Active Learning Based Constrained Clustering For Speaker Diarization

Chengzhu Yu, *Student Member, IEEE,* and John H. L. Hansen, *Fellow, IEEE*

***Abstract*—Most speaker diarization researches focus on unsupervised scenarios, where no human supervision is available,. However, in many real-world applications, certain amount of human input could always be engaged, especially when minimal human supervision could brings significant performance improvement. Motivated by this, we propose an active learning based speaker clustering algorithm, to effectively improve speaker diarization performance with limited human input. Specifically, proposed algorithm has two different active learning components: *explore* and *consolidate*, acting in initial stages of bottom-up speaker clustering. While the concept of *explore* and *consolidate* are borrowed from the area of data mining, a substantial changes are made here for the application of speaker diarization. The main purpose of *explore* and *consolidate* stages in proposed algorithm, is to boost the speaker clustering process with reliably estimated initial speaker clusters. To achieve this, we uses farthest first query search (FFQS) with active learning to quickly discover at least one sample for each speaker during *explore* phase, and employ nearest neighbor query search (NNQS) during *consolidate* phase to ensure reliable instances for each discovered speaker cluster. After *explore* and *consolidate* phases, the standard bottom-up clustering is performed with a constraint that the clusters discovered during *explore* phases are not merged with each other. Finally, we also propose an active learning based cluster reassignment approach, where clustered segments that accounted for largest expected speaker error are chosen for human evaluation. We evaluate the proposed algorithm on a subset of Apollo multi-track corpus and AMI meeting corpus. The results indicate significant improvement of speaker diarization performance with relatively small amount of human supervision.**

***Index Terms*—Speaker diarization, active learning, bottom-up clustering**

I. Introduction

Speaker diarization is the process of automatically detecting *who spoke when* in an audio sequence. With increasing amount of audio resources, speaker diarization becomes an important technology in many applications such as information retrieval, metadata extractions, meeting annotations, and conversation analysis. Recent developments for speaker diarization algorithms have been largely driven by Rich Transcription (RT), where speaker diarization plays a role to provide speaker index and other information to achieve improved speech-to-text transcriptions.

As a sequential process, speaker diarization normally involves several components such as voice activity detection, speaker change detection (segmentation), clustering, and re-segmentation. Among theses components, the core of speaker diarization is clustering, where it separate segments from different audio sources such as speaker, music, and noise, and group them together. Due to its significance in speaker diarization, various solutions have been proposed for speaker clustering. Theses include but not limited to bottom-up approach, also known as agglomerative hierarchical clustering (AHC), top-down approach, and recently proposed global optimization approaches.

Bottom-up clustering is in general the most popular strategy among various clustering solutions. It starts by treating each individual segments, obtained in the segmentation stage, as separate clusters, and iteratively merging the closest two clusters until a specified stopping criteria is satisfied. While not as popular as its counterpart, top-down approach has also been applied widely and some studies have reported that it could achieve comparable result as bottom-up clustering [Evans, 2012]. Different from bottom-up based approach, top-down clustering starts from modeling the entire audio as single model and iteratively splitting the model into sub clusters until a stopping criteria is met. Despite their differences, both bottom-up and top-down based approach are iterative process and has the drawback of error propagation. A recently proposed clustering algorithm, Interger Linear Programming (ILP), attempts to overcome this drawback by finding the cluster assignments that jointly minimizing the number of assigned clusters as well as within-cluster dispersion. While the ILP could avoid the drawbacks of error propagation, it has to start with initial clusters that contains sufficient samples for estimating its characteristic (e.g., i-vector). Therefore, ILP is mostly performed on top of a bottom-up clustering.

Along with the development in speaker clustering, the distances metric for performing speaker clustering has also made significant improvement from original Bayesian informative criteria (BIC), generalized log-likelihood ration (GLR), Kullback-Leibler (KL) divergence, to recent i-vector based distances such as cosine distance score (CDS) as well as probablistic linear discriminant analysis (PLDA) based distance. Other alternative distance metrics, such as these based on information theoretic framework are also proposed and showed competitive results.

