DETECTION OF TARGET SPEAKERS IN AUDIO DATABASES

Ivan Magrin-Chagnolleau, Aaron E. Rosenberg and S. Parthasarathy

AT&T Labs Research - Florham Park, New Jersey - USA
ivan@ieee.org - aer@research.att.com - sps@research.att.com

ABSTRACT

The problem of speaker detection in audio databases is addressed in this paper. Gaussian mixture modeling is used to build target speaker and background models. A detection algorithm based on a likelihood ratio calculation is applied to estimate target speaker segments. Evaluation procedures are defined in detail for this task. Results are given for different subsets of the HUB4 broadcast news database. For one target speaker, with the data restricted to high quality speech segments, the segment miss rate is approximately 7%. For unrestricted data, the segment miss rate is approximately 27%. In both cases the segment false alarm rate is 4 or 5 per hour. For two target speakers with unrestricted data, the segment miss rate is approximately 63% with about 27 segment false alarms per hour. The decrease in performance for two target speakers is largely associated with short speech segments in the two target speaker test data which are undetectable in the current configuration of the detection algorithm.

1. INTRODUCTION

The problem addressed in this article is the detection of one or more target speakers in broadcast news programs. Detection, in this context, means the estimation of a beginning and an ending time for each segment in which a target speaker is speaking. This problem is an emerging one which has been reported on recently [6, 7, 8, 5]. It is a potentially important problem since, as more and more audio and multimedia data are recorded and archived, the need grows for useful cues to segment, classify, and organize this data.

In this paper, two sets of experiments are described. In the first, the detection of a single target speaker is addressed; the second generalizes the problem to two target speakers. The evaluation procedures are slightly different in each case, and are described in detail in the article.

2. DATABASE

The HUB4 database [1] is composed of 174 broadcasts from 11 news and commentary programs. For each program, from 6 to 37 broadcasts corresponding to different dates are available. The duration of the longest broadcast is 2 hours, the duration of the shortest is 26 minutes. Speech portions of each broadcast have been transcribed, segmented, and labeled in terms of speaker; mode (spontaneous or planned); fidelity, classifying the quality of the recording environment and transmission channel (High, Medium, or Low); and background, describing the nature of secondary audio material mixed with the primary speech signal (Speech, Music, or Other). Commercials and sports results have not been transcribed.

For our experiments, the data are classified into 4 categories as follows. The category *high-fidelity* refers to speech labeled High fidelity with no background; the category *clean* refers to speech with all 3 quality categories but no background; the category *allspeech* refers to data of all quality categories with and without background; and the category *alldata* refers to data including the previous *allspeech* category plus all the untranscribed portions. When experiments refer to test data belonging to the *high-fidelity* category only, it means that, for each program, all the data belonging to the *high-fidelity* category are artificially concatenated together, and similarly for the *clean* and *allspeech* categories. The category *alldata* corresponds to the data as originally provided.

The one-target-speaker detection experiments are conducted on the subset aABC_NLI of the HUB4 database, which corresponds to the program ABC Nightline. The target speaker is Ted Koppel. 6 broadcasts are set aside for training the target model and two background models. 12 broadcasts are used as test broadcasts. The duration of the longest test broadcast is 35 minutes, the duration of the shortest is 26 minutes. The remaining 5 programs are not used since they contain no data of the target speaker.

The two-target-speaker detection experiments are conducted on the subset bABC_WNN of the HUB4 database, which corresponds to the program ABC World News Now. The target speakers are Mark Mullen (T1) and Thalia Assures (T2). 8 broadcasts are used to train the two target models and two background models. 16 broadcasts are used as test broadcasts. The duration of the longest test broadcast is 31 minutes; the duration of the shortest is 29 minutes.

3. MODELING

To detect a target speaker in a program, at least two models need to be built: a target speaker model and a background model which is intended to represent speech from speakers other than the target speaker or other types of sounds. These models are built using feature vectors extracted from labeled segments. In the case of several target speakers, at least one model for each target speaker is needed. Finally, more generally, there could be several background models, representing different types of non-target speaker sounds, and several models for each target speaker, representing different speech qualities. In the experiments presented in this article, only one model is built for each target speaker, and one or two background models are used.

