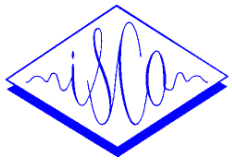

7th International Conference on Spoken Language Processing [ICSLP2002]
Denver, Colorado, USA
September 16-20, 2002

UNSUPERVISED SPEAKER SEGMENTATION OF TELEPHONE CONVERSATIONS

Aaron E. Rosenberg, Allen Gorin, Zhu Liu, S. Parthasarathy

Voice Enabled Services Research Lab
AT&T Labs
Florham Park, NJ 07932 USA

ABSTRACT

A process for segmenting 2-speaker telephone conversations by speaker with no prior speaker models is described and evaluated. The process consists of an initial segmentation using acoustic change and pause detection, segment clustering, and iterative modeling of segment clusters and resegmentation. The technique has been evaluated on 6, approximately 3 min long, customer care conversations. The technique does not resolve short ($<$ 2 secs) or overlapping segments very well, but is capable of detecting longer segments ($>$ 4 secs) with miss rates of the order of 10% and confusion rates 2% or less.

1. INTRODUCTION

The segmentation of multispeaker audio data by speaker has received considerable attention in recent years [8, 9, 7, 4, 1, 2, 6]. Applications that have been considered include the following: indexing archived recorded broadcast news programs by speaker to facilitate browsing and retrieval of desired portions; tagging speaker specific portions of data to be used for adapting speech models in order to improve the quality of automatic speech recognition (ASR) transcriptions; tracking speaker specific segments in telephone conversations to aid in surveillance applications. The specific problem considered in this paper is the unsupervised segmentation by speaker of telephone conversations between two speakers. The solution to the problem provides a set of segments for each speaker. The intended application is aid in locating and selecting speech data for use in training an ASR system for a natural speech automatic call classification system. The calls are from customers who are making some sort of customer care request for information or service. In order to train the automated system it is necessary to find the customer segment or segments in training conversations in which the request is stated in order to label such segments and add the data to the training database. Such segments are usually the longest segments spoken by the customer and typically occur early in the conversation.

An iterative approach to speaker segmentation is proposed here in which acoustic change detection and segment clustering initializes the process in a way similar to processes described in earlier studies [4, 6]. Following this, a Gaussian Mixture Model (GMM) is constructed for the pooled data associated with each segment cluster. The input sample is compared with each such model to output a detection score as a function of time which is used to obtain a new segmentation estimate. This process is iterated until stable segmentations are obtained. The motivation for adopting this iterative approach is that the initial segmentation we obtain is generally incomplete and imprecise. This is attributable to the short duration of a large number of the speaker segments in telephone conversations. In order to resolve short segments, the data window used to detect acoustic changes and mark the segments must also be short to avoid including more than one speaker change in the window. However, the generalized likelihood ratio (GLR) computation [9, 1] which we use for acoustic change detection becomes variable and unstable for short duration windows. To compensate for this instability, our initial segmentations are generally underestimates. The iterative process following the initial segmentation and clustering is expected to help refine and fill out the segmentations.

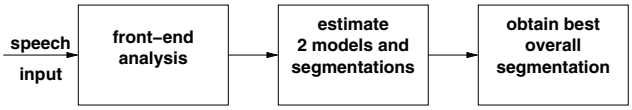

Fig. 1. Overall processing block diagram.

2. SEGMENTATION PROCESS

An overall block diagram of the segmentation process is shown in Fig. 1. The speech sample is input to the front end processing described below. This is followed by the unsupervised segmentation and modeling process. It is assumed that the speech sample consists of a conversation between two speakers. The output of this stage is two distinct segmentations corresponding to two speakers. Finally, in a postprocessing step, the two segmentations are combined into an overall optimum segmentation according to a maximum likelihood criterion. Any residual overlap between the two component segmentations is eliminated by this postprocessing.

2.1. Front end processing

Each input sample is digitized at an 8 kHz rate. Twelfth-order cepstral coefficients are calculated every 10 ms (80 samples) over 20 ms (160 sample) windows by applying a discrete cosine transform (DCT) to the sample data in the window. Real-time energy normalization is applied with a 300 ms look ahead window. The cepstral coefficients are augmented by twelfth-order delta-plus delta-delta-cepstral coefficients plus energy, delta-energy, and delta-delta-energy coefficients. Frames with energy falling below a specified level below peak energy are eliminated.

