

Meeting Analytics: Creative Activity Support Based on Knowledge Discovery from Discussions

Katashi Nagao
Graduate School of Informatics, Nagoya University
nagao@i.nagoya-u.ac.jp

Abstract

We are researching a mechanism to promote innovation by supporting discussions based on the premise that innovation results from discussions. Ideas are created and developed mainly by conversations in creative meetings like those in brainstorming. Ideas are also refined in the process of repeated discussions.

Our previous research called discussion mining was specifically used to collect various data on meetings (statements and their relationships, presentation materials such as slides, audio and video, and participants' evaluations on statements). We extracted important statements to be considered especially after the meetings had been held and actions had been undertaken, such as investigations and implementations that were performed in relation to these statements by using the collected data.

Here, we present high-probability statements that should lead to innovations during meetings and facilitate creative discussions. We also propose a creative activity support system that should help users to discover and execute essential tasks.

1. Introduction

For meeting and discussion activities in particular, it is important to organize the various thoughts of a group, solve problems, and decide policies. Analyzing and reviewing the content of such meetings is effective for promoting future activities. For example, in seminars with presentations about research progress held regularly in university laboratories, there is much advice and many proposals given to a presenter. Although analyzing and reviewing statements that indicate future tasks is essential in setting a new objective, such statements are sometimes difficult to extract due to other statements appearing in the meeting minutes. Overlooking “task statements” leads to a loss of productivity for not only the presenter but also the entire organization.

Our research was aimed at deeply analyzing face-to-face meetings and supporting human creative activities by appropriately feeding back knowledge discovered

from the meetings. We particularly analyzed the features of statements made during discussions. We have been developing a system called “meeting recorder” for that purpose. Meeting recorder consists of a 360 degree panoramic video camera that records meetings in audio-visual scenes, a tablet application that allows users to browse meeting materials and add various notes to them with a stylus, speech recognition that identifies speakers and transcribes the speech content of all meeting participants, and a minute server that integrates all meeting-related information and creates the meeting minutes.

For this study, we applied machine learning techniques for discussion-activity data obtained using a discussion mining (DM) system [1] that structures the meaning of meeting content and records it. We call this discipline of analyzing meetings and mining valuable information from them automatically “meeting analytics.”

We organized and developed ideas that became central topics at meetings or planned activities to solve issues stated in the task statements after meetings had been held based on important statements that had been made during the meetings. We propose a system that supports these activities called the “creative activity support system.” This system supports users in quoting statements extracted from the minutes, in writing notes and reports, in creating activity plans, in managing schedules to accomplish tasks, and in evaluating other members' results within the group.

2. Meeting Recorder

We developed a system called a meeting recorder to record and analyze small-scale and face-to-face meetings in detail. It consisted of an Internet of Things device with a 360-degree all-around panoramic camera, tablets for all participants, and a server computer.

This was aimed at achieving a simplified portable version of the discourse mining system that will be described later and it was also aimed at fully automating the creation of the minutes.

2.1. System Configuration

The system records meeting scenarios with a panoramic camera and tracks participants' faces by assuming that there are people walking around in meetings. The system also simultaneously collects all the participants' voices with small wearable microphones. The speech recognition results and meeting materials are synchronously displayed on the tablets of participants. In addition, users can freely draw and mark figures on documents with stylus pens. The stylus inputs are also shared on all the participants' tablets in real time.

There is a photograph of usage situation of the meeting recorder we developed in Figure 1.

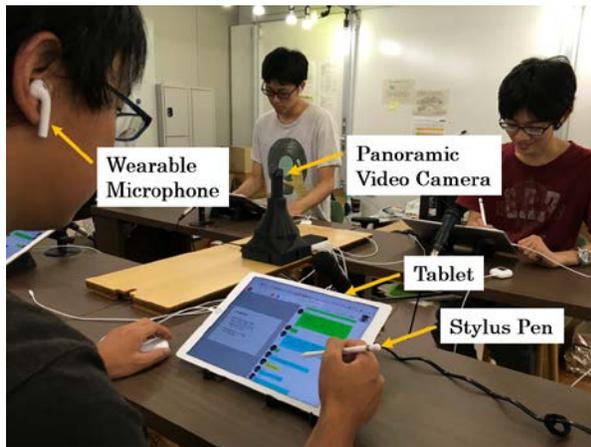


Figure 1. Meeting recorder in action

2.2. Speech and Handwriting Inputs

The voices of meeting participants that are input from the wearable microphones are recorded on the meeting server together with the IDs of the speakers and when their speech started and ended. These are simultaneously transcribed using speech recognition on the IBM Bluemix cloud and displayed on the tablets of all participants. The facial images of the speakers obtained from the panoramic camera are then also simultaneously displayed.