Despite the success of recent improvement on speaker clustering algorithms, distance computations, as well as other non-trivial components, speaker diarization still remained as challenging tasks in many real-word applications, especially where the audio quality is suboptimal or the speech communications comprises large proportions of fast speaker turns. For example, diarization of telephone conversations are notably more challenging compared with broadcast news diarization, as well as meeting room diarization.

Due to the limitation within current speaker diarization systems using exclusively audio/speech information, a number of recent studies have proposed to exploit auxilary infor-

mations for improved speaker diarization performance. For example, the linguist information such as speaker name occurring patterns are extracted from the speech transcripts to provide additional information during speaker clustering. The speech transcript could be obtained from manual transcript as well as automatic speech recognition system. Another important supplementary information that presents in many speaker diarization applications is the visual information. The audio-visual speaker diarization has also been studied.

In this study, we propose an active learning based, bottom-up speaker clustering algorithm that effectively utilize human input to improve speaker diarization performance. Our proposed algorithm is based on an assumption that human input could be engaged during speaker diarization process in certain applications. This scenario is especially plausible if small amount of human engagement could brings significant performance improvements. Another assumption we made in this paper is that, human is better than machine for recognizing whether a given segment pair is from the same speaker or not. This assumption is based on the results from previous studies that while current speaker recognition systems shows competitive performance as human in clean speech condition, in adverse conditions human significantly outperform machine. Besides, many audio streams for speaker diarization applications contain higher level information such as video, spoken names, and contextual informations that could effectively employed by human for determining whether the two segments are from the same speaker or not.

The format of query for human to provide ground truth in speaker diarization, is *yes or no* type of question asking whether a given segments pair belongs to the same speaker or not. The total number of queries for obtaining perfect clustering results requires $O(N^2)$, where N indicates the number of segments. Therefore, an effective active query selection strategy is necessary in order to effectively employ human input to boosting the speaker diarization performance.

To effectively employ human input, we first need to identify which part of speaker clustering components has most effect on final speaker diarization performance. In other words, involving human input in which part of current speaker clustering algorithm, could most effectively improve the overall speaker diarization performance. A recent study by [Simon King] has evaluated several key components of speaker diarization and concluded that initialize speaker models with pure and reliably labeled data could lead to significant improvement to the overall speaker diarization performance. Motivated by the above findings, we designed our active learning algorithm to quickly discover all or most of the speakers in audio stream in the *explore* phase, and ensure sufficient and reliable samples for each cluster to during *consolidate* phase. The initial speaker models for trained after explore and consolidate phases are much reliable than those obtained in fully unsupervised manner, and are expected to lift the speaker clustering process.

In addition to active learning for improved speaker clustering process, we also investigate the use of active learning for cluster reassignment approach. The objective of proposed cluster reassignment approach, is to actively select certain speech segments from clustered result for human evaluation/correction. The key of active learning based cluster reassignment, is to effectively find most informative segments, from current clustering results. In this study, we find speech segments with largest expected speaker error difference as a candidates for human evaluation.

To summary, in this study, we investigate the use of active learning for bottom-up clustering stage of speaker diarization. We proposed two different strategies where active learning are employed for robust initial cluster estimation and post clustering reassignment, respectively. The remainder of paper is organized as follows.

