Speaker Segmentation and Clustering in Meetings

Qin Jin and Tanja Schultz

Interactive Systems Laboratories
Carnegie Mellon University, USA
{qjin, tanja}@cs.cmu.edu

Abstract

This paper describes the automatic speaker segmentation and clustering system for natural, multi-speaker meeting conversations based on multiple distant microphones. The system was evaluated in the NIST RT-04S Meeting Recognition Evaluation on the speaker diarization task and achieved speaker diarization performance of 28.17%. This system also aims to provide automatic speech segments and speaker grouping information for speech recognition, a necessary prerequisite for subsequent audio processing. A 44.5% word error rate was achieved for speech recognition.

1. Introduction

In recent years, automatic processing of natural, multi-speaker meeting audio has seen growing interest. This is reflected by the appearance of large meeting corpora from different research groups and the new evaluation paradigm presented by NIST, called Rich Transcription on meetings. Automatic processing of a meeting to generate a full representation of the meeting has been considered as an "AI complete", as well as "ASR complete" problem [1]. It includes issues about transcription, meta data extraction, summarization and so on. Automatic speaker segmentation and clustering is one type of meta information extraction.

NIST started the "Who Spoke When" speaker diarization evaluation (which is the speaker segmentation and clustering task) on telephone conversations and Broadcast News in 2002. However, it is more challenging to segmenting and clustering speakers involved in meetings with overlaps and with distant microphones. Therefore, NIST initiated the same evaluation on meetings in the spring of 2004.

Speaker segmentation and clustering consists of identifying who and when a speaker speaks in a long meeting conversation. Ideally, a speaker segmentation and clustering system will discover how many people are involved in the meeting, and outputs clusters with each corresponding to one speaker. This paper focuses on the automatic speaker segmentation and clustering of meetings based on multiple distant microphones.

The remainder of this paper is organized as follows. In section 2, we briefly describe the overall system overview. Then we explain the speaker segmentation and speaker clustering components in detail in section 3 and section 4 respectively. Section 5 presents our experimental results and conclusions follow in section 6.

2. System overview

This system proceeds in following steps: 1) initial speech/non-speech segmentation on each channel; 2) unification of the initial segmentations across multiple channels; 3) best channel selection for each segment; 4) speaker change detection in long segments; 5) speaker clustering on all segments; 6) smoothing processing.

The **initial speech/non-speech segmentation** is produced based on the acoustic segmentation software CMUseg_0.5. We removed the classification and clustering components and just used it as a segmenter. Detailed description about the algorithms used in this software can be found in [2].

In the **multiple channel unification** step, the segment boundaries are unified across multiple channels. Figure 1 shows an example for two distant microphone channels. The initial segmentation produces two speech segments on channel A, (t2, t3) and (t5, t7); and two segments, (t1, t4) and (t6, t8), on channel B. After unification, the segments across the two channels are (t1, t2), (t2, t3), (t3, t4), (t5, t6), (t6, t7) and (t7, t8).

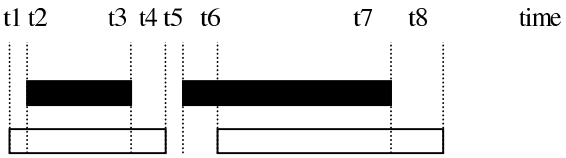

Figure 1: Multiple Channel Unification

We conduct **best channel selection** for each of the segments produced during the unification step. We compute the minimum energy ($MinE_i$), maximum energy ($MaxE_i$), and the signal-to-noise ratio (SNR_i) within one segment on each channel. We select the best channel for each segment according to following criterion:

$$i^* = \arg\min \frac{MinE_i}{MaxE_i} \bullet \frac{1}{SNR_i} \quad (1)$$

Speaker change detection is followed on any segment that is longer than 5 seconds. We choose 5 seconds because this was found to give optimal segmentation accuracy via cross-validation on the development set. **Speaker clustering** is then performed on all the segments. We will discuss the speaker change detection and speaker clustering modules in detail in the following two sections.