The material displayed on tablets are images of a file (mainly in PDF format) transmitted beforehand to the meeting server, which can be marked with a pen linked to the tablets. Materials and markings are synchronously displayed on the tablets of all participants, and the statements and the materials (including markings) are automatically associated and used for structuring the discussion, which will be described later.

2.3. Generation of Minutes

The minutes include video, audio, images, and text. An image photographed with a panoramic camera is

automatically scrolled so that a speaker is always displayed at the center of the screen. The material and the statement text are associated and displayed in list form. Based on topic segmentation, which will be described later, the list is divided into topics and displayed in chronological order.

2.4. Topic Segmentation

A machine learning model was constructed to classify statements from the viewpoint of newly introducing topics or continuing previous topics that was based on data obtained by discussion mining, which will be described later. As a result, we classified the speech recognition results in real time and identified topic breaks.

The features extracted for model building are summarized in Table 1.

Table 1. Features for topic segmentation

Feature	F1 measure
All features used	0.9555
Relationship between statement and presentation material	0.7611
Whether the speaker of statement was the same as that of two previous statements	0.9101
Elapsed time since previous statement	0.9434
Cosine similarity between statement and previous statement	0.9519
Morpheme unigram and bigram	0.9525
Whether the slide change was made while speaking	0.9537
Entity grid [2]	0.9543
Length of sentence	0.9555
Duration of use of pen	0.9555
Region overlap of pen drawing	0.9555
Number of sentences in statement	0.9555
Presence of reference expression (e.g., pronouns) in statement	0.9555

Machine learning was carried out by logistic regression analysis, and the training data used the minutes accumulated from discussion mining, which will be described later.

The F1 measure when experimenting with the test data is also listed in Table 1. The F1 measure to the right of the feature in the table represents the value when the feature was not used for learning. This indicates that topics were segmented with sufficient precision.

We then aimed at extracting more advanced information from the minutes. It was necessary to analyze the language content of statements and materials in detail to achieve that purpose.

Although the meeting recorder accurately recorded and processed all meeting information, more advanced knowledge discovery could be achieved by manually processing the meeting information partially.

3. Discussion Mining

Data for training are necessary to analyze the data acquired by the meeting recorder. We have thus far collected data by using a mechanism for analyzing discussions.

Seminar-style meetings that are regularly held at university laboratories are places where exchanges of opinions on research content occur. Many comments on future work are included in their meeting records. However, as discussions at meetings are generally not recorded in detail, it is difficult to use these for discovering useful knowledge. Our laboratory, on the other hand, has been developing and operating a system that systematically records the content of face-to-face meetings with metadata and achieves support, such as that from reviews of discussion content. Although it is essential to review tasks to set new goals in research activities, their existence may be concealed in many other statements in the minutes.

Therefore, we propose a system of supporting task execution in student research activities in this research by combining discussion mining that applies data mining to the content of discussions and a method of machine learning called active learning.

We conducted experiments with students who were about to graduate and complete their studies at universities as subjects. We also carried out computer experiments on improving the accuracy of extraction with the proposed method of active learning to demonstrate the effectiveness of this system.

3.1. Discussion Mining System

Our laboratory has documented records of seminars in detail for more than a decade [1]. The discussion mining system enables all participants to cooperate to create and use structured minutes.

Instead of a fully automated system like the meeting recorder that was described in the previous section, the secretary manually describes the content of a speech, and each speaker tags his/her speech, so it is possible to generate data with considerably high levels of precision.

The content of a meeting is specifically recorded with a Web browser based presenter/secretary tool by using the multiple cameras and microphones installed in the discussion room shown in Figure 2. There is also a main screen that displays the presentation material and the state of the demonstration at the center of the discussion room, and there are sub-screens for displaying meeting metadata on both sides, including information on participants who are currently speaking and camera images.

There are two kinds of tags to apply to statements, the first is a tag called start-up to introduce new topics,

and the second is a tag called follow-up to continue the topic already being discussed. It is necessary for the follow-up statement to clarify from which statement it is continuing.

