Multimedia Information Access Using Multiple Speaker Classifiers

Mahesh Viswanathan, Fereydoun Maali,[*] Homayoon S.M. Beigi,[†] and Alain Tritschler
IBM T.J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598, USA
maheshv@watson.ibm.com, maali-Sigrec@worldnet.att.net
beigi@internetserver.com, alain.tritschler@voila.fr

Abstract

There have been several new systems for multimedia information access reported in recent years. The system presented here shares many of their aspects, but it differs in a significant way from them; it extends the realm of multimedia access to include speaker-based information. We have already prototyped and reported such a system elsewhere whose main features include SVAPI-based speaker recognition combined with speech recognition for joint text- and speaker-based retrieval from audio and video. A vital component of such a system is speaker identification whose performance degrades for utterances smaller than eight seconds to such an extent that such segments have to be dismissed with a catch-all, neutral label. Here, we use a Bayesian Information Criterion based speaker clustering technique to analyze the same audio data. The results of this classifier are combined with those from our SVAPI-based speaker classifier using a decision integration scheme to produce new labels for many such short speaker segments. We discuss the details of this combined analysis and its results. We additionally report on a on-the-fly speaker enrollment scheme using this BIC-based speaker clustering technique.

Introduction

Multimedia information access of live audio and video information is greatly enhanced when retrieval permits textual, image-based and speaker-based queries. We have prototyped and reported a system to process audio derived from a video stream, such as broadcast news, to produce text automatically via speech-to-text transcription and to identify the speakers via speaker recognition (Viswanathan et al., 1999).

Short speaker segments be it genuinely short or as a result of over-segmentation degrade speaker identification performance since these segments are currently dismissed as un-identifiable. In our system previously reported, we label such segments as “Inconclusive” without any attempt to identify the speaker in that segment. Here, to remedy this shortcoming, we use multiple classifiers. Integrating decisions due to two classifiers yields a speaker label for many short speaker segments that would otherwise be dismissed as “Inconclusive”. The focus of this paper is on the decision integration scheme that we have adopted to alleviate the short utterance speaker labeling problem. This will be presented in the context of our overall system. An additional area that we address in this paper is on-the-fly speaker enrollment which greatly facilitates the training of new speakers for the speaker data store.

[*] Signal Recognition Corporation, P.O. Box 7010, New York, NY 10128, USA

[†] On leave during 2000

The indexing system consists of a real-time, on-line audio analysis phase, followed by an off-line indexing phase. The on-line phase consists of: automatic speech transcription obtained using the IBM ViaVoice Broadcast News engine; a speaker segmentation engine that uses a Bayesian Information Criterion (BIC) for acoustic change detection; a text-independent, language-independent, speaker identification engine which is SVAPI-compliant; and an additional speaker classifier – an extension of the segmentation engine – which clusters the BIC segments and is used to reinforce or handicap the SVAPI-classifier's decision. All three engines run concurrently to produce their respective outputs in real-time on a 400 MHz IBM-compatible PC.

The off-line indexing is automatically triggered after the first phase ends to generate an index in two stages: one, for text-based retrieval, consisting of statistics extraction for Okapi-based retrieval incorporating chunking of the transcript into manageable "documents", tokenization, part-of-speech tagging for morphological analysis (or intelligent stemming), followed by index building. And two, for speaker-based retrieval, consisting of score-sorted speaker segments, with each segment being associated the audio file source identifier, start and end times of each segment, assigned label, and match score.

