Using Simple Speech–Based Features to Detect the State of a Meeting and the Roles of the Meeting Participants

Satanjeev Banerjee and Alexander I. Rudnicky

Language Technologies Institute
School of Computer Science
Carnegie Mellon University, Pittsburgh, PA, USA
`banerjee+@cs.cmu.edu, air+@cs.cmu.edu`

Abstract

We introduce a simple taxonomy of meeting states and participant roles. Our goal is to automatically detect the state of a meeting and the role of each meeting participant and to do so concurrent with a meeting. We trained a decision tree classifier that learns to detect these states and roles from simple speech–based features that are easy to compute automatically. This classifier detects meeting states 18% absolute more accurately than a random classifier, and detects participant roles 10% absolute more accurately than a majority classifier. The results imply that simple, easy to compute features can be used for this purpose.

1. Introduction

We are interested in creating conversational agents that can participate, in a natural fashion, in multi–participant conversations, such as meetings between two or more human beings. Such agents would function as assistants rather than as full-fledged participants, and so would not need to function at the level of the humans in the meeting. But to be effective and to be able to take initiative in matters that they are competent in, such agents will need to form an "understanding" of the discussion in progress. A first step towards creating such an understanding is to automatically detect the state of the meeting and the roles of the different meeting participants at various times during the meeting. A coarse (but robust) labelling of state can then be used as a fremework for more detailed interpretation of meeting activity.

Previous work (e.g., [1]) suggests that utterance lengths and timing information may play a crucial role in detecting meeting states, [2] shows the usefulness of structural information (e.g. who spoke when) in automatically summarizing broadcast news. In this paper, we first introduce a simple taxonomy of meeting states and participant roles, and then investigate the role that simple speech based features can play in automatically detecting these states and roles.

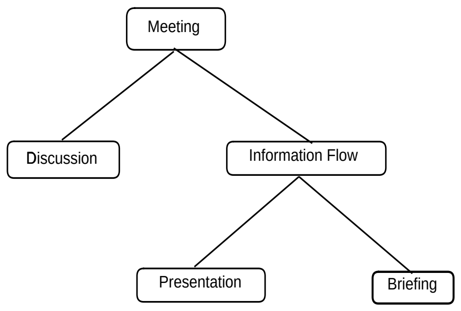

Figure 1: *Taxonomy of meeting states.*

2. A Taxonomy of Meeting States and Participant Roles

Our goal is to create a taxonomy of meeting states and participant roles that has high coverage (every time slice of a meeting can be classified with at least one meeting state label) and low ambiguity (every time slice can be classified with at most one label). Based on a manual analysis of video recordings of several kinds of meetings, we introduce the taxonomy shown in figure 1.

In this taxonomy, we define two major types of meeting states, *discussion* and *information flow*. Typically a *discussion* between a group of people is characterized by each person raising issues, asking questions, making comments, etc. On the other hand we define *information flow* as being a meeting state where essentially one person is giving information to one or more meeting participants. We further sub–divide this state into two states: *presentation* and *briefing*. A presentation consists of information flow from one person to all the other meeting participants, and typically includes a slide or a whiteboard presentation. Further, in this state, typically only one person has the right to speak; when other meeting participants desire to speak, they engage in formal mechanisms of requesting a speaking turn, for example by raising a hand. We define a *briefing* as a less formal information flow from one person to one or more meeting participants (not necessarily all the remaining participants). Typically a briefing does not include any formal slide/whiteboard presentation, and other participants have much more freedom in interrupting the presenter whenever they wish. Thus, the basic difference between presentations and briefings is in the degree of formality, while one of the major differences between information flow and discussion is in how much speech is produced by the different participants.