In the final **smoothing** step, we merge any two segments that belong to the same speaker and have less than 0.3 seconds gap between them. This is based on our experience in the RT-03S evaluation.

3. Speaker segmentation

For any segment that is longer than 5 seconds, we use a speaker change detection procedure to check whether there exist speaker turn changes that have not been detected. The procedure is shown in Figure 2.

We first compute the distances between two neighboring windows. The window size is one second each and it shifts

every 10ms. The distance between Win1 and Win2 is defined as follows

$$D(Win_1, Win_2) = -\log \frac{P(X_C|\theta_C)}{P(X_A|\theta_A)P(X_B|\theta_B)} \quad (2)$$

where X_A, X_B, and X_C are the set of feature vectors in Win1, Win2, and the window which concatenates Win1 and Win2. θ_A, θ_B, and θ_C are statistical models built on X_A, X_B, and X_C respectively. We can see from (2) that the larger the distance is, the more possible a speaker turn change exists.

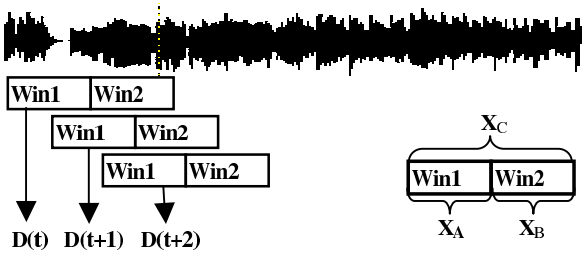

Figure 2: Speaker Change Detection

We assume a speaker turn change exists if the local maximal of distances satisfy (3).

$$\begin{aligned} &D_{\max} - D^L_{\min} > \alpha \\ &D_{\max} - D^R_{\min} > \alpha \\ &Min\left(|I_{\max} - I_{\min}|^L_R \right) > \beta \end{aligned} \quad (3)$$

where $D_{\max}$ refers to the local maximal distance value and $D^L_{\min}$ and $D^R_{\min}$ refer to the left and right local minimal distance values around the local maximum. $I_{\max}$ refers to the index of the local maximum. The third item in (3) means that we not only consider the value of the local maximum but also its shape. α and β are constant thresholds. We found the optimal values for them via cross-validation on the development set. α equals to the variance of the all the distance values times a factor of 0.5. β is set to be 5. Our approach differs from other approaches such as in [3, 4] in the sense that, in our implementation, we build a Tied GMM (TGMM) using the entire speech segments and generate a GMM for each segment via adapting the TGMM. The advantage is that a more reliable model can be estimated with Tied GMM.

4. Speaker clustering

Fro speaker clustering, we use a hierarchical, agglomerative clustering technique TGMM-GLR. We first train a TGMM θ based on the entire speech segments. Adapting θ on one segment generates the GMM for that segment. The definition of the GLR distance between two segments is the same as in (2). A symmetric distance matrix is built by computing the pairwise distances between any two segments. At each clustering step, the two segments, which have the smallest distance, are merged, and the distance matrix is updated. We use the Bayesian Information Criterion as stopping criterion.

4.1. Bayesian Information Criterion

The Bayesian Information Criterion (BIC) is a model selection criterion widely used in statistics. It was introduced for speaker clustering in [3]. The Bayesian Information Criterion states that the quality of model M to represent data $\{x_1, \cdots, x_N\}$ is given by

$$BIC(M) = \log L(x_1, \cdots, x_N | M) - \frac{\lambda}{2} v(M) \log N \quad (4)$$

with $L(x_1, \cdots, x_N | M)$ representing the likelihood and $v(M)$ representing the complexity of model M, which equals to the free model parameters. Theoretically, λ should be equal to 1, but it is a tunable parameter in practice.