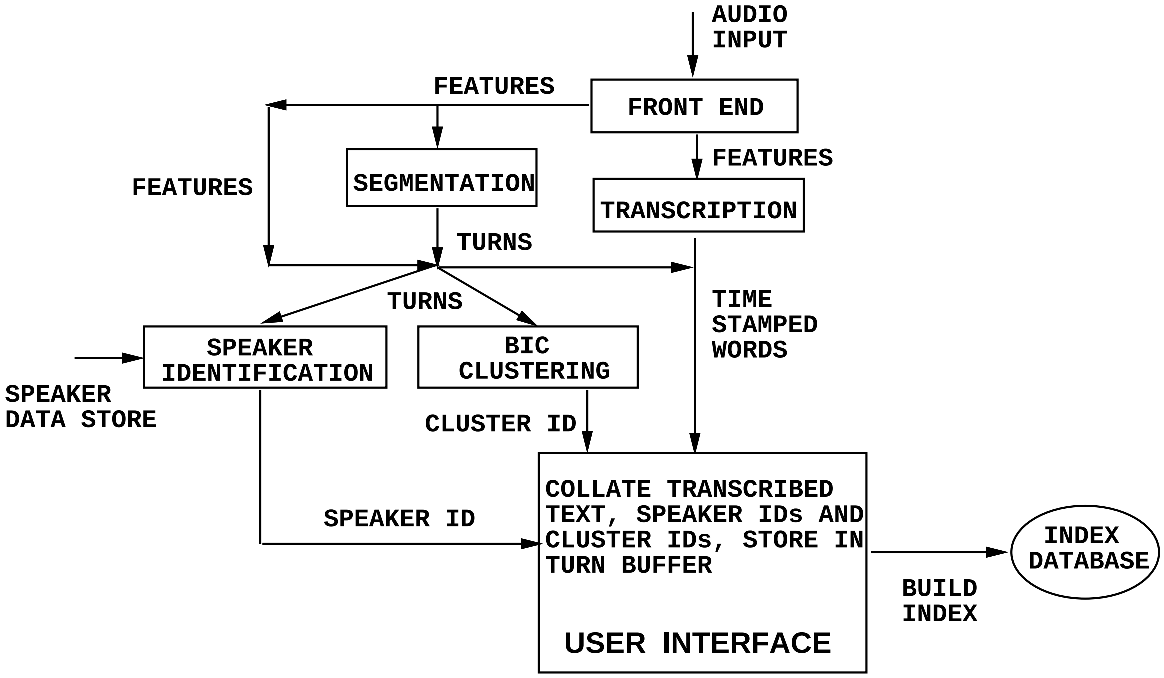


Figure 1: The architecture of a system for real-time speech recognition, speaker segmentation, speaker clustering (BIC), and speaker identification (SVAPI) for multimedia indexing and retrieval. This architecture is realized in an application which analyzes the input audio using three engines in real-time on a 400MHz Pentium II PC. Indexing is triggered automatically when the audio terminates. Note, that the data store shown is generated off-line.

Audio from a live TV broadcast or equivalent audio source is the input to this system. The system uses a common front-end signal processing module which converts the input audio into mel-cepstral feature vectors. These multi-dimensional feature vectors are simultaneously delivered to the engines above in a multi-process and multi-threaded programming environment. The three engines are all programmable via APIs called SMAPI, SEGAPI, and SVAPI. At the conclusion of the audio broadcast, the indexing API is invoked automatically to generate the index files (30 secs/audio hour) (Figure 1).

Hirschberg discusses the issues relating to building user interfaces for audio browsing and retrieval systems (Hirschberg et al., 1999). Satoh describes a system called “Name-It” that uses a combination of face recognition and close-captioning information from a video sequence (Satoh et al., 1999). Srinivasan describes “CueVideo” which video key frames detection and speech recognition for audio/video browsing and indexing (Srinivasan et al., 1999). A search and retrieval scheme for audio with emphasis on using acoustic and perceptual features for short and single-gestalt sounds is discussed in (Wold et al., 1996). All of these approaches use various facets of video and audio processing to arrive at a solution to the vexing problem of indexing multimedia content.

The next section is a very brief review of speech recognition as it applies to our audio analysis system for multimedia indexing. This is followed by a discussion of BIC-based speaker segmentation. A description of speaker identification follows. A second classifier using BIC once again for clustering speakers is discussed next followed by a section on how the two speaker classifiers can be combined. We review the results and close with a summary.

Overview of the Multimedia Information Access System Components

Speech Transcription

The IBM ViaVoice Broadcast News engine is used for transcribing the frames delivered by the front-end to recognized text. This engine uses a vocabulary of about 60,000 words; an acoustic model trained with 70 hours of broadcast news data; and a language model built using the transcripts for the aforementioned 70 hours plus a corpus of 400 million words of broadcast news text. The output of this module is a succession of time-stamped words. Table 1 presents the transcription performance on a standard two-hour broadcast news evaluation test.