The feature vectors consist of cepstral and Δ-cepstral coefficients extracted in the following way: a Winograd Fourier Transform (WFT) is computed on Hamming windowed signal frames of 31.5 ms (504 samples) at a frame rate of 10 ms (160 samples). For each frame, spectral vectors of 31 Mel-Scale Triangular Filter Bank coefficients are then calculated from the Fourier Transform power spectrum, and expressed in logarithmic scale. An algorithm based on a bimodal Gaussian modeling of the energy (sum of the 31 coefficients) is then used to remove low energy vectors. This algorithm removes between 10 % and 30 % of the frames. Cepstral coefficients c_1 to c_{20} are then calculated using cosine transforms. The coefficient c_0, which corresponds to the energy, is omitted. Finally, Δ-coefficients are calculated over 5 frames, providing 20 additional coefficients. The feature vectors are thus 40-dimensional.

Gaussian mixture models (GMM) [3] are used for modeling target speakers and backgrounds. These models have been used successfully in text-independent speaker recognition [4]. The number of mixture components is set equal to 64, and diagonal covariance matrices are used. The parameters of the GMM are estimated using the Expectation Maximization (EM) algorithm [3], initialized by a binary splitting Vector Quantization (VQ) algorithm [2].

Only *high-fidelity* speech is used to build each target speaker model. For each target speaker, three 90 second segments of speech, extracted from 3 broadcasts uttered by the target speaker, are concatenated together. Feature vectors are then extracted, and a GMM model is trained for each target speaker, as described previously.

For the first background model, *high-fidelity* speech only is used. Samples from 4 female and 4 male speakers, 60 seconds each, are concatenated together. The speakers for this background model are chosen from different broadcasts. Feature vectors are then extracted, and a GMM model is trained, as described previously. This background model is referred to as λ_{B1}.

For some experiments, a second background model is used. Only non-speech segments are used for this second model, that is music only (10 %), noise only (10 %), and commercials (80 %). Three 90 second segments, chosen from 3 broadcasts, are concatenated together. Feature vectors are then extracted, and a GMM model is trained, as described previously. This background model is referred to as λ_{B2}.

None of the broadcasts and none of the speakers used to build the target or the background models are used in the test data.

4. DETECTION ALGORITHM

The simplest experimental problem is the detection of one target speaker, using only one target model (λ_T) and one background model (λ_B).

Given a test broadcast, the task is to detect the segments in which the target speaker is speaking, that is to estimate the beginning and the end of these segments. First of all, feature vectors are extracted from the test broadcast, using the signal analysis described previously. Let $\{\mathbf{x}_t\}_{1\leq t\leq T}$ denote the sequence of feature vectors. For each vector $\mathbf{x}_t$, we calculate the likelihood of $\mathbf{x}_t$ given the model λ_T, denoted by $\mathcal{L}(\mathbf{x}_t|\lambda_T)$, the likelihood of $\mathbf{x}_t$ given the model λ_B, denoted by $\mathcal{L}(\mathbf{x}_t|\lambda_B)$, and finally the likelihood ratio of $\mathbf{x}_t$ given λ_T and λ_B, denoted by $\mathcal{R}(\mathbf{x}_t|\lambda_T;\lambda_B)$, whose logarithm is expressed by:

$$\log \mathcal{R}(\mathbf{x}_t|\lambda_T;\lambda_B) = \log \mathcal{L}(\mathbf{x}_t|\lambda_T) - \log \mathcal{L}(\mathbf{x}_t|\lambda_B)$$

Before applying the segmentation algorithm, smoothing is needed to attenuate variations of the logarithm of the likelihood ratio. The smoothing is an arithmetic mean of a specified number of consecutive values of the logarithm of the likelihood ratio. Two parameters define the smoothing: the number of values used for the average calculation, denoted by τ; and the delay between two calculations, that is the number of feature vectors between two average calculations, denoted by d. The smoothed value of the logarithm of the likelihood ratio is:

$$\begin{aligned}
&\overline{\log \mathcal{R}}(\mathbf{x}_{t_0-t'}, .., \mathbf{x}_{t_0+t'}|\lambda_T;\lambda_B) \\
&= \frac{1}{\tau}\sum_{t=t_0-t'}^{t_0+t'} \log \mathcal{R}(\mathbf{x}_t|\lambda_T;\lambda_B) \\
&= \overline{\log \mathcal{L}}(\mathbf{x}_{t_0-t'}, .., \mathbf{x}_{t_0+t'}|\lambda_T) - \overline{\log \mathcal{L}}(\mathbf{x}_{t_0-t'}, .., \mathbf{x}_{t_0+t'}|\lambda_B)
\end{aligned}$$

with $\tau = 2t' + 1$. In our experiments, τ is set to 100 vectors (1 second) and d to 20 vectors (0.2 second).