2.2. Initial Segmentation

An overall block diagram of the segmentation and modeling process is shown in Fig. 2. Following the front-end analysis, the processing continues with an initial segmentation and clustering of segments (Block 1). The Generalized Likelihood Ratio (GLR) formulation is used to carry out this initial segmentation. Suppose there are 2 segments, X_1 and X_2 represented by feature vectors $X_1 = \{x_{11}, x_{12}, \ldots, x_{1N_1}\}$ and $X_2 = \{x_{21}, x_{22}, \ldots, x_{2N_2}\}$,

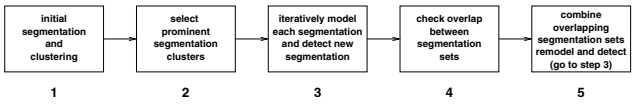

Fig. 2. Segmentation and modelling block diagram.

each segment containing speech generated by one speaker. Let the null hypothesis H_0 be that the speakers in the 2 segments are the same, and the alternative hypothesis, H_1, be that they are different. Let $L(X_1;\lambda_1)$ and $L(X_2;\lambda_2)$ be the likelihoods of X_1 and X_2 where λ_1 and λ_2 represent model parameters which maximize the likelihoods. Similarly let $X = X_1 \cup X_2$ be the union of X_1 and X_2 and $L(X;\lambda_{1+2})$ be the maximum likelihood estimate for X. Then

$$LR = \frac{L(X;\lambda_{1+2})}{L(X_1;\lambda_1)L(X_2;\lambda_2)} \quad (1)$$

In this study X_1 and X_2 are adjacent equal-duration intervals in a window interval X and the model parameters λ_1, λ_2, and λ_{1+2} are GMM's derived from a GMM representing the whole data sample by adapting the component weights in the respective intervals, holding the means and variances fixed. To determine the location of boundaries between speaker segments, the GLR function is calculated over successive overlapping windows throughout the data sample. When the window is contained within a speaker segment the value of LR should be close to 1. If the window interval X is centered over a boundary between speaker segments then the LR function should exhibit a distinct dip. For the GLR to perform well the window should be long enough to obtain stable statistics yet short enough to avoid containing more than one speaker segment change. In this study, the conversations are likely to contain many short, one-word response segments. The window duration is shortened to 1.6 secs and the window is shifted every 0.2 secs to resolve many such segments, but this duration generates a significant amount of variability in the GLR function as a function of time due to the variation in window content. Dips in the GLR computation as a function of time are not generally distinctly discernible. Therefore, instead of estimating speaker segments by detecting GLR dips, we select regions in which the GLR functions remains above a specified threshold for at least some minimum duration. Such regions are likely to be associated with a single speaker (or channel) but generally do not comprise an entire speaker segment. Pauses are possible, but not reliable, indicators of speaker changes. Pause locations are combined with the segment estimates obtained from the GLR function primarily by not allowing a segment to overlap a significant pause.

A plot of the log of LR as a function of time for a fragment of a conversation sample is shown in Fig. 3a. Estimated segments are marked by dashed vertical lines. Each of these estimated segments contain speech from a single speaker, as hypothesized.

2.3. Clustering

The segments obtained by scanning the input sample with the windowed GLR function are input to an agglomerative hierarchical clustering algorithm [5] in order to associate groups of segments with different speakers. The clustering procedure is used to obtain an initial grouping of segments. Models are created for the pooled segments in each cluster and the input is rescanned with these models to resegment the data. The process continues iteratively with the ultimate goal in our 2-speaker application of providing two groups of segments, one for the customer and the other for the representative. It is not possible to attach an actual speaker

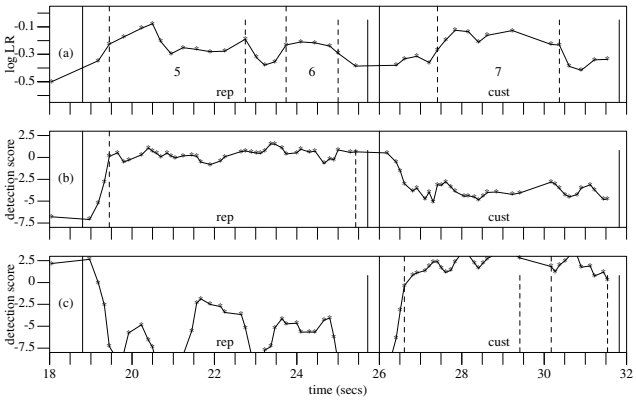

Fig. 3. 14-second fragment of a sample showing (a) the likelihood ratio score used to obtain initial segments, (b) the detection score for the segment cluster model for the representative, and (c) the detection score for the segment cluser model for the customer. The solid vertical lines mark the actual segments while the dashed lines mark the estimated segments (long for segment start and short for segment end).

label to each group unless some additional *a priori* information is provided. In our case, it is known that the first segment in the conversation is a greeting spoken by the representative. Thus all segments grouped with that first segment should be labeled as spoken by the representative.