Speech Conditions	WER (%)
Prepared Speech	22.3
Spontaneous Speech	29.6
Low fidelity Speech	39.6
Speech+Music	37.5
Speech+Background noise	35.1
Non-native speakers	29.7
Overall	29.7

Table 1: Transcription performance as measured by word error rate (WER) for IBM’s 1997 real-time system for broadcast news.

BIC Segmentation

The BIC-based segmentation engine uses the Bayesian Information Criterion to partition the frames produced by the front-end (Akaike, 1974; Chen & Gopalakrishnan, 1998; Delacourt et al., 1999); The basic problem may be viewed as a two-class classification problem where the object is to determine whether N consecutive audio frames constitute a single homogeneous window of frames W or two such windows: W_1 and W_2 with the boundary frame (or turn) occurring at the ith frame. In order to detect whether a speaker change occurred within a window of N frames, two models are built. One represents the entire window by a Gaussian characterized by $\{\mu, \Sigma\}$; and a second that represents the window up to the ith frame, W_1 with $\{\mu_1, \Sigma_1\}$ and the remaining part, W_2, with a second Gaussian $\{\mu_2, \Sigma_2\}$. This formulation assumes independent feature vectors but not uncorrelated feature elements. The details of this classifier may be formulated as:

$$\Delta BIC(i) = -\frac{N}{2}log|\Sigma| + \frac{N_1}{2}log|\Sigma_1| + \frac{N_2}{2}log|\Sigma_2| + \frac{\lambda}{2}(d + \frac{d(d+1)}{2})logN$$

where d is the dimension of the cepstral vector; $N_1 = i$ is the number of frames in W_1, $N_2 = (N - i)$ is the number of frames of the second part; and lambda (= 1.3) is a penalty function. $\Delta BIC < 0$ implies, taking the penalty into account, the model splitting the window into two Gaussians is more likely than the model representing the entire window with only a single Gaussian. The minimizer of all the ΔBIC's within the window is considered the frame where the acoustic change occurred.

Speaker Recognition

The speaker recognition module receives the frames from the front-end directly while obtaining the turns information from the segmentation module (Figure 1). The IBM speaker recognition engine is text-independent and language-independent and is SVAPI-compliant (Beigi et al., 1998).

Speaker identification calls for a database of pre-existing voice-print models and names (later returned as labels when identified) of the speakers of interest. At run-time, the first eight seconds (that is all is needed by the engine to make a positive determination of the identity of the speaker and segments shorter than eight seconds are labeled “Inconclusive”) of the segments delineated by the segmentation process are submitted for identification and subsequent verification. Identification involves finding the closest match to the run-time data from the enrolled speakers in the database. Verification confirms or rejects the identification result by comparing the run-time segment against the speaker's enrolled model and a set of cohort speaker models. These cohort models form part of a separate verification binary tree that is built when a speaker sample is submitted for enrollment. Each identification label derived from comparison against the speaker enrollment database includes a match score. Also returned are the next $(\mathrm{K} - 1)$ closest matches to the test segment.

Each enrolled speaker is modeled by a set of multi-dimensional Gaussian distributions for which the number of distributions, mean vectors, covariance matrices and priors are retained in the data store. Let $\{M_i \mid i = 1..I\}$ denote the models pertaining to each of the enrollees. Each model M_i can have N_j distributions associated with it. Let ω_{ij} refer to the jth distribution of model i. Also, let $\{\vec{x}_t \mid t = 1..T\}$ denote the frames representing the test utterance, **z**, whose label is sought. During run-time, a test utterance **z** is identified with model i according to:

$$\text{assign } \mathbf{z} \longrightarrow M_q \;\; iff \;\; D_q = \min_{i=1..I} \; [D_i], \text{ where}$$

$$D_i = \sum_{t=1}^{T} d(i,t), \;\; i = 1..I, \text{ and}$$

$$d(i,t) = -log[\sum_{j=1}^{N_j} P(\omega_{ij}) \; p(\vec{x}_t|\omega_{ij})],$$

with $P(\omega_{ij})$ being the prior of the jth distribution of model i, and $p(\vec{x}_t|\omega_{ij})$ being the conditional pdf of the tth frame of the test utterance conditioned on the jth component of model i. A Normal representation for $p(\vec{x}_t|\omega_{ij})$ is used.