Within each of these meeting states, we define the participant roles as follows. In the discussion state, those persons taking part in the discussion are labeled *discussion participators*. Within the presentation state, the person presenting is called the *presenter*, while within the briefing state, the person giving the information is called the *information provider* while those meeting participants he is giving the information to are called *information consumers*. Observe that in the briefing state, we draw a distinction between information consumers and other meeting participants who are merely passively observing. From our experience it is usually easy to make this distinction based on the context of the meeting, the content of what is being said, who the speaker is looking at, who is engaging in back–channelling, who is asking follow up questions, etc.

3. Data Collection and Annotation

To gather a corpus of meeting data, we are creating rich multi–modal recordings of meetings between various faculty, staff and students at Carnegie Mellon University, as described in [3]. The two principal streams of information recorded during the meeting are video (recorded using a single whole–scene camera) and speech recorded from close–talking head–mounted micorphones worn by each meeting participant.

Our aim is to train a classifier that uses simple speech features to detect the state of a meeting and the roles of the meeting participants at any given point during a meeting. We created training data for such a classifier by annotating recordings of meetings with the state of the meeting and with the role of each participant at all times through the duration of the meeting, using the taxonomy described above. The annotator works primarily from the whole-scene video of the meeting since it not only contains the speech being spoken but also non–verbal information such as the direction the various participants are looking at, the gestures that the participants are making, the spatial arrangement of the participants, etc. We annotate video using the Anvil tool [4]. This tool allows an annotator to create temporal annotations by first selecting start and stop times of “events”, and then associating the event with user–defined (sets of) labels. In our case, the annotator manually marked off the boundaries at which the meeting state changed, and then labeled each interval between those boundaries with one of the three meeting states: discussion, presentation, briefing. If the annotator could not classify an interval as being any of these three states, he was allowed to label it as “undefined”. Within each such meeting state, the annotator was also required to specify the role of each meeting participant from the list of roles specified above.

4. Detecting Meeting States

We use machine learning techniques to train a classifier on the annotated data created above to learn to detect the state of a meeting. Specifically we design a classifier that takes as input a feature representation of a short window of time during the meeting, and detects the state of the meeting *at the end of that window*. Thus the output of the classifer is one of the four labels: discussion, presentation, briefing, and undefined. The feature representation of the window is described below.

4.1. Features

We used the following four features, all of which are defined for the window of meeting time under consideration. The first feature is the number of times there was a change in speaker within the window of meeting time. Note that this is different from counting the number of times there is a change in *turn*; for example, this count includes back–channels which are usually not regarded as causing a change of turn. The second feature is the number of meeting participants who have spoken at some point or the other within the window. This feature can help differentiate between the situation where there are many people who are contributing to the conversation (for example, during a discussion session) versus a question answer session between an information provider and an information consumer. The third feature is the number of overlaps in speech in the window. This feature serves to capture how often a new speaker starts speaking before the current speaker has stopped. We expect this feature to help capture the difference between a question/answer session on the one hand (since usually the person answering a question waits for the question to end) and a discussion session where participants are more likely to interrupt each other. The last feature is the average length of the overlaps in seconds. This feature is expected to help us distinguish between overlaps due to back–channels (which are usually very short) and other overlaps. Such a distinction should in turn help to distinguish between briefings (which include a lot of back–channeling) and a discussion where overlaps tend to consist of content words (and not just back–channels).

4.2. Data Preparation

As training data we annotated a little over 30 minutes of data from one meeting, while for testing we annotated 15 minutes of data from a different meeting. Next, we created a data point for each second of the meetings, leading to a total of 1,805 training and 902 test data points. We labeled each such data point with the state of the meeting at the end of that second according to the human annotator. Further, for each such data point we computed the above four features for a given window of seconds immediately preceeding the second to which the data point belonged. When the window was longer than the data available (in the beginning of the meetings for example), we used as much data as we could. To keep the features as free of noise as possible, we computed them manually by carefully marking the start and stop times of speech from each speaker.