A segmentation algorithm is then applied on the average values previously calculated. The algorithm is described in detail in the Block 1.

```
From first block to last block, do:
• If $v > \theta_1^B$ and $detectFlag = 1$ and $endFlag = 1$
  endFlag = 0
• If $v > \theta_1^B$ and beginFlag = 0
  Set the beginning time for a possible segment
  beginFlag = 1 (a beginning time has been set)
• If $v > \theta_2^B$ and detectFlag = 0
  detectFlag = 1 (a target speaker segment has been detected)
• If $v < \theta_1^E$ and $beginFlag = 1$ and $detectFlag = 0$
  beginFlag = 0
• If $v < \theta_1^E$ and $detectFlag = 1$ and $endFlag = 0$
  Set the ending time for the detected segment
  endFlag = 1 (an ending time has been set)
• If $v < \theta_2^E$ and $detectFlag = 1$
  Record the detected segment
  beginFlag = 0 (reset of beginFlag)
  detectFlag = 0 (reset of detectFlag)
  endFlag = 0 (reset of endFlag)
```

Block 1: *Segmentation algorithm.*

v is the average value of the logarithm of the likelihood ratio for a block, θ_1^B and θ_2^B are two thresholds used for the detection of the beginning of a segment, and θ_1^E and θ_2^E are two thresholds used for the detection of the end of a segment. In our experiments, the values of the thresholds are the following: $\theta_1^B = 0$, $\theta_2^B = 3$, $\theta_1^E = 0$, and $\theta_2^E = -0.5$.

Finally, the minimum duration for an estimated segment is set to 2.5 seconds (every segment whose duration is smaller is omitted), and the minimum interval between two consecutive segments to 1 second (two consecutive segments are merged if the interval between them is smaller).

The segmentation algorithm provides the estimated beginning and end times for the target speaker segments.

In the general case, several background models are available (λ_{B1}, λ_{B2}, ...), as well as several target speakers and several models for each target speaker (λ_{T1}, $\lambda_{T1'}$, ..., λ_{T2}, $\lambda_{T2'}$, ...). The average value of the logarithm of the log-likelihood of a block is then given by:

$$\begin{aligned}
&\overline{\log \mathcal{R}}(\mathbf{x}_{t_0-t'}, .., \mathbf{x}_{t_0+t'}|\lambda_{T1}, \lambda_{T1'}, .., \lambda_{T2}, \lambda_{T2'}, ..; \lambda_{B1}, \lambda_{B2}, ..) \\
&= \max_{T\in\{T1,T1',..,T2,T2',..\}} \overline{\log \mathcal{L}}(\mathbf{x}_{t_0-t'}, .., \mathbf{x}_{t_0+t'}|\lambda_T) \\
&- \max_{B\in\{B1,B2,..\}} \overline{\log \mathcal{L}}(\mathbf{x}_{t_0-t'}, .., \mathbf{x}_{t_0+t'}|\lambda_B)
\end{aligned}$$

Quality	high		clean		allspeech		alldata	
Total duration	141 min		194 min		242 min		359 min	
# target segments	137		257		318		354	
Duration	61 min		69 min		78 min		78 min	
Background	B1	B1, B2	B1	B1, B2	B1	B1, B2	B1	B1, B2
# estimated segments	129	129	195	195	238	237	256	254
FMIR	3.90 %	4.01 %	4.92 %	5.07 %	6.73 %	7.23 %	8.89 %	9.57 %
FFAR	10.75 %	10.52 %	9.19 %	8.80 %	7.22 %	7.00 %	5.66 %	5.44 %
SMIR	7.30 %	7.30 %	19.07 %	19.46 %	25.16 %	26.42 %	27.40 %	29.66 %
SFAR	5.53 / hour	5.53 / hour	8.35 / hour	7.42 / hour	5.95 / hour	5.21 / hour	4.35 / hour	4.18 / hour

Table 1: *Results of the one-target-speaker detection experiments.*

In the case of several target speakers, each target segment is labeled with the identity of the detected target speaker.