Recognition comprises two stages. (1) Identification, and (2) Verification. First, in the class assignment stage, the test utterance is assigned one of the prototype classes. This stage produces an ID for the speaker. Next, in a verification stage the resultant class assignment (ID) from the first stage is subjected to a verification test. During verification the claimed speaker ID is confirmed using a second pass over the same data [7].

Although the first stage of the identification process is inherently a closed-set, i.e., the only possible labels are those in the database of enrolled speakers), the subsequent verification stage transforms it into an open-set, as unverified speaker labels can be rejected. The combined performance of the speaker segmentation and identification components of our system are shown in Table 2.

Speaker segments	104
Segments reported	84/104 (80.8%)
Segments missed	25/104 (23%)
Oversegmentations	5/104 (4.8%)
Identified	70/75 (93.3%)
Mis-identified	5/75 (6.7%)
Inconclusive	9/84 (10.7%)
Verified (from identified)	70/70
Mis-verified (from identified)	0/70
Verified (from mis-identified)	4/5
Mis-verified (from mis-identified)	1/5
Overall Verification	74/75 (98.7%)

Table 2: Speaker segmentation, identification, and verification results on a single broadcast news audio file with multiple speakers. Segments smaller than eight seconds are assigned an "Inconclusive" label. 75 segments were submitted for identification. 70 were identified and verified. Of the five mis-identifications, four were upheld, and one was erroneously rejected.

Indexing and Retrieval

A detailed account of the information presented in this section is in (Viswanathan et al., 2000). The recognizer generates words along with time-alignments for each word (the start time of each word relative to the start of the audio or video clip) which are collected into "documents". For each of these "documents" statistics required by the Okapi equation are gathered and recorded in the index files along with the media source file name. The time involved in generating the various index files is around 1–2% of the time required in transcription.

The index file for speaker-based retrieval is built from the combined results of speaker identification and BIC clustering (to be described). Each classification result is accompanied by a score which is the distance from the original enrolled speaker model to the audio test segment, start and end times of the segment relative to the beginning of the audio clip concerned, label (name of the speaker supplied during enrollment), and media source file name. The speaker index is a compilation of the components of the classification result arranged in a speaker-by-speaker basis. For each speaker record, the individual segments are stored in a score-sorted fashion from the best match (between any test segment and that speaker's voice-print in the data store) in descending order.

The retrieval engine can process text-based and speaker-based queries either individually or together. For a text-only query, the top N documents are retrieved from the text index and displayed via the user interface which includes the means to play the corresponding video or audio clip. Speaker-only queries are handled in the same manner with the retrieved portions being the transcribed text corresponding to the best matched speaker segments (top N from best to Nth best), along with the video or audio clips.

When a text-and-speaker query is specified, the best matched segments are those that contain the relevant subject material and are spoken by the speaker desired. The candidate documents are first gathered based on the text part of the query. The start and end times of each of these documents are compared against the start and end times of all the segments for the user-specified speaker. (This latter information is available in the speaker index.) All overlapping portions between the document segments and the speaker segments satisfy the user query. These are collected, sorted, normalized and the top N are presented to the user, as transcripts along with access to the corresponding video or audio clips.

We ran experiments for the entire system using five hours of broadcast news video data and 43 enrolled speakers. A 30-minute video segment was used in testing. Combined test-speaker retrieval accuracy measured as the ratio of number of relevant documents among all the retrieved documents over all queries was 84%. When available, the top 10 documents were retrieved for each query. Retrieved documents for queries like ”defense secretary” are considered relevant only if they contain both words. In the combined search, errors were both due to speaker mis-classifications and irrelevant documents being retrieved. Sample queries include ”land mines/Patrick Leahy” and ”plane crashes/Natalie Allen”).