II. Related Work

Active learning based constrained clustering has been extensively studied for image clustering tasks and in broader area of data mining. A number of active learning algorithms have been proposed to use active learning for clustering unlabeled data with human in the loop. The key idea behind these algorithms is to actively select pairs of data for human to provide answer in a form of yes or no. Most of the algorithms proposed in these studies, are targeting k-means clustering, and normally composed of two stage: *explore* and *consolidate*. The purpose *explore* phase is to find the centroids of different clusters, and the aim of consolidate stage is to locate most informative data pair for human labeling. While the fundamental problem of these studies is much similar to the problem we have in speaker diarization, active learning for speaker diarization is more challenging task in general due to the two reasons. Firstly, the bottom-up clustering in speaker diarization requires to iteratively clustering two closest segments and update the cluster statistics after each iteration. Therefore, the decisions in current iteration has reliance on previous iteration, and therefor it is difficult to quantify the importance of each segment pair. Another important difference is that, the total number of cluster numbers in speaker diarization are unknown most of time. Due to these differences, the direct replication of active learning algorithms developed in these studies, is not viable.

On the other hand, only a limited number of studies in the area of speaker recognition and diarization, has investigates the use of active learning for speaker diarization tasks. For example, the study by [Shum] has investigate to use active learning to obtain background speaker labels from unlabeled data for training PLDA system. While the study by [Shum] bears similarity with this paper, but the ultimate goal of [Shum] is to locate reliable samples sufficient enough to train PLDA system, rather than clustering entire data as in speaker diairization task. Another study, that employ active learning for speaker diarization is by [Mateusz], where the criteria of active selection for human labeling is simply based on the length of speech, which will not work in many speaker diarization task where the variance of segment length is small.

III. Baseline System

The baseline speaker diarization used in this study, is a bottom-up speech clustering algorithm with i-vector cosine

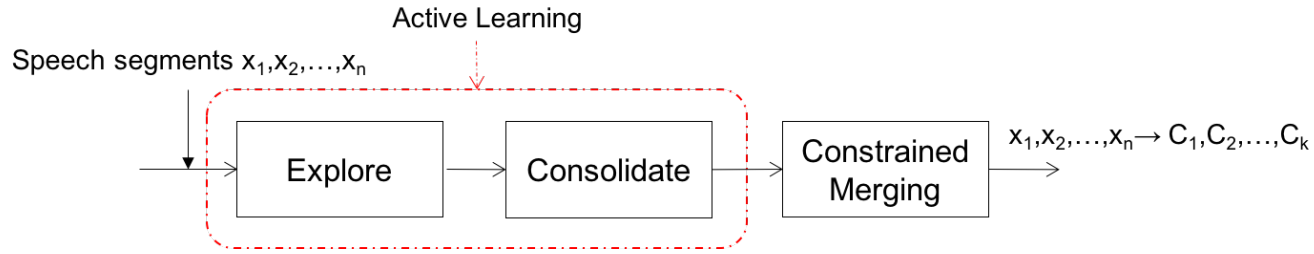

Fig. 1. A boat.

distances score as distance metric between two speech segments. In this section, we will give a brief discussion about i-vector extraction, cosine distance score (CDS) and bottom-up clustering approach.

A. i-vector extraction

In i-vector extraction framework, speaker and channel dependent GMM supervector is modeled as follows:

$$M = m + Tw, \tag{1}$$

where m is the supervector obtained from the universal background model (UBM), T is the low rank total variability matrix representing the basis of reduced total variability space, and w is the low rank factor loadings referred to as i-Vectors.

The estimation of the total variability matrix T employs expectation maximization (EM) method as described in [1]. After training the total variability matrix, the i-Vector of given speech utterance is extracted as the conditional expectation of i-Vector distribution given observation features.

$$w_s^* = E[P(w_s|X_s)], \tag{2}$$

where w_s^* is the i-Vector of the given speech utterance s, X_s is the clean observation features, $P(w_s|X_s)$ is the conditional distribution of the i-Vector given observation features, and $E[\cdot]$ indicates the expectation. Finally, the i-Vector of the given speech utterance can be represented using the Baum-Welch zeroth (N_s) and centralized first (F_s) order statistics,

$$w_s^* = (T' N_s \Sigma^{-1} T + I)^{-1} T \Sigma^{-1} F_s, \tag{3}$$

where Σ is the covariance matrix obtained from UBM model and I is the identity matrix.