Of the 1805 training data points, 845 were labeled as being in the presentation state, 455 in the discussion state, 467 in the briefing state and 38 were marked as being undefined. Of the 902 testing data points, there were 670 labeled as discussion, 232 as presentation, and none were labeled as being either in the briefing state or in an undefined state. One of the reasons for this vast difference in label distribution between the training and the test data is that both the meetings were "natural" with no prior constraint on the structure. As a result the training and the test meetings were of very different structure. For example the meeting state changed 17 times in the training data and only 5 times in the test data. As we gather and annotate larger numbers of meetings, we expect the skew in the training set to go down, since we do not expect any one kind of meeting state to occur significantly more often than the others in a large corpus of meetings. For the experiments in this paper, we smooth out the distribution of labels in the training data by keeping only 455 random data points from each of the three major meeting states (presentation, discussion, and briefing), as well as the 38 undefined data points, leading to a total of 1,403 training data points. Since the structure of the meeting is not fixed ahead of time, a random classifier will predict each of the three meeting states with equal probability and get a classification accuracy of 33%. Note that for our experiments here we leave the test data set untouched since we have no control on label skew in unseen data.

4.3. Evaluation and Results

We used the C4.5 decision tree learning algorithm [5] to induce a classifer from the training data prepared above. Table 1 and figure 2 show the classification accuracies obtained when testing the learned classifier on the test data. The classifier beats the random classifier with just 2 seconds of history. A perusal of the tree induced in the two–second–history situation revealed that this tree labeled a data point as "presentation" if and only if the number of speaker changes in the past one second was 0. If this number was greater than zero, but there were no overlaps in speech, it labeled the data point as being a discussion.

The accuracy of the classifier improves with increasing window sizes, reaching 51% around the 20 second window mark. This represents a 18% absolute (54% relative) improvement over the random classifier. The tree learnt at the 20 second mark also revealed the number of speaker changes to be the top–most node, implying that that is one of the most important features for detecting the state of the meeting.

Table 1: *Accuracy of detecting meeting states and participant roles.*

Window size (in seconds)	Meeting state detection accuracy (in %)	Participant role detection accuracy (in %)
1	32.9	46.1
2	39.9	47.1
3	42.7	47.8
4	48.2	49.0
5	46.3	49.3
10	39.9	51.3
15	38.7	53.1
20	51.1	52.8
25	50.3	52.6
30	38.6	53.4
45	37.6	52.5
60	32.8	48.3

5. Detecting Participant Roles

Our second goal is to automatically detect the role of each participant. Specifically we wish to create a classifier that takes as input the same kinds of features as the meeting state classifier, and outputs one of the participant role labels, namely discussion participator, presenter, information provider, information consumer, and undefined.

5.1. Features

For the participant role classifier we use all the features described in section 4.1. In addition we use the following three features, each of which is defined for the participant in question during the given window of time. The first feature is the total amount of speech spoken by this participant in seconds. We expect this feature to be very useful for being able to tell apart the presenter from the others in the presentation state, and the information provider from the information consumers in the briefing state. Besides this feature we also include the total amount of speech of this participant in seconds that is overlapped with speech from other participants. We split this information into two features: counting in one feature all those overlaps that were initiated by this participant (for example, through back–channels and interruptions done by this participant), and counting in another feature all the overlaps initiated by some other participant (for example where this participant was interrupted). These two