BIC Clustering

The BIC-based speaker segmentation engine, described above, has also a clustering capability that we exploit for both on-the-fly speaker enrollment and decision integration with SVAPI classifier. As the BIC segmentation process produces new speaker segments, the BIC clustering process assigns newly produced segments to clusters (Tritschler & Gopinath, 1999). These clusters are generated automatically and is completely data driven. Our thesis is that each cluster has the characteristic of being acoustically similar and is therefore derived from a single speaker. The converse of the BIC rule applied during segmentation is used here. The ΔBIC between each new segment and all existing clusters is computed. Then, the segment is assigned to the cluster which engendered the largest positive ΔBIC. If none of the computed ΔBICs are positive, then the segment forms a new cluster. In our implementation, a BIC identity can be established in three seconds. (Segments shorter than 3 seconds are assigned the cluster ID 0 which is synonymous with SVAPI classifier’s ”Inconclusive” label.) Since the SVAPI-based speaker classifier uses eight seconds for a decision, all segments which run less than eight seconds and over three seconds can be assigned a labeled based on the consensus of the two classifiers, BIC and SVAPI. The algorithm is quite simple. All segments are concurrently processed using both classifiers. Therefore, each segment has two labels – a speaker label (name) and a BIC cluster ID. Both these assignments are duly recorded in our own internal data structure, the “turn buffer”. For every segment with an “Inconclusive” label, we obtain the cluster ID, and scour the turn buffer to find any previously recorded label corresponding to this ID. There are three possibilities – zero, one, or more than one corresponding labels. If there is one, we re-assign the “Inconclusive” label to that SVAPI label. If there is more than one, then we can re-assign it to the more frequently occurring label, if there is a preponderance of evidence towards one SVAPI label; otherwise no re-assignment is applied. If the turn buffer reveals no matching labels for a cluster id, no re-assignment can be done.

The Turn Buffer

As the segmentation process progresses, the segment boundary information, the turns, are recorded in a data structure along with a host of additional information. Each segment is identified by its leading turn. Hence with the segment number one can extract the audio frame number of the leading turn, the ranked identity set pertaining to that segment along with their respective scores, and the BIC-based cluster ID that has been assigned to the segment. The SVAPI-based speaker identification algorithm determines the “distance” of the candidate segment to each of the speakers in the data store. The shortest distance corresponds to the best match. Each test segment engenders a full list of matches which is trimmed to retain just the top 10 speakers. All of these 10 labels and the corresponding scores (distances) are recorded in the turn buffer. The turn buffer also retains same for the clustering process on a segment-by-segment basis. This information is later used in decision integration of the two lists to obtain a single consensus label for each segment.

Combining Classifiers Decisions

Speaker identification with the SVAPI engine does not perform adequately on utterances shorter than eight seconds. We therefore dismiss such segments with a catch-all "Inconclusive" label. Remember, that the BIC segmentation exhibits over-segmentation rate of 6%. To address this shortcoming, we have integated the result of the SVAPI- and BIC-based classifiers. The decision integration process starts after the audio ends (either naturally or by user action) and prior to the commencement of indexing.