The question if there is a speaker change at point i in data $X = \{x_1, \cdots, x_N\}$ can be converted into a model selection problem. The two alternative models are: 1) model M_1 assumes that X is generated by a multi-Gaussian process, that is $\{x_1, \cdots, x_N\} \sim N(\mu, \Sigma)$, 2) model M_2 assumes that X is generated by two multi-Gaussian processes, that is $\{x_1, \cdots, x_i\} \sim N(\mu_1, \Sigma_1)$, $\{x_{i+1}, \cdots, x_N\} \sim N(\mu_2, \Sigma_2)$. The BIC values for the two models are:

$$\begin{aligned} BIC(M_1) &= \log L(x_1, \cdots, x_N | \mu, \Sigma) - \frac{\lambda}{2} v(M_1) \log N \\ BIC(M_2) &= \log L(x_1, \cdots, x_i | \mu_1, \Sigma_1) \\ &\quad + \log L(x_{i+1}, \cdots, x_N | \mu_2, \Sigma_2) \\ &\quad - \frac{\lambda}{2} v(M_2) \log N \end{aligned}$$

The difference between the two BIC values is:

$$\begin{aligned} \Delta BIC &= BIC(M_1) - BIC(M_2) \\ &= \log \frac{L(x_1, \cdots, x_N | \mu, \Sigma)}{L(x_1, \cdots, x_i | \mu_1, \Sigma_1) L(x_{i+1}, \cdots, x_N | \mu_2, \Sigma_2)} \\ &\quad + \frac{\lambda}{2} [v(M_2) - v(M_1)] \log N \end{aligned} \quad (5)$$

If the value of ΔBIC is negative, it claims that model M_2 fits the data better, which means that there is a speaker change at point i. Therefore, when the value of ΔBIC for the two closest segments (candidates for merging) is negative, we stop the clustering process.

5. Experimental results

5.1. Data

All experiments throughout this paper were conducted on the RT-04S meeting data. Each meeting was recorded with personal microphones for each participant (close-talking microphones), as well as room microphones (distant microphones) placed on the conference table. In this paper we

focus on the task of automatic speaker segmentation and clustering based on the distant microphone channels only.

Both the development and the evaluation datasets from the NIST RT-04S evaluation were used. The data were collected at four different sites, including CMU, ICSI, LDC, and NIST [5, 6, 7, 8]. The development dataset consists of 8 meetings, two per site. 10-minute excerpts of each meeting were transcribed. The evaluation dataset also consists of 8 meetings, two per site. 11-minute excerpts of each meeting were selected for testing. All of the acoustic data used in this work is of 16kHz, 16-bit quality. Table 1 gives a detailed description of the RT-04S development dataset, on which we report detailed performance numbers.

Table 1: Development Dataset

Meeting ID (abbreviation)	# Spkrs	# distMic
CMU_20020319-1400 (CMU1)	6	1
CMU_20020320-1500 (CMU2)	4	1
ICSI_20010208-1430 (ICSI1)	7	4
ICSI_20010322-1450 (ICSI2)	7	4
LDC_20011116-1400 (LDC1)	3	8
LDC_20011116-1500 (LDC2)	3	8
NIST_20020214-1148 (NIST1)	6	7
NIST_20020305-1007 (NIST2)	7	6

Note that we ignored the "PDA" low quality channels in the ICSI meetings.

5.2. Speaker Segmentation Performance

Good speaker segmentation should provide the correct speaker changes, as result, each segment should contain exactly one speaker. There are two types of errors related to speaker change detection: insertion error (when a speaker change is detected but it does not exist in reference) and deletion error (an existing speaker change is not detected). These two types of errors have different impact depending upon the application. In our system, the segmentation stage is followed by a clustering stage. Therefore, insertion errors (resulting in an over segmentation) are less critical than deletion errors, since the clustering procedure has the opportunity to correct the insertion errors by grouping the segments related to the same speaker. While deletion errors cannot be recovered in the clustering stage.

A reference of speaker change is required for analyzing these errors. The reference was generated from the manual transcription of a meeting. However, the exact speaker change point is not very accurate in the reference, since the perception of speaker changes is very subjective. Therefore, we define an accuracy window around the reference speaker change point, following [9] it is set as 1 second. For example, If N_r and N_h are reference and hypothesized speaker change points respectively, they are mapped to one-another and we call the hypothesis N_h is a hit if 1) N_h is the hypothesized change point closest to N_r and 2) N_r is the reference change point closest to N_h and 3) the distance between N_r and N_h is less than 1 second. From the formed mapping between reference and hypothesis, we can determine the precision (percentage of hit among all the hypothesized change points) and recall (percentage of hit among all the reference change points). Deletion errors will directly lower the recall. Insertion errors will reduce the precision. Generally we seek systems that exhibit both high recall and precision. However, as mentioned previously, deletion errors are more critical than insertion errors, we care more about the recall value.