The input to the clustering process is a table of pairwise distances between each segment and every other segment. Each segment is modeled by a low-order (typically 2- or 4-component) GMM and a symmetric distance measure is derived from the likelihood of each segment with respect to every other segment.

The clustering algorithm starts with each segment in a group of its own. At each iteration it merges two groups to form a new group such that the merger produces the smallest increase in distance. The "compact" criterion is used for group distance in which the distance between two groups is defined as the largest distance between any two members of each group. The process continues until all segments are merged into one group and the output is a binary classification tree whose nodes indicate segment groupings and whose levels indicate the merging distances. The process concludes by selecting the most prominent clusters (Block 2). This is an empirical process which selects at least 2 non-intersecting clusters at the lowest merge levels such that the clusters contain at least a specified number of segments. Currently, this minimum is set at 1/3 the total number of segments. However, it can be adjusted downward to force at least two selected clusters. Ideally this process outputs 2 prominent clusters corresponding to the two speakers in the conversation. The case of more than two clusters is considered in Section 2.5.

2.4. Segmentation modeling and detection

The process continues with modeling and resegmentation (Block 3). The data in each cluster of segments selected by the clustering process is pooled and a GMM is constructed to represent it. The input sample is scanned to calculate a frame-by-frame likelihood ratio score for the cluster model compared with a background model representing the whole sample. Both models are typically 64-component GMMs. These scores are used to estimate a new segmentation in a previously described method [7]. In the experiments carried out here, the modeling/resegmentation process is it-

erated 3 times in order to obtain stable segmentations. Likelihood ratio detection scores for 2 segmentation models created after 3 iterations are shown in Fig. 3b and Fig. 3c. The estimated segments are shown by the dashed lines. It can be clearly seen that one model represents the representative while the other represents the customer.

2.5. Checking segmentation overlap

The final segmentations associated with each initial cluster are compared with each other to determine the amount of overlap between them (Block 4). If there are two segmentations and the overlap between them falls below a specified threshold, the process is considered successful and each segmentation is considered to be associated with one of the speakers in the conversation. If the segmentations overlap significantly, the process is considered to fail. This outcome implies that the segments in the initial segmentations were excessively contaminated by the presence of data from other speakers and that the iterative modeling and detection process could not overcome the original contamination. If there are more than two segmentations and there is no overlap among the segmentations, the process is also considered to fail. It could mean that there are more than the hypothesized two speakers in the conversation. If, however, one or more of the final segmentations overlaps with another, the overlapped segments are pooled and the modeling/detection process is restarted (Block 5). The outcome is then checked anew for overlapping segmentations and continues until two distinct segmentations are obtained.

2.6. Optimum Overall Segmentation

After two final segmentations are obtained, a postprocessor is invoked to combine them into an overall optimum segmentation (see Fig. 1). A segmentation lattice is created that allows segment changes to occur at any of the segment boundaries from both segmentations. The best path through this lattice, a sequence of non-overlapping segments, is obtained such the overall segmentation likelihood is maximized.

3. EXPERIMENTAL EVALUATION

3.1. Database

The experimental database consists of 6-minute recordings of telephone conversations between AT&T long distance customers and customer care representatives. The calls are initiated by the customer to make some sort of billing or service inquiry. Typically, a recording contains 1 or 2 2 to 3 minute conversations. Each recording is truncated at 6 minutes even if a conversation is not completed. For the purposes of this study 6 recordings have been selected at random. The customer-representative conversation has been extracted from each recording and hand labeled.

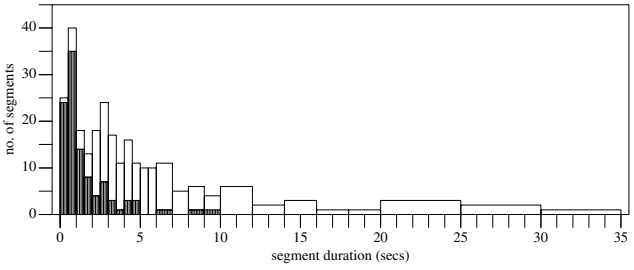

Fig. 4. Histogram of speaker segment durations; the filled blocks show the number of misses and confusions.

A histogram of segment durations for the labeled segments pooled from all 6 conversations is shown in Fig. 4. There are many short duration segments consisting typically of single-word responses. The median segment duration is 2.76 secs while the average is 3.95 secs. Overlap between customer and representative segments can also be found. The customer representatives are all female and all but one of the customers is also female.