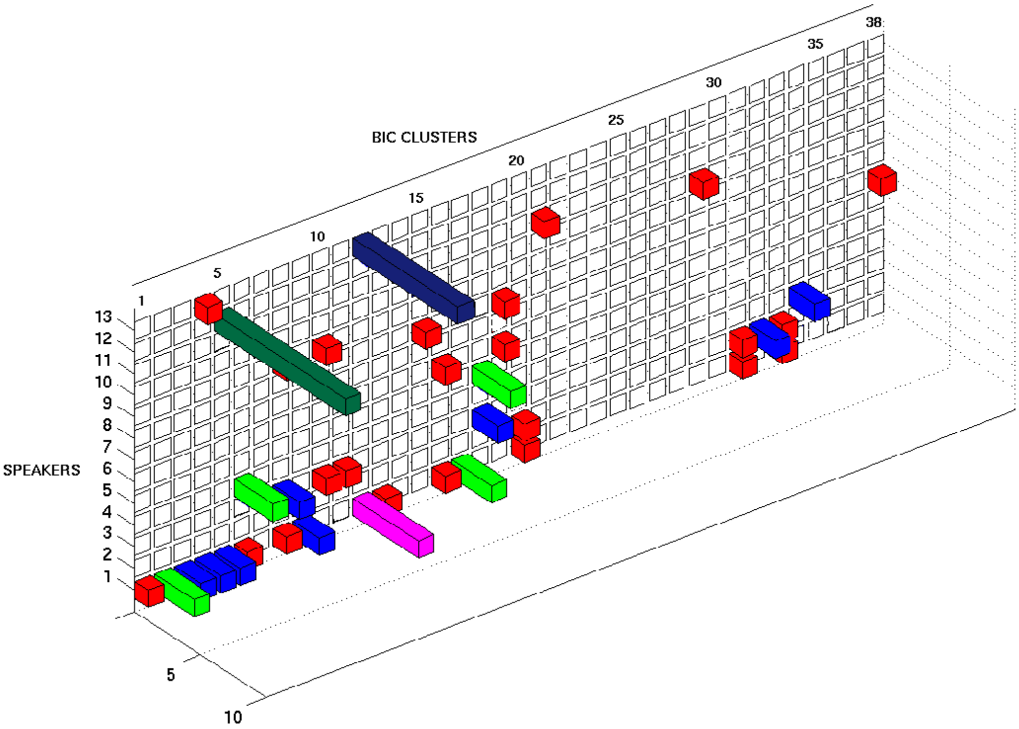


Figure 2: Scattergram for a 30-minute TV broadcast news segment including commercials showing conformity between SVAPI and BIC based speaker classification. There are 12 speakers in the segment. Speaker 1 and cluster ID 0 correspond to "Inconclusive."

This decision integration is best visualized by constructing a scattergram with the BIC-reported cluster ID's and the SVAPI-reported speakers along the two axes. Accumulation in any given resolution cell of the scattergram is viewed as increased SVAPI-BIC conformity. Thus to unravel the identity of segments shorter than eight seconds, we scan all resolution cells corresponding to a given cluster ID with a view to locate the one with the highest frequency of occurrence. If the frequency of a located resolution cell exceeds a user-defined threshold then all inconclusive segments with that cluster ID will assume the SVAPI-assigned name pertaining to that resolution cell. The turn buffer, described above, which relates the cluster ID's to the segment number and thence to their corresponding SVAPI-assigned ID's is updated with the new speaker identities.

Figure 2 shows the scattergram resulting from a 30-minute broadcast news video segment with 12 scored speakers using a 43-speaker data store. Note the conformity in the cells corresponding to speaker 12–cluster ID 5, and speaker 13–cluster 12. We deduce, therefore, that the "Inconclusive" segments with cluster IDs 5 and 12 belong to speakers 12 and 13 respectively. This corresponds to the ground truth. The threshold we use for re-assignment is three, i.e., accumulation must exceed three to warrant re-assignment. This has been confirmed by experimentation. Hence, the above speakers are the only ones re-labeled in this particular run of the test depicted in the figure. Therefore, all the "Inconclusive" segments with cluster ID 5 can be re-labeled as speaker 12, and cluster ID 12 as speaker 13. Pursuing the steps delineated above, the number of segments labeled as ”Inconclusive” were reduced from 27 to 20, a 26% reduction. The highest accumulations in the scattergram are due to the anchors in the broadcast news segments since they do tend to appear frequently during and in between news stories. (The news segment included commercials which engender more segments per commercial than warranted because of the frequent transition in acoustic conditions from music to voice to voice over music and more. For this reason, the commercial sections were excluded in our analysis.)