Table 2: Segmentation Performance

	Precision	Recall
Initial	86.83%	11.60%
Unification	87.74%	19.00%
Turn Detection	85.17%	76.41%

Table 2 shows the speaker segmentation performance at different system steps. Not surprisingly, the low recall of the initial segmentation indicates high deletion errors, which means a lot of speaker changes are missed. Multiple channel unification compensates a little for the deletion errors. Speaker change detection obtained the big gain on the recall while only suffering little precision decrease.

5.3. Speaker Diarization Performance

We use the standard performance measurement, speaker diarization error, for speaker segmentation and clustering which was used in the NIST RT-03S evaluation [10]. The overall speaker segmentation and clustering performance can be expressed in terms of the miss (speaker in reference but not in system hypothesis), false alarm (speaker in system hypothesis but not in reference), and speaker error (mapped reference speaker is not the same as the hypothesized speaker) rates. The speaker diarization score is the sum of these three components and can be calculated using this formula:

$$DiaErr = \frac{\sum_{allS} \{dur(S) * (\max(N_{ref}(S), N_{sys}(S)) - N_{correct}(S))\}}{\sum_{allS} \{dur(S) * N_{ref}(S)\}}$$

where $DiaErr$ is the overall speaker diarization error, $dur(S)$ is the duration of the segment, $N_{ref}(S)$ is number of reference speakers in the segment, $N_{sys}(S)$ is the number of system speakers in the segment, $N_{correct}(S)$ is the number of reference speakers in the segment for whom their mapped system speakers are also in the segment. This formula allows the whole file to be evaluated, including regions of overlapping speech.

Table 3: Speaker Diarization Performance (in %)

	Development Set		Evaluation Set	
	Overlap	No overlap	Overlap	No overlap
Miss	8.7	0.0	19.8	0.4
FA	3.3	2.9	2.6	4.1
SpkrErr	25.1	26.7	17.8	23.4
DiaErr	**37.11**	**29.59**	**40.19**	**28.17**

Table 3 shows the overall speaker diarization performance on the development set and the evaluation set under the condition of including the regions of overlapping speech and excluding the regions of overlapping speech. Comparable results are achieved on both datasets. The dominant error among the three error components is speaker error.

In table 4, we show the speaker diarization performance on individual meetings of the development set. The results exhibit large variability over meetings collected at different sites. We think that this variability may be due to unquantified meeting characteristics such as overall degree of crosstalk, general meeting geometry including room acoustics and microphone variability within a meeting. However, we noticed that there is a general trend that our system usually under-estimates the number of speakers involved in a meeting. Although, on meetings CMU2 and NIST1, the system under-estimates the number of speakers, it still achieves better performance compared to most other meetings. This is due to the fact that both these two meetings have a dominant speaker who talks more than 70% of the time during the whole meeting.

Table 4: Individual Speaker Diarization Performance on dev set including overlapping speech (in %)

Meeting	Miss	FA	SpkrErr	DiaErr	#ref	#sys
CMU1	12.6	4.3	30.3	47.12	6	4
CMU2	3.4	5.0	16.3	24.72	4	2
ICSI1	4.7	2.9	35.0	42.62	7	4
ICSI2	9.8	1.1	37.0	47.92	7	3
LDC1	6.2	2.6	9.0	17.78	3	3
LDC2	17.3	1.1	11.0	29.41	3	3
NIST1	7.2	7.1	11.7	26.01	6	2
NIST2	6.5	3.1	49.5	59.04	7	2

We conducted an interesting experiment. We assume a one-to-one mapping between channel and speaker. We use the best channel information only, which was provided in the channel selection step described in section 2. We did not do speaker clustering. For any two segments, if the channel selection process produces the same best channel for them, we assume these two segments belong to the same speaker. We got 55.45% and 52.23% speaker diarization errors under the condition of including and excluding overlapping speech. This indicates that there is rich information that can be used to help speaker segmentation and clustering from the multi-channel recordings. Our current system utilizes such information implicitly by doing best channel selection. In the future work, we will explore more efficiently using the information provided by multi-channel recordings, such as timing information which relates to the speaker locations.