B. Cosine Distance Score

Comparison i-vectors from two different segments, could be successfully achieved with simple cosine similarity. The cosine distance score between two vectors could be expressed as follow:

$$score(w_i, w_j) = \frac{w_i^T \cdot w_j}{||w_i|| \cdot ||w_j||}. \tag{4}$$

Note that, the score of cosine distance ranges between [-1, 1]. The higher the number towards 1, the more similarity exist between two vectors. The cosine distance score has been a popular metric for speaker recognition and verification in i-vector space.

C. Bottom-up Speaker Clustering

Bottom-up clustering, also known as hierarchical agglomerative clustering, has been the most popular speaker clustering algorithm used for speaker diarization. It typically start by treating all homogeneous segments as a separate cluster, and iteratively merge two closest clusters. In our study, for each iteration, we find two segments that has the highest cosine similarity, and merge them into a single cluster. After each iteration, i-vectors of all segments are normalized to have zero mean. We continue the iteration until CDS of two closest segment reached stopping criteria.

IV. Active Learning for Cluster Initialization

Due to the error propagation characteristics of bottom-up speaker clustering, having good initial cluster model, with pure speaker segments in each clustering, has significant impact on the final speaker diarization performance. In this section, we will give a detailed description of our proposed active learning strategy for cluster initialization. As mentioned in Introduction, the proposed algorithm has two active learning components: *explore* and *consolidate*.

A. Explore

The purpose of explore phase, is to quickly discover all the speaker clusters within the audio streams, finding at least one speech segment for each speaker cluster. To achieve this, we use the farthest first query search (FFQS) proposed by [Basu]. Specifically, a speech segment is randomly selected from all speech segments to be used as seed segment. The selected speech segment is then used to initialize the first cluster. After creating the first speaker cluster, the next segment is selected which is farthest from existing clusters. The chosen segment is then provided for human to provide expert opinion. If the chosen segment belongs to existing clusters after pairwise comparison, the new segment is merged to the corresponding cluster. Otherwise, a new cluster is created from the selected segment. Note that, in order to decided if a given speech segment belongs to a target cluster or not, we compose a query pair using the segment in question and the longest segment within the target cluster. If the answer returns by human expert is true, then the segment belongs to the target cluster, and vice versa. The FFQS process continues until the pairwise comparison operations reached specified maximum number specified by user.

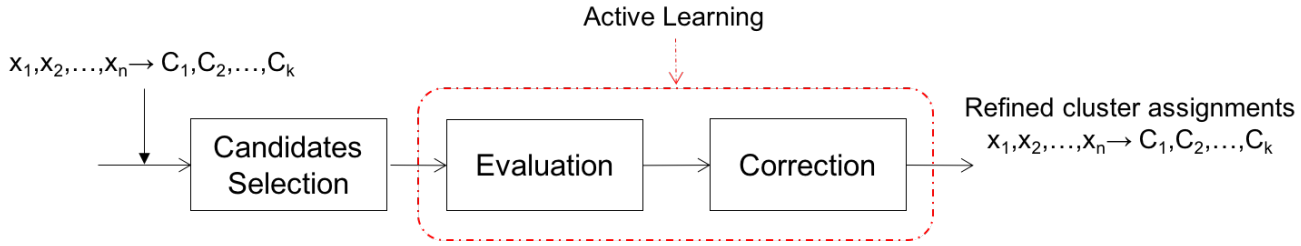

Fig. 2. A boat.