3.2. Performance Measurements

The object of the experimental evaluation is to determine how well the processing described here can detect the speaker segments or turns in the sample conversations in our database. Two sets of measurements are used. The first set of measurements is associated with segment detectability, measuring the fraction of actual segments correctly detected and labeled. If an actual segment is overlapped by an estimated segment by at least a fraction $pdet \geq 0.5$, then it is counted as either a hit or a confusion. If an actual segment does not count as either a hit or a confusion, then it is considered a miss. If the total number of actual segments is $nsgs$, and the total number of hits, confusions, and misses is $nhit$, $ncnf$, and $nmis$, respectively, then $nsgs = nhit + ncnf + nmis$. The segment miss and confusion rates, psg_{mis} psg_{cnf}, are defined as $nmis/nsgs$ and $ncnf/nsgs$, respectively. We don't know *a priori* which segmentation and model corresponds with which speaker. To determine whether a segment is "correctly" detected, we select the mapping between models and speakers which maximizes the hit rate.

It is also possible for an estimated segment to overlap no actual segment from either speaker. This counts as a false alarm. This occurs rarely in our sample conversations since, after energy thresholding,there are few signal portions not generated by one or the other speaker. The number of false alarms defined in this way is negligible in our evaluation and will not be reported.

The second set of measurements specifies how close the detected segment durations are to actual segment durations. Let dur_{act} be the total number of frames in actual speaker segments in the sample. Let dur_{hit} and dur_{cnf} be the total number of actual frames that overlap estimated frames correctly and incorrectly, respectively. Then the frame hit rate and frame confusion rates are given by $pfr_{hit} = dur_{hit}/dur_{act}$ and $pfr_{cnf} = dur_{cnf}/dur_{act}$, respectively. The frame miss rate is given by $pfr_{mis} = 1 - pfr_{hit} - pfr_{cnf}$.

We can also define some useful measurements to examine the performance of the clustering procedure. Let $ovlap(sega, segb)$ be the duration of the overlap between segmentations a and b, and $dur(sega)$ be the total duration of the segmentation $sega$. Let $act(j)$ and $est(j)$ be the actual and estimated segmentation for speaker j in a given conversation sample. The coverage for the segmentation estimate for speaker j, the contamination associated with speaker j by the segmentation for the other speaker j', and the overlap between the two segmentation estimates are given by

$$
\begin{aligned}
p_{cvrg}(j) &= ovlap(act(j), est(j))/dur(act(j)) \\
p_{cntm}(j) &= ovlap(act(j), est(j'))/dur(act(j)) \\
p_{ovlap}(j) &= ovlap(est(j), est(j'))/\min(dur(est(j)), dur(est(j')))
\end{aligned}
$$

3.3. Results

We have described several parameters controlling the process which are possible experimental variables. The results reported here are restricted to just two of these, the minimum duration of segments

avg. no. of segs	avg. error rates(%)			
	psg_{mis}	psg_{cnf}	pfr_{mis}	pfr_{cnf}
42.8	25.1	15.8	20.6	5.3

Table 1. Average error rates(%) over all 6 samples for all segments. psg_{mis}, psg_{cnf}, pfr_{mis}, and pfr_{cnf} are defined in the text. The value of $pdet$ is 0.5.

to be detected, $mindur$, and the number of modeling/resegmentation iterations.

Segment and frame error rates averaged over all samples are shown in Table 1. The error rates are rather high with a segment miss and confusion rates approximately 24% and 18%, respectively. The duration miss rate is consistent with the segment miss rate but the duration confusion rate is significantly lower than the segment confusion rate. This is because most of the segment confusions are attributable to short segments which, because the process has limited resolution, are not reliably detected. This effect can be seen in the histogram shown in Fig. 4 where the shaded area represents the number of segments which are missed or confused. Most such errors occur for segments with durations less than 2 s.

$mindur$ (secs)	avg. no. of segs	avg. error rates(%)	
		psg_{mis}	psg_{cnf}
0	42.8	25.1	15.8
2	26.8	13.9	3.2
4	15.2	11.0	2.0
6	7.3	8.7	0.0

Table 2. Average error rates(%) over all samples for segments as a function of minimum segment duration, $mindur$, in secs. psg_{mis} and psg_{cnf} are defined in the text. The criterion for segment detectability is $pdet = 0.5$.