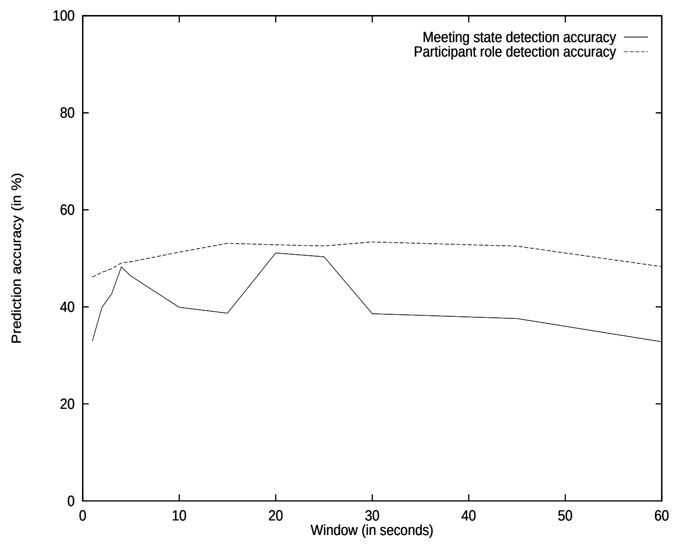

Figure 2: *Meeting state and participant role detection accuracy.*

features are aimed at capturing such pieces of information as the amount of back–channelling this participant is engaging in, etc.

5.2. Data Preparation

Data was prepared for this task in the same way as it was for the meeting state detection task. Each second of the meetings led to as many data points as there were meeting participants. Our training data had a total of 12,635 data points. Of these, 9,561 were undefined (which included data from persons who were temporarily away from the meeting, as well as those who were at the meeting but are not participating in the conversations, etc.), 1,252 were labeled as discussion participator, 845 as presenter, 507 as information consumer and 470 as information provider. The test data consisted of 6,314 data points, of which 3,311 were of type discussion participator, 232 presenter and 2,771 were undefined. Unlike the meeting state task, we do not expect participant roles to be uniform in distribution. For example, although presentations are as likely to take place as discussions, far fewer people–seconds are spent presenting than discussing or remaining silent at a meeting. Thus, we leave the training data unbalanced, and use "undefined" as the majority classifier's label, which leads to an accuracy of 43% on the test data.

5.3. Evaluation and Results

We again used the C4.5 decision tree learning technique to induce a classifier from the training data, which was then tested on the test data. Table 1 and figure 2 show the accuracies obtained. The classifier gets a classification accuracy of more than 46% with only one second of data, which beats the majority classifier. The top–most node in the tree was the total length of speech of the participant, implying that the amount of speech a participant produces is the most important indicator of his role in the meeting. Similar to the meeting state detector, the accuracy of the participant role detector increases with increasing window size, reaching about 53%, which is a 10% absolute (23% relative) improvement over the majority classifier.

6. Conclusion

In this paper we have presented a new taxonomy of the states in a meeting, and the roles of the various meeting participants. We have trained a decision tree classifier that performs 18% (absolute) better than random at detecting the state of the meeting, and 10% (absolute) better than the majority classifier at detecting participant roles, using about 20 seconds of meeting history. We expect these results to improve with models trained on larger corpora of meeting data. In the future we also plan to investigate how results are affected when using automatically generated features.

7. Acknowledgements

This work was supported by DARPA grant NBCH-D-03-0010. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

8. References

[1] H. Yu, T. Tomokiyo, Z. Wang, and A. Waibel, "New developments in automatic meeting transcription," in *Proceedings of the International Conference on Spoken Language Processing*, Beijing, China, 2000.

[2] S. Maskey and J. Hirschberg, "Automatic summarization of broadcast news using structural features," in *Proceedings of Eurospeech 2003*, Geneva, 2003.

[3] S. Banerjee, J. Cohen, T. Quisel, A. Chan, Y. Patodia, Z. Al-Bawab, R. Zhang, P. Rybski, M. Veloso, A. Black, R. Stern, R. Rosenfeld, and A. I. Rudnicky, "Creating multi-modal, user–centric records of meetings with the Carnegie Mellon meeting recorder architecture," in *Proceedings of the ICASSP Meeting Recognition Workshop*, Montreal, Canada, 2004.

[4] M. Kipp, "Anvil – a generic annotation tool for multimodal dialogue," in *Proceedings of Eurospeech 2001*, Aalborg, 2001, pp. 1367–1370.

[5] I. Witten and E. Frank, *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco, CA: Morgan–Kaufmann, 2000.