Data: Set of speech segments $X = \{x_i\}_{i=1}^{n}$, acess to the answers of pairwise queries, maximum number queries Q user specified.
Result: $C_{k=1}^{k}$ initialized clusters
Start from null cluster $C = \{\}$;
Select a segment x at random, and create the first cluster as $C_1 = \{x_i\}$, $\lambda \leftarrow 1$;
while *maximum queries not reached,* $\lambda < Q$ **do**
  Find speech segment x_λ farthest from existing clusters in C, based on i-vector cosine similarity scores.;
  if x_λ *belongs to any clusters in* C **then**
    Add speech segment x_λ to cluster matching cluster
  else
    Create new cluster C_k with speech segment x_λ;
  end
  Increase λ, for each query access.
end

Algorithm 1: FFQS with random seed during *explore* phase.

While the above algorithms could effectively explore the speaker clusters in the audio streams, its performance varies a lot depending on which seed segment is selected. This problem is due to the randomness during seed selection. The previous studies in k-means clustering has revealed that, the initial seed data should be closer to actual centroids in order to achieve favorable clustering results. Motivated by this, we first performed a fully unsupervised bottom-up clustering on the entire speech segments, and taking the centroid segment of each cluster as initial active selection candidates to start FFQS algorithm during *explore* phase.

B. Consolidate

The FFQS during *explore* stage is very effective in discovering all involved speaker with given audio stream. However, it is not effective, when we need to quickly gather enough instances for each cluster to obtain reliable speaker models. For this purpose, for each discovered clusters obtained from *explore* stage, we uses a nearest neighbor query search (NNQS) approach, to select speech segments closest to it for human evaluation.

C. Constrained Merging

After *explore* and *consolidate* phases, the standard bottom-up clustering is performed with two important exceptions. First, the clusters $C_{k=1}^{k}$ created during *explore* and *consolidate* stages, are restricted to not merge with each other. Second, the stopping distance threshold during bottom-up clustering is no longer necessary, as we have assumed all involved speakers are discovered during *explore* phase. The bottom-up clustering will continues until only $C_{k=1}^{k}$ clusters remain.

V. Active Learning for Cluster Reassignment

In previous section, we propose an active learning based algorithm to obtain better initial speaker models. Alternatively, human input could be involved after clustering finishes, to evaluate and fix incorrectly assigned clusters. This is quite similar to the use of active learning for automatic speech recognition (ASR), where the sentence with less confidence is passed to human for correction. However, the same algorithm used for ASR could not be directly applied in speaker diarization for several reasons. First, the confidence measure used for ASR is not appropriate for speaker diarization. Second, assuming we have identified potential erroneous segments, the human evaluation for evaluate and reassign these segments are much more complex than human transcription in ASR.

The proposed active learning based cluster reassignment has three major components that we will explain in this section.

Data: $C_{k=1}^{k}$ initialized clusters, set of speech segments $X = \{x_i\}_{i=1}^{n}$, acess to the answers of pairwise queries, maximum number queries per cluster Q_k user specified.
Result: $C_{k=1}^{k}$ with more instances.
for *each cluster* C_k *in* $C_{k=1}^{k}$ **do**
  while *maximum queries per cluster* Q_k *not reached* **do**
    Find speech segment **x** closest from C_k, based on i-vector cosine similarity scores.;
    if ***x*** *belongs to any clusters in* C_k **then**
      Add speech segment **x** to cluster C_k
    else
      Exlude this segment **x** from search in next iteration.
    end
  end
end

Algorithm 2: NNQS algorithm during *consolidate* phase.

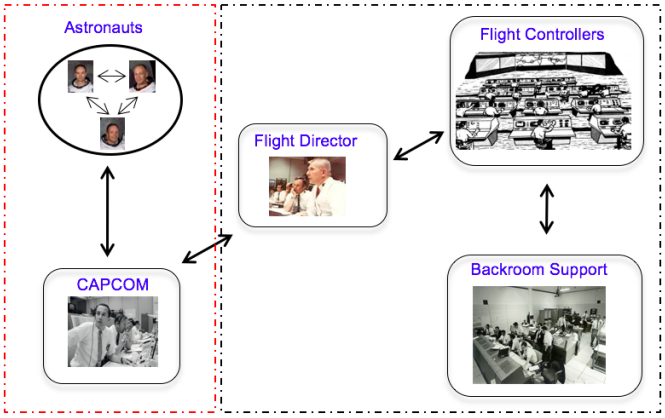

Fig. 3. A boat.