5. EVALUATION

The performance of the detection algorithm has been evaluated in two ways, one giving the performance at the frame level, the other at the segment level. Each frame inside an estimated target speaker segment is referred to as an *estimated target frame*, each frame outside as an *estimated non-target frame*. The estimated target frames and the estimated target segments are then compared with those provided by the database. Each frame inside a labeled target speaker segment provided with the database is referred to as a *labeled target frame*, each frame outside as a *labeled non-target frame*.

The **Frame-level MIss Rate (FMIR)** is the number of labeled target frames which have not been estimated as target frames (that is which have been estimated as non-target frames or as target frames from another speaker in the case of several target speakers), divided by the total number of labeled target frames.

The **Frame-level False Alarm Rate (FFAR)** is the number of estimated target frames which are in fact labeled non-target frames, divided by the total number of labeled non-target frames.

In the case of the detection of several target speakers, the proportion of missed frames due to a confusion with another speaker is also indicated in parentheses by the **Frame-level COnfusion Rate (FCOR)**, which is the number of labeled target frames which have been estimated as target frames of another speaker, divided by the total number of labeled target frames. FCOR is a component of FMIR.

A missed segment is a labeled target segment for which the proportion of frames estimated as target frames (of the correct target speaker in the case of several target speakers) is less than the threshold of Frames Correctly Detected (FCD). For our experiments, FCD has been set to 75 %. The **Segment-level MIss Rate (SMIR)** is the number of missed segments divided by the total number of target segments.

In the case of one target speaker, a false alarm segment is an estimated target segment for which the proportion of labeled non-target frames is greater than the proportion of labeled target frames. The **Segment-level False Alarm Rate (SFAR)** is defined as the total number of false alarm segments divided by the total duration of the broadcast in hours.

The following definitions are needed to define the SFAR and the Segment-level Confusion Rate (SCOR) in the case of the detection of several target speakers. A *false-alarm* frame is an estimated target frame which is actually a non-target frame. A *confusion* frame is an estimated target frame which is actually a target frame of another target speaker. A *hit* frame is an estimated target frame which is actually a target frame of the correct speaker.

A false alarm segment is an estimated target segment for which the proportion of *false alarm* frames is greater than the proportion of *confusion* frames and also greater than the proportion of *hit* frames. The **SFAR** is then the total number of false alarm segments divided by the total duration of the broadcast in hours.

A confusion segment is an estimated target segment for which the proportion of *confusion* frames is greater than the proportion of *false alarm* frames and also greater than the proportion of *hit* frames. The **Segment-level COnfusion Rate (SCOR)** is then the total number of confusion segments divided by the total duration of the broadcast in hours.

6. EXPERIMENTAL RESULTS

Table 1 reports results for the one-target-speaker detection experiments. Results are reported for the 4 categories of sound: *high-fidelity*, *clean*, *allspeech*, and *alldata*. For each category, the total duration of test material, all 12 programs together, is indicated, as well as the number of target segments and their total duration. Results are given when B1 only is used as the background model, and when B1 and B2 are used together. In both cases, the number of estimated target segments is indicated, and the 4 error rates are given.

There is not a lot of difference between the results obtained when B1 is used alone, or when B1 and B2 are used together. With B1 only, the FMIR is generally slightly lower, but the FFAR is slightly higher. With data from the *alldata* category, the sum of the two rates is better when B1 is used alone (14.55 % versus 15.01 %). At the segment level, the same trend can be observed.

The FMIR degrades through each category from the *high-fidelity* category to the *alldata* category, because of the increase of the mismatch between the test data and the data used for the training of the models. This problem might be solved by training the target speaker model with data from different categories pooled together, or training several models for the target speaker using data from each category.

Conversely, the FFAR improves through each category from the *high-fidelity* category to the *alldata* category. This may be because there is proportionally less and less *high-fidelity* data through each category, *high-fidelity* data being more likely to be confused with the *high-fidelity* target model than data from the other categories.

Similar observations can be made at the segment level.