5.4. Speech Recognition Performance

Speech recognition system achieved a 44.5% word error rate on the evaluation set when using segments provided by our system, refer to [11] for detail. We have noticed that speech recognition has different requirement for speaker segmentation and clustering. In speech recognition, the goal of speaker segmentation and clustering is to serve speaker adaptation. Speaker adaptation concerns more about regression of speakers, not strict classification of speakers. So if two speakers sound similar, they can be considered as equal and grouped into one cluster. It actually would be rather desirable for speech recognition to group similar speakers together, so that it can get enough adaptation speech. Therefore, a specific speaker segmentation and clustering system tuned in favor of speech recognition may achieve better word error rate although the speaker diarization performance might get worse.

6. Conclusions

We described our automatic speaker segmentation and clustering system for natural, multi-speaker meeting conversations based on multiple distant microphones. The performed experiments show that the system is capable of providing useful speaker information on a wide range of meetings. The system achieved 28.17% speaker diarization score in the NIST RT-04S evaluation. The speech recognition system achieved the performance of 44.5% word error rate when using segments provide by this system in RT-04S.

7. Acknowledgement

We would like to thank Michael Dambier for providing the best channel selection scripts, Hua Yu for helping with using CMUseg-0.5 as a segmenter, Florian Metze and Christian Fügen for providing feedback about segmentation impact on speech recognition and useful discussion.

8. References

[1] N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke, "The Meeting Project at ICSI," *HLT*, San Diego, March 2001.

[2] M. Siegler, U. Jain, B. Raj, and R. Stern, "Automatic Segmentation, Classification and Clustering of Broadcast News Audio," *Proc. DARPA Speech Recognition Workshop*, Chantilly, Virginia, 1997.

[3] S.S. Chen and P.S. Gopalakrishnan, "Clustering via the Bayesian Information Criterion with Applications in Speech Recognition," *ICASSP*, 1998.

[4] P. Delacourt and C.J. Wellekens, "DISTBIC: A Speaker-based Segmentation for Audio Data Indexing," *Speech Communications*, 32, 111-126, 2000.

[5] S. Strassel, M. Glenn, "Shared Linguistic Resources for Human Language Technology in the Meeting Domain," *Proc. NIST Meeting Recognition Workshop*, Montreal, Canada, 2004.

[6] S. Burger, Z. Sloane, "The ISL Meeting Corpus: Categorical Features of Communicative Group Interactions," *NIST Meeting Recognition Workshop*, Montreal, Canada, 2004.

[7] A. Janin, J. Ang, S. Bhagat, R. Dhillon, J. Edwards, N. Morgan, B. Peskin, E. Shriberg, A. Stolcke, C. Wooters, B. Wrede, "The ICSI Meeting Project: Resources and Research," *NIST Meeting Recognition Workshop*, Montreal, Canada, 2004.

[8] V. Stanford, J. Garofolo, "Beyond Close-talk – Issues in Distant speech Acquisition, Conditioning Classification, and Recognition," *NIST Meeting Recognition Workshop*, Montreal, Canada, 2004.

[9] A. Vandecatseye, J. Martens, "A Fast, Accurate Stream-based Speaker Segmentation and Clustering Algorithm," *Eurospeech*, Geneva, Switzerland, 2003.

[10] The Rich Transcription Spring 2003 Evaluation Plan, http://www.nist.gov/speech/tests/rt/rt2003/spring/docs/rt03-spring-eval-plan-v4.pdf, 2003.

[11] F. Metze, Q. Jin, C. Fügen, K. Laskowski, Y. Pan and T. Schultz, "The ISL Meeting Transcription System," *NIST Meeting Recognition Workshop*, Montreal, Canada, 2004.