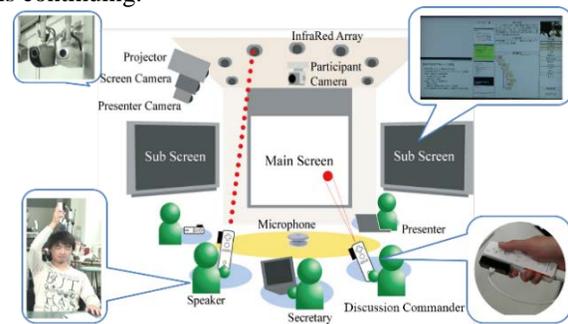


Figure 2. Discussion room

A discussion constitutes a tree structure by speakers always attaching attributes to their statements and associating them with other statements. This tree structure is visualized in real time during a meeting and it is used to oversee the discussion.

The meeting style supported by the discussion mining (DM) system involves the presenter explaining it while displaying slides, and question and answer sessions with the meeting's participants starting during the presentation or when it ends. The DM system records slides that are displayed by the presenter and question-and-answer sessions carried out with subsequent participants in a segmented manner. As a result, the content illustrated in Figure 3 is generated.



Figure 3. Meeting content

Every participant inputs metadata about his/her speech using a dedicated device that is called a discussion commander, as shown in Figure 2. Participants who specifically ask questions or make comments on new topics assign start-up tags to their statements. Also, if they want to speak in more detail on topics related to the immediately preceding statement, they provide a follow-up tag. Furthermore, the system records pointer designates the location/time for the slide and information on the button for or against the

statement during the presentation and during the question and answer session. Marking information on important statements is also recorded.

We also created a system of searching for and viewing recorded data. It is possible to search content from the date and participant information, to view past discussions similar to an ongoing debate, and to effectively visualize the state of discussions in the meeting content browsing system.

3.2. Structuring Discussions

Methods of acquiring metadata about meetings include a method using an automatic recognition technique, such as a meeting browser [3] and a method of human input using devices and tools such as a conversation quantizer [4]. The amount of human effort when acquiring metadata is very small with the former method, but it is currently difficult to automatically record all necessary information on a computer. Although it is possible to search for the keywords of statements, they do not contain sufficient information on the content to be understood when browsing. Therefore, we adopted a method in this research in which humans and machines cooperatively input these metadata.

In the DM system, the presenter uploads a presentation slide by using a special tool and the system automatically transmits the slide information when the presenter changes the slide page. In addition, all the participants use the portable discussion commander. The start time for the statement and the statement type are recorded, in addition to the speaker ID and the seating position of the speaker, by placing this device on the top. The end time for the statement is input by pressing a button on the device. The system segments video/audio information on each statement by acquiring the start/end times of the statement. In addition, it is possible for a user to express an attitude (agree or disagree) with respect to the statement by pressing a button on the device or to mark a statement that has an important meaning to himself/herself.

The DM system is provided with a statement reservation function as a mechanism for controlling the order of statements by participants to prevent crosstalk. When someone signals his/her intention to speak by holding the discussion commander above them, his/her name with the type of statement is added to the statement reservation list, and his/her speaking turn automatically shifts to him/her when the immediately preceding statement ends.

This statement reservation function is not only used to control the order of statements but also to create a discussion structure that reflects human intentions. If the system only uses the statement type without using the statement reservation function, the discussion structure

that is created will be a list structure starting from the start-up statement. However, the discussion structure when multiple participants express opinions from various perspectives on the content of one statement is considered to be a tree structure rather than a list structure. Therefore, when a follow-up reservation is added while someone is speaking, the DM system generates link information between the follow-up statement and the ongoing statement. In other words, when multiple reservations are added during speaking, a tree structure in which a plurality of follow-up statements is repeated is automatically created for one statement.

The root of the tree structure in this case is a start-up statement, where all the others are follow-up statements. If several follow-up statements are added at the same time for one statement, the branches of the tree increase. Follow-up statements are attached to the preceding statement, which deepens the tree structure.

3.3. Summarizing Discussions

Important statements are those that have a great influence on participants' thoughts and opinions, more active discussions, and well-organized discussions up to that point. What can be considered as factors that affect the importance of statements are the number of branches of that statement (the number of follow-up statements following the statement), whether it is on a longer thread, the social position of the speaker, and the cumulative number of follow-up statements.

Although we wanted to determine what statements were important, start-up statements were not always important, and statements by specific speakers were not always consistently important.

Therefore, we propose an algorithm in this paper to discover important statements by spreading activation. This is based on a network where nodes are statements and links are introduced based on the structure of the discussion segment (start-up/follow-up tags) and links based on the pointer referents (links of statements having the same referred object are linked).

Spreading activation is a model in which the activity value of a node diffuses to a neighboring node through a link in a network represented by nodes and links, as outlined in Figure 4. In other words, the activation values of nodes closer in distance to a node with a higher activation value are also higher, and conversely, a node whose distance is farther is also lower in activation value. By applying this method, it is possible to calculate the relative activation value of each node for all the nodes in the network.