Table 2 shows more clearly the effect of segment duration on performance for segment miss and confusion rates. Each row in this table excludes actual segments whose duration is less than the specified value of $mindur$. It can be seen that the segment miss rate is reduced almost 2 to 1 for $mindur$ equal to 2 secs while the segment confusion rate is reduced by 6 to 1.

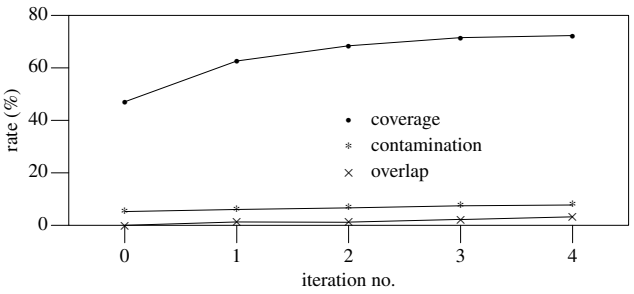

Fig. 5. Clustering performance in terms of coverage, contamination, and overlap (see text for definitions).

Clustering performance in terms of the measurements previously defined in Section 3.2 is shown in Figure 5 as a function of segmentation/modeling iteration. Coverage, p_{cvrg}, contamination, p_{cntm}, and overlap, p_{ovlap}, are averaged over all samples and speakers. Iteration number 0 refers to the initial segmentation while iteration numbers greater than 0 refer to successive modeling and resegmentation stages. It can be seen that all three measurements increase monotonically through successive iterations. Overlap, an *a priori* measurement, is seen to be a reasonable predictor of contamination. Note that the maximum value of coverage, approximately 72%, is consistent with the segment miss rate shown in Table 1.

4. CONCLUSION

We have described and evaluated a process for unsupervised segmentation of speakers in a telephone conversation. Here, unsupervised means that there is no prior information or models for the speakers. Compared to supervised segmentation, where representative models exist for the speakers, the task is a difficult one leading to poorer performance. The process is limited by the amount of coverage obtainable in the initial segmentation and the possibility for contamination of one speaker's model with data from the other. Another serious limitation is the ability of the process to resolve short segments, common in telephone conversation, because of the inherent instability of short analysis windows.

The results described here are obtained from too small a set of conversations to be considered completely reliable. Nevertheless, they do suggest that the performance that is obtained is suitable for the intended application, namely to provide some automated assistance to human labelers to locate the longest segments originating from the customer in a customer/representative conversation. Performance of the system improves significantly for longer segment durations (see Table 2) and the segments do not need to be located with great precision, so that a detectability criterion of $pdet = 0.5$ is adequate.

5. REFERENCES

[1] J-F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, C. Wellekens, A Speaker Tracking System Based on Speaker Turn Detection for NIST Evaluation, *Proc. ICASSP 2000*, IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, Istanbul, 1177-1180, 2000.

[2] R. Dunn, D.A. Reynolds, and T.F. Quatieri, Approaches to speaker detection and tracking in conversational speech, *Digital Signal Processing*, **10**, 93-112, 2000.

[3] H. Gish, M-H. Siu, and R.Rohlicek, Segregation of Speakers for Speech Recognition and Speaker Identification, *Proc. ICASSP 91*. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, Toronto, 873-876, 1991.

[4] J-L. Gauvain, L. Lamel, G. Adda, Partitioning and Transcription of Broadcast News Data, *Proc. of ICSLP98*, 5th Intl. Conf. on Spoken Lang. Processing, Sydney, 1335-1338, 1998.

[5] A. D. Gordon, *Classification, Methods for the exploratory analysis of multivariate data*, Chapman and Hall, London and New York, 1981.

[6] J. Makhoul, F. Kubala, T. Leek, D. Liu, L. Nguyen, R. Schwartz, and A. Srivastava, Speech and Language Technologies for Audio Indexing and Retrieval, *Proc. of the IEEE*, **88**, 1338-1352, August, 2000.

[7] A.E. Rosenberg, I. Magrin-Chagnolleau, S. Parthasarathy, and Q. Huang, Speaker Detection in Broadcast Speech Databases, *Proc. of ICSLP98*, 5th Intl. Conf. on Spoken Lang. Processing, Sydney, 1339-1342, 1998.

[8] M-H. Siu, G. Yu, and H. Gish, An unsupervised, sequential learning algorithm for the segmentation of speech waveforms with multiple speakers, *Proc. ICASSP 92*, IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, San Francisco, vol. II, 189-192.

[9] L. Wilcox, F. Chen, D. Kimber, and V. Balasubramanian, Segmentation of speech using speaker identification, *Proc. ICASSP 94*, IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, Adelaide, 161-164, 1994.