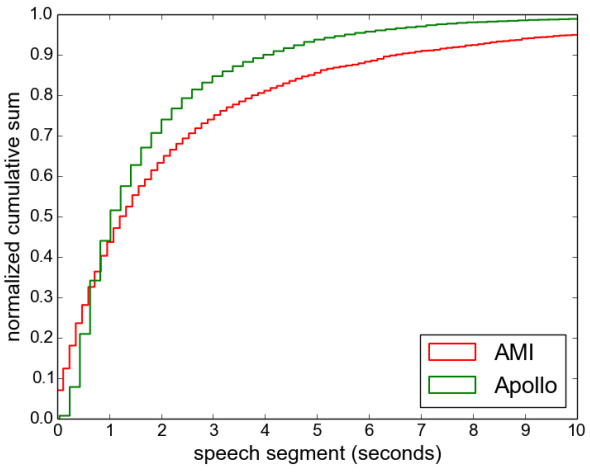

Fig. 4. A boat.

A. Candidates selection

The cluster reassignment algorithm starts with selecting the candidate speech segments and its assigned cluster for human expert to review later. In order to effectively select most informative segments, we ranks all speech segments in terms of expected speaker error reduction. In other words, we will select the speech segments that will produce largest expected speaker error reduction. The expected speaker error for each speech segment could be computed as below.

$$\begin{aligned} E(x_j) =& P(x_j|C_j) \cdot J_{x_j \in C_j} \\ &+ (1 - P(x_j|C_j)) \cdot J_{x_j \notin C_j} \end{aligned} \tag{5}$$

where x_j indicates jth speech segment, C_j is the cluster assigned to speech segment x_j, $P(x_j|C_j)$ is the probability of segment x_j belongs to cluster c_j, $J_{x_j \in C_j}$ is the speaker error if x_j belongs to cluster C_j, and $J_{x_j \notin C_j}$ is the speaker error if x_j not belongs to cluster C_j. We could also write that

$$\begin{aligned} J_{x_j \in C_j} &= 0 \\ J_{x_j \notin C_j} &= \frac{d_j}{\sum_{i=1}^{n} d_i} \end{aligned} \tag{6}$$

where d_j is the length of speech segment x_j, and $\sum_{i=1}^{n} d_i$ is total length sum of all speech segments of the testing audio stream.

We compute $P(x_j|C_j)$ by modeling a multivariate Gaussian distribution for each $C_{k=1}^{k}$ using i-vectors of all the segments of given cluster. We normalize the probability to sum to one.

$$\sum_{i=1}^{k} P(x_j|C_i) = 1; \tag{7}$$

B. Evaluation

After selecting the candidate segments, human expert will determine whether the given segment belongs to its assigned cluster. The process of employing human input to decide whether a segment belongs to a cluster, is more difficult than the strategy we used in *explore* and *consolidate* phases of previous section. We could not compose query pair with the segment in question and a single longest segment in the cluster. This is due to the fact that the chosen longest segment of the cluster, could also be an incorrect assignment. Therefore, we employ a majority voting based segment cluster evaluation strategy. Under this strategy, the segment in question is paired with each segments within assigned cluster, resulting in multiple query pairs. If majority answers of these pairs are true (two segments belong to the same cluster), we will make a decision that the given speech segment has correct cluster assignment, and vice versa.

While the above strategy is robust for evaluating whether a given segment belongs to target cluster, it involves significant number of query pair for human evaluation. One heuristic that we used in our study to effectively reduce the query pair number is to set a maximum query number limit V for each segment evaluation. We rank each segments with the target cluster by its confidence $P(x|c)$, and select top V confident segments for composing pairs with test segment.