Table 2 reports results for the two-target-speaker detection experiments. The results are reported only for data from the *alldata*

Target speakers	T1			T2			T1, T2		
Segt. duration	> 4s	> 2s	All	> 4s	> 2s	All	> 4s	> 2s	All
# target segments	119	200	315	125	191	310	244	391	625
Duration	29 min	33 min	35 min	41 min	44 min	46 min	71 min	78 min	82 min
# est. segments	–	–	216	–	–	183	–	–	349
FMIR	14.06 %	19.34 %	22.70 %	17.40 %	21.42 %	23.90 %	36.52 %	38.79 %	40.27 %
(FCOR)	–	–	–	–	–	–	(24.47 %)	(24.55 %)	(24.69 %)
FFAR	–	–	9.75 %	–	–	7.21 %	–	–	18.38 %
SMIR	23.53 %	42.50 %	58.10 %	28.00 %	45.03 %	64.19 %	42.62 %	52.69 %	62.88 %
SCOR	–	–	–	–	–	–	–	–	5.27 / hour
SFAR	–	–	17.41 / hour	–	–	12.59 / hour	–	–	27.51 / hour

Table 2: *Results of the two-target-speaker detection experiments for the alldata category.*

category (total duration = 410 minutes), and when B1 only is used as background model. (The experiments using B1 and B2 do not give better results.) The results for each target speaker alone are given first, and then the results for the two target speakers together. The number of estimated target segments is indicated, and the 6 error rates are given. Because of the nature of the program, the two target speakers are often engaged in a dialog. This produces a high proportion of short target segments and also a small proportion of overlap segments. (4.5 % of all the labeled target frames overlap the other target speaker.) To study the effect of segment duration on performance, FMIR and SMIR results are shown for different values of minimum target segment durations.

Performance approaching the one-target-speaker results with Ted Koppel are obtained only for target segment durations greater than four seconds. In fact, most of the Ted Koppel segments (72 %) are longer than 4 seconds. For Mark Mullen, only 38 % of the segments are longer than 4 seconds, and 40 % for Thalia Assures.

Because the duration of most target segments is smaller than four seconds, the choice of the smoothing block length may be important. If this length is decreased, the FMIR and the SMIR may be improved, at the expense of degrading the FFAR and the SFAR. One solution to this problem could be a two-pass algorithm, each pass with different smoothing parameters. Different values for the minimum duration of an estimated segment have also been tested (from 2.5 seconds to 1 second), but the results do not change significantly. In fact, most of the very short segments are completely missed, whatever the minimum duration of an estimated segment is, due to the block smoothing. Some of them are combined into longer estimated segments comprising two or more short segments.

In some cases, short segments of one target speaker (particularly those which are shorter than the block smoothing length) are merged with longer adjacent segments of the other target speaker, again because of the block smoothing. Because of that, the contribution of the FCOR to the FMIR is high.

7. CONCLUSION

A new approach for detecting target speakers in audio databases has been proposed. This approach, based on Gaussian mixture modeling and a likelihood ratio calculation, gives good results if the duration of the target speaker segments is long enough. However, in the case of short segments, performance degrades, particularly at the segment level. More experiments are needed to provide better understanding of this problem, and the detection algorithm needs to be refined for the detection of short segments.

Further studies in speaker detection will also include the use of more than one model for each target speaker, and the use of other background models. The performance will also be studied as a function of the smoothing parameters and the segmentation algorithm parameters.

8. REFERENCES

[1] D. Graff, Z. Wu, R. MacIntyre, and M. Liberman. The 1996 broadcast news speech and language-model corpus. In *Proceedings of the 1997 DARPA Speech Recognition Workshop*, February 1997. Chantilly, Virginia.

[2] Yoseph Linde, Andres Buzo, and Robert M. Gray. An algorithm for vector quantization design. *IEEE Transactions on Communications*, 28(1):84–95, January 1980.

[3] Geoffrey J. McLachlan and Kaye E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988.

[4] Douglas A. Reynolds. Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, 17(1–2):91–108, August 1995.

[5] Aaron E. Rosenberg, Ivan Magrin-Chagnolleau, S. Parthasarathy, and Qian Huang. Speaker detection in broadcast speech databases. In *Proceedings of ICSLP 98*, December 1998.

[6] M.-H. Siu, G. Yu, and H. Gish. An unsupervised sequential learning algorithm for the segmentation of speech waveforms with multiple speakers. In *Proceedings of ICASSP 92*, volume II, pages 189–192, 1992.

[7] M. Sugiyama, J. Murakami, and H. Watanabe. Speech segmentation and clustering based on speaker features. In *Proceedings of ICASSP 93*, volume II, pages 395–398, 1993.

[8] Lynn Wilcox, Francine Chen, Don Kimber, and Vijay Balasubramanian. Segmentation of speech using speaker identification. In *Proceedings of ICASSP 94*, volume 1, pages 161–164, 1994.