTL). Presented in the *Proceeedings of the 2002 Rich Transcription Workshop*, (2002) Vienna, VA, USA.

[22] B.NEDIC, ET AL., The Elisa'99 Speaker Recognition and Tracking Systems. Presented in the *Proceedings of IEEE Workshop on Automatic Advanced Technologies*, (1999).

[23] J. PELECANOS, S. SRIDHARAN, Feature warping for robust speaker verification. Presented in the *Proceedings of the Speaker Odyssey Workshop*, (2001) Crete, Greece.

[24] G. POTAMIANOS, ET AL., Audio-visual Automatic Speech Recognition: An Overview. *Issues in Visual and Audio-visual Speech Processing*. G. Bailly, E. Vatikiotis-Bateson, P. Perrier (Eds.), MIT Press, 2004.

[25] M. A. PRZYBOCKI, A. F. MARTIN, NIST Speaker Recognition Evaluation Chronicles. Presented in the *Proceedings of the Speaker Odyssey Workshop 2004*, (2004) Toledo, Spain.

[26] D. A. REYNOLDS, ET AL., Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10 (2000), pp. 19–41.

[27] D. A. REYNOLDS, P. TORRES-CARRASQUILLO, The MIT Lincoln Laboratory RT-04F Diarization Systems: Applications to Broadcast Audio and Telephone Conversations. Presented in the *Proceeedings of the Fall 2004 Rich Transcription Workshop*, (2004) Palisades, NY, USA.

[28] M. SIEGLER, ET AL., Segmentation, Classification and Clustering of Broadcast News Data. Presented in the *Proceedings of the DARPA Speech Recognition Workshop*, (1997) Chantilly, VA, USA.

[29] R. SINHA, ET AL., The Cambridge University March 2005 speaker diarisation system. Presented in the *Proceedings of Interspeech 2005 — Eurospeech*, (2005) Lisbon, Portugal.

[30] S. THEODORIDIS, K. KOUTROUMBAS, *Pattern Recognition (2nd edition)*. Academic Press, 2003.

[31] S. TRANTER, D. REYNOLDS, An Overview of Automatic Speaker Diarisation Systems. *IEEE Transactions on Speech, Audio and Language Processing, Special Issue on Rich Transcription*, 14 (2006)), pp. 1557–1565.

[32] A. TRITSCHLER, R. GOPINATH, Improved speaker segmentation and segments clustering using the Bayesian information criterion. Presented in the *Proceedings of Eurospeech 99*, (1999) Budapest, Hungary.

[33] P. C. WOODLAND , The development of the HTK Broadcast News transcription system: An overview. *Speech Communications*, 37 (2002), pp. 47–67.

[34] C. WOOTERS, ET AL., Towards Robust Speaker Segmentation: The ICSI-SRI Fall 2004 Diarization System. Presented in the *Proceeedings of the Fall 2004 Rich Transcription Workshop*, (2001) Palisades, NY, USA.

[35] B. ZHOU, J. HANSEN, Unsupervised Audio Stream Segmentation and Clustering via the Bayesian Information Criterion. Presented in the *Proceedings of International Conference on Spoken Language Processing (ICSLP 2000)*, (2000) Beijing, China.

[36] X. ZHU, ET AL., Combining Speaker Identification and BIC for Speaker Diarization. Presented in the *Proceedings of Interspeech 2005 — Eurospeech*, (2005) Lisbon, Portugal.

[37] J. ŽIBERT, F. MIHELIČ, Development of Slovenian Broadcast News Speech Database. Presented in the *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2004)*, (2004) Lisbon, Portugal.

[38] J. ŽIBERT, ET AL., The COST278 Broadcast News Segmentation and Speaker Clustering Evaluation — Overview, Methodology, Systems, Results. Presented in the *Proceedings of Interspeech 2005 — Eurospeech*, (2005) Lisbon, Portugal.

[39] J. ŽIBERT, ET AL., Speech/Non-Speech Segmentation Based on Phoneme Recognition Features. *EURASIP Journal on Applied Signal Processing*, 6 (2006)), 1–13.

[40] J. ŽIBERT, ET AL., 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, 23–48.

Received: June, 2007
Revised: February, 2008
Accepted: February, 2008

Contact address:

Janez Žibert
Faculty of Electrical Engineering
University of Ljubljana
Tržaška 25, SI-1000 Ljubljana, Slovenia
e-mail: janez.zibert@fe.uni-lj.si

JANEZ ŽIBERT was born in 1974. He received the B.Sc. degree in mathematics in 1998 from the Faculty of Mathematics and Physics and the M.Sc. and the Ph.D. degrees in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana in 2001 and 2006, respectively. He is currently working as a research associate at the Laboratory of Artificial Perception, Systems and Cybernetics, at the Faculty of Electrical Engineering, University of Ljubljana. His research interests include audio signal processing, automatic speech and speaker recognition and audio information retrieval. Janez Žibert is a member of the International Speech Communication Association and a member of the Slovenian Pattern Recognition Society and of the Slovenian Language Technologies Society.

BOŠTJAN VESNICER was born in 1976. He received the B.Sc. and M.Sc. degrees in electrical engineering in 2000 and 2003, respectively, both from the University of Ljubljana, Slovenia. Until recently he worked on statistical speech synthesis at the Laboratory of Artificial Perception, Systems and Cybernetics, Department of Electrical Engineering, University of Ljubljana, where he holds a research position. Currently he is preparing a Ph.D. thesis on channel compensation techniques for speaker recognition. Boštjan Vesnicer is a member of the Slovenian Pattern Recognition Society and of the Slovenian Language Technologies Society.

FRANCE MIHELIČ was born in 1952. He studied at the Faculty of Natural Sciences, Faculty of Economics and Faculty of Electrical Engineering all at the University of Ljubljana. There he received the B.Sc. degree in technical mathematics, the M.Sc. degree in operational research and the Ph.D. degree in electrotechnical sciences in 1976, 1979 and 1991, respectively. Since 1978 he has been a staff member at the Faculty of Electrical Engineering in Ljubljana, where he is currently associate professor. His research interests include pattern recognition, speech recognition and understanding, speech synthesis and signal processing. He has authored and co-authored several papers and 2 books addressing several aspects of the above areas. France Mihelič, Ph.D., is a member of IEEE, International Speech Communication Association, the Slovenian Mathematician's, Physicist's and Astronomer's Society, Slovenian Pattern Recognition Society and the Slovenian Language Technologies Society.