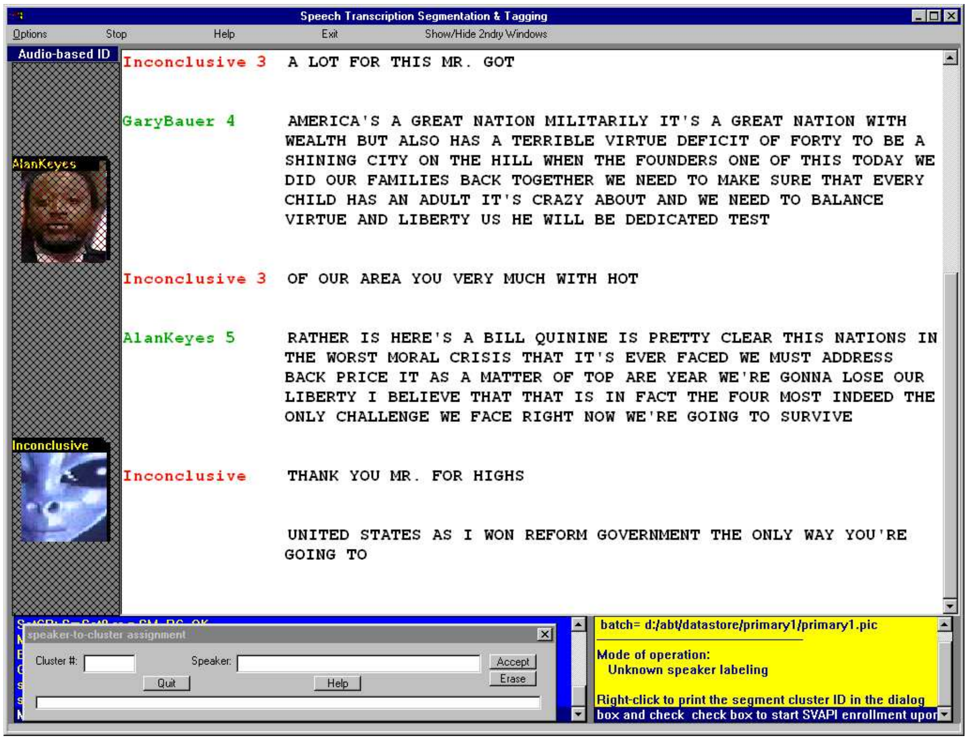
On-the-fly Speaker Enrollment

Speaker identification with the SVAPI identification engine calls for a pre-existing voice-prints of the speakers of interest so far in the course of an off-line process. The task of registering new speakers into a speaker data store is a manual one. It requires a collection of audio files for all speaker of interest, with one or more files for each speaker. Each file of set of files are submitted in sequence to the enrollment system along with a label for the speaker. (This is the label that the system later produces during the speaker identification stage.) This approach presumes the existence of sample voice files for all speaker of interest.

Using the segment cluster ID process in conjunction with SVAPI labeling it is possible to partially automate this process. The automation permits the nomination of speakers of interest when the speaker identification system is running. This comprises of:

- Feeding in a live or pre-recorded audio or video stream. As the input stream is transcribed, segmented, and SVAPI-labeled, the cluster ID's of the respective segments are also assigned.
- Then, the SVAPI-label of any segment is changed using an user-interface assisted pop-up to "correct" the SVAPI-label. The user can correct any segment label because the user is presumably listening to the audio and knows the identity of the speaker. Moreover, the speaker identification process is limited to the list of labels available in the data store.
- The turn buffer then records the changed values along with a flag indicating a user override. The system records the audio along with the new user-defined label for later enrollment.
- Finally, when the audio terminates, all the recorded audio clips for the different speakers as defined by the user are submitted in batch mode for enrollment. The data store is updated in one step.

Figure 3 shows a snapshot of our application for on-the-fly enrollment. The biggest advantage this approach offers other than the ease of use in adding speakers is that all this can be done in "production mode" of the system, i.e., when the system is actually being used for audio analysis and preparing indexes for audio/video retrieval. Another feature is that once a speaker name is entered, all subsequent segments with that cluster ID are assigned the same name. (In fact, all these segments are collected for later enrollment.) This feature permits a speaker to be tracked throughout transcription, which in itself can be useful application.



```
Inconclusive 3   A LOT FOR THIS MR. GOT

GaryBauer 4      AMERICA'S A GREAT NATION MILITARILY IT'S A GREAT NATION WITH
                 WEALTH BUT ALSO HAS A TERRIBLE VIRTUE DEFICIT OF FORTY TO BE A
                 SHINING CITY ON THE HILL WHEN THE FOUNDERS ONE OF THIS TODAY WE
                 DID OUR FAMILIES BACK TOGETHER WE NEED TO MAKE SURE THAT EVERY
                 CHILD HAS AN ADULT IT'S CRAZY ABOUT AND WE NEED TO BALANCE
                 VIRTUE AND LIBERTY US HE WILL BE DEDICATED TEST

Inconclusive 3   OF OUR AREA YOU VERY MUCH WITH HOT

AlanKeyes 5      RATHER IS HERE'S A BILL QUININE IS PRETTY CLEAR THIS NATIONS IN
                 THE WORST MORAL CRISIS THAT IT'S EVER FACED WE MUST ADDRESS
                 BACK PRICE IT AS A MATTER OF TOP ARE YEAR WE'RE GONNA LOSE OUR
                 LIBERTY I BELIEVE THAT THAT IS IN FACT THE FOUR MOST INDEED THE
                 ONLY CHALLENGE WE FACE RIGHT NOW WE'RE GOING TO SURVIVE

Inconclusive     THANK YOU MR. FOR HIGHS

                 UNITED STATES AS I WON REFORM GOVERNMENT THE ONLY WAY YOU'RE
                 GOING TO
```

batch= d:/abt/datastore/primary1/primary1.pic

Mode of operation:
Unknown speaker labeling

Right-click to print the segment cluster ID in the dialog box and check check box to start SVAPI enrollment upon