C. Correction

After detecting segments with incorrect cluster assignment in *evaluation* stage, we need find the correct cluster designation for these segments. To achieve, we employ N-best cluster evaluation. We find the N most possible cluster candidates for given segment by ranking the i-vector Gaussian posterior probabilities $P(x|C)$. The human expert will evaluate in order whether the given segment belongs to any of these N clusters, using the majority voting scheme as in the *evaluation* stage.

VI. Test Data

We performs experiments on two different speech database: Apollo-11 Mission Control Center (MCC) audio corpus and AMI meeting corpus.

A. Apollo-11 MCC Audio Corpus

During the NASA Apollo mission, all communications between astronauts, flight controllers inside mission control center (MCC), and their backroom support teams are continuously recorded using a 30-track analog reel-to-reel recording

machine. During each mission, a total of 60 audio channels are simultaneously recorded including the voices from more than hundred of different participants of the mission. The University of Texas at Dallas (UTD), University of Maryland College Park (UMD), and Johnson Space Center(JSC) has combined the effort to digitize this data resource and have generated up to 19,000 hours of audio data from various missions of both Apollo and Gemini missions. Those mission audio records the full detail of the Apollo communication, therefore extremely attractive for learning human-to-human communications, group interaction, as well as developing robust speech system.

Moreover, as the speech community (including us) rely on labeled audio data to perform scientific research as well as algorithmic development, we have prepared a 'Task Specific' corpus based on a subset of Apollo-11 audio recordings. We performed our experiment on this subset of Apollo-11 audio recordings includes 3 synchronized channels (Flight director loop, EECOM loop, GNC) spanning 10-hours before, and after lunar landing. This initial 30 hour task corpus has been transcribed to have speaker labels by well-trained speech science students from UTDallas [1].

The audios in Apollo-11 MCC audio corpus includes two types of communication: space-to-ground communications between astronauts and CAPCOM, and ground communication between flight directors and "backroom" support staff. Most of the audios are recorded with close-talking microphones and the audios are in general good quality. The audios from astronauts transmitted through Earth's dedicated telephone channels to Houston from ground stations where the signal was received. The flight directors as well as their "backroom" supports' voice are recorded through intercom circuit called "loops". Each flight director has their own loops, which records the entire communication within that channel.

The voice communication style withing Apollo mission control center is much different from both traditional meeting corpus, including many short speech segments which was intended to improve communication efficiency.

B. AMI Meeting Dataset

We also evaluate proposed algorithms in the popular 12-meeting subset of Augmented MultiParty Interaction (AMI) corpus. This is approximately 5.4 hours of data with each session varying between 15-30 minutes. The AMI corpus contains both audio and visual data, and we uses only the audio data recorded with headset microphones in our experiments. The corpus represents a natural meeting scenarios. A total of four participants are involved in each session, discussing about the task to design a new new remote control device.

VII. Experiments and Results

In this section, we will perform experiments to evaluate proposed active learning based algorithms for speaker diarization.

[1]The task corpus will be released to the speech community for research and algorithmic development.

TABLE I
MY CAPTION

Session Name	Speech (seconds)	Speech Segments	Participants
FD-01	252	161	9
FD-02	314	152	9
FD-03	123	63	10
FD-04	651	358	13
FD-05	457	226	14
FD-06	979	531	12
FD-07	394	267	13
FD-08	486	340	15
FD-09	217	126	13
FD-10	964	713	13
EECOM-01	1206	585	20
EECOM-02	563	252	20
EECOM-03	1014	471	31
EECOM-04	808	384	20
EECOM-05	812	357	26
EECOM-06	475	270	23
EECOM-07	553	337	21
EECOM-08	411	261	19
EECOM-09	744	430	31
GNC-01	859	270	17
GNC-02	735	346	21
GNC-03	653	291	25
GNC-04	1347	494	20
GNC-05	798	440	21
GNC-06	985	456	24
GNC-07	829	481	24
GNC-08	764	435	29
GNC-09	1728	995	29

A. System Setup

1) Segmentation: As the purpose of this paper is to evaluate active learning based bottom-up cluster, we uses reference boundaries to define our segments. The use of such oracle segmentation information in our study is important, as we want to focus only the bottom-up clustering part, and not confused by the errors caused by incorrect segmentation. The previous study has shown that the clustering part of speaker diarization could be developed independent of other modules. Besides, having fixed segmentation and oracle pairwise query answers between segments, are necessary for developing active learning based algorithm, in order to avoid expensive human labeling.

2) i-vector extraction: The i-vector is extracted using the Mel-Frequency Cepstral Coefficients (MFCCs). The 13 dimensional MFCC with deltas are computed every 10ms using 25ms window. We uses 512 mixture universal background model (UBM) trained using the entire corpus data. The final i-vector has 32 dimension.

B. Active Learning Based Cluster Initialization

In this experiments, we evaluate the performance of active learning based cluster initialization, by varying the amount of human effort in terms of number of queries. Note that, the total number of oracle queries to achieve perfect clustering result is $O(N^2)$, where N is total segment in test audio. For all our experiments in this study, we will access to the reference answers of questioned pair, assuming no errors from

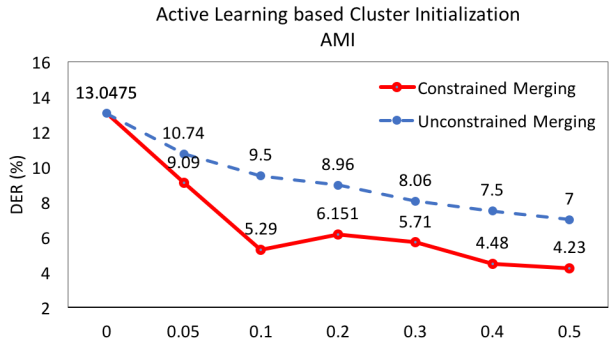

Fig. 5. A boat.

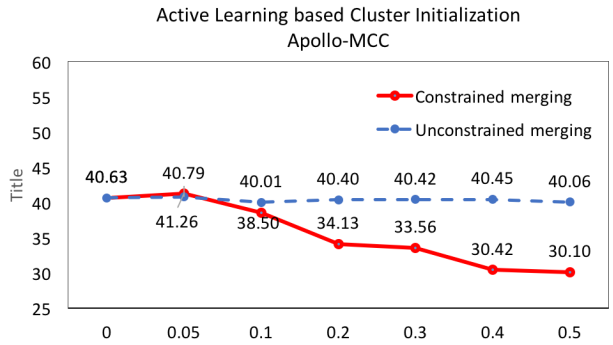

Fig. 6. A boat.

human experts. However, in future study, it will be necessary to consider the human error into account.

We will quantify the number of pair in terms of total segment number of N (e.g., $0.1 \cdot N$, $0.2 \cdot N$). In this experiment, we will use the same amount of queries for both *explore* and *consolidate* stage. For example, both using $0.1 \cdot N$.

C. Active Learning Based Cluster Reassignment

ACKNOWLEDGMENTS

This research was supported by National Science Foundation (NSF) under Grant 1219130.

REFERENCES

[1] P. Kenny, G. Boulianne, and P. Dumouchel, "Eigenvoice modeling with sparse training data," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 3, pp. 345–354, 2005.

TABLE II
MY CAPTION

	explore (0.05)	explore (0.1)	explore (0.2)
Baseline	13.04	5.29	13.04
Random Selection	9.05	6.08	6.17
FFQS (random seed)	9.12	6.17	6.02
FFQS (cluster seed)	9.09	5.29	6.15