Figure 3: On-the-fly enrollment in progress. The user interface at the bottom left of the figure is active throughout the duration of the application. The cluster ID appears along side the speaker label as seen. A speaker label assignment can be altered by entering the cluster ID in the “Cluster ID” field, the new speaker label in the “Speaker” field, and clicking on “Accept”. Any speaker label change will replace all the assignments for that cluster ID in the turn buffer. Enrollment of speakers occurs when the application ends.

Conclusion

We have presented a solution for coping with the problem of building speaker indexes with short speaker segments, arising from genuine short utterances or over-segmentation. We extended our BIC-based segmentation component to act as a second classifier by providing a cluster ID for reported speaker segments in addition to the SVAPI labels. We then integrated these cluster IDs with those assigned by the SVAPI-based classifier to overturn the decisions dismissing many short speaker segments (less than eight seconds) as “Inconclusive.” In our tests, some 26% of such segments were revived and assigned new speaker labels which conformed to the ground truth without any mis-assignment.

We further used the BIC cluster IDs to make on-the-fly enrollment of new speakers a reality which eliminates the need to prepare separate audio clips for training new speakers for addition into the speaker database. With this feature existing speaker data stores can be augmented with additional speakers while in performance mode. Formerly, this had to be done when the system was off-line.

We plan on continuing our research along these lines incorporating results from multiple classifiers while retaining the all-important real-time nature of the underlying system. Possible avenues for further study include the integration of face recognition as yet another “speaker” identification scheme.

References

[1] Akaike, H. (1974). A New Look at the Statistical Model for Identification. *IEEE Transactions on Automatic Control*, AC–19, pp. 716– –723.

[2] Beigi, H.S.M., Maes, S., Chaudhari, U.V, & Sorensen, J.S. (1998). IBM Model-based and Frame-by-frame Speaker Recognition. In *Proceedings of Speaker Recognition and its Commercial and Forensic Applications*.

[3] Chen, S.S. & Gopalakrishnan, P.S. (1998). Speaker, Environment and Channel Change Detection and Clustering Via the Bayesian Information Criterion. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop* (pp. 127– –132).

[4] Delacourt, P., Kryze, D., & Wellekens, C.J. (1999). Speaker-based Segmentation for Audio Data Indexing. In *Proceedings of EuroSpeech99* (pp. 1195– –1198).

[5] Hirschberg, J., Whittaker, S., Hindle, D., Periera, F., & Singhal, A. (1999). Finding Information in Audio: A New Paradigm for Audio Browsing and Retrieval. In *Proceedings of the ESCA Tutorial and Research Workshop*.

[6] Satoh, S., Nakamura, Y., & Kanade, T. (1999). Name-It: Naming and Detecting Faces in News Videos. *IEEE Multimedia*, 6(1), 22– –35.

[7] Srinivasan, S., Petkovic, D., Ponceleon, D., & Viswanathan, M. (1999). The CueVideo Spoken Media Retrieval System. *IBM Research Report*, RJ 10143 (95018).

[8] Trischler, A. & Gopinath, R.A. (1999). Improved Speaker Segmentation and Segments Clustering Using the Bayesian Information Criterion. In *Proceedings of EuroSpeech99* (pp. 679– –682).

[9] Viswanathan, M., Beigi, H.S.M., Dharanipragada, S., & Tritschler, A. (1999). Retrieval from Spoken Documents Using Content And Speaker Information. In *Proceedings of International Conference on Document Analysis and Retrieval* (pp. 567– –572).

[10] Viswanathan, M., Beigi, H.S.M., Dharanipragada, S., Maali, F., & Tritschler, A. (2000). Multimedia Document Retrieval Using Speech and Speaker Recognition. To appear: *International Journal of Document Analysis and Recognition*, Spring 2000.

[11] Wold, E., Blum, T. & Keislar, D. (1996). Content-based Classification, Search, and Retrieval of Audio. *IEEE Multimedia*, 3(3), 27– –36.