Specific methods of calculation are as follows. First, matrix A that represents the presence or absence of a

link between each node is created by assuming that the number of nodes existing in the network is n .

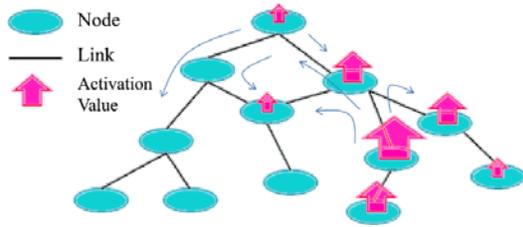


Figure 4. Spreading activation

Here, the value of (i, j) is one if there is a link between nodes i and j , and zero if it does not exist. At this time, matrix W that is related to the weight of active diffusion is created by using matrix A . Matrix W is an $n \times n$ matrix obtained by dividing the value of each row of A by the number of non-zero elements of that row (this is done to converge active diffusion). Matrix C that represents a constant activation value spreading from the respective nodes to the whole network is next introduced (red arrow in Figure 4).

Here, the value of c_i is the external input value of the constant given to node i .

Then, by solving the recurrence formula, $X_0 = C$, $X_{t+1} = W \cdot X_t + C$, the activation value in the network at the time of step t ($= 0, 1, 2$) is obtained. Here, $x_{t,i}$ represents the activation value of node i at step t . Finally, the activation value of each node in the network is calculated as $t \rightarrow \infty$. Also, when $t \rightarrow \infty$, the above recurrence formula can be expressed as $X = W \cdot X + C$, so that $X = (E - W)^{-1} \cdot C$ is obtained (where E represents an identity matrix).

We implemented a mechanism on the browser that selects important statements according to the purpose of browsing to efficiently browse the recorded discussion. We specifically constructed a network where nodes from each statement from the link information were obtained from the discussion structure and pointing acts, calculated the activation value given to each node from the agreement/opposite button and marking, and used the spreading activation algorithm. We then ranked importance ranking and filtered the statements.

For example, if the follow-up statement of statement i is j , there is a link between statement i and statement j . By repeating this, a network of statements is formed for each discussion segment starting with the start-up statement. Further, the links of all the start-up statements in each discussion segment are linked with a certain virtual node. All the discussion segments recorded from this are represented by one network structure. We can then execute the spreading activation algorithm on this network structure and find the relative activation value for each statement. Values that take into

consideration various metadata acquired at a meeting are used as external input values to be given to each node. Specific metadata include statement types, button information for agree/disagree, marking information, speaker names, and keywords included in the object. Since the start-up statement is a premise of follow-up statements in the same discussion segment, it is set so that the activation value is higher than that of the follow-up statements.

The users of the discussion browser (Figure 5) use the user interface to find how much importance is given to each item of metadata. They can interactively use this interface and discover a set of statements that is suitable for their own reading purposes. For example, if a user increases the importance of the start-up statements, he/she can efficiently discover the statement set that has the same topic as the start-up statements, and the statement set that gained numerous agreements when the importance of the agree button information was increased by the user.

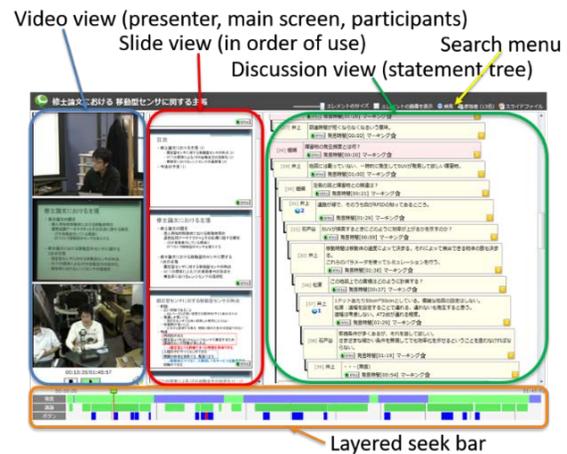


Figure 5. Discussion browser

3.4. Task Statement Extraction

Remembering past discussion content helps us to seamlessly carry out future activities. For example, presenters in laboratory seminars can remember suggestions and requests about their research activities from discussion content that has been recorded in detail. The meeting content contains useful information for the presenters, but it is onerous to read the information. As necessary information is concealed in a large number of statements, it is not easy to find. This is problematic if past discussions are not being reviewed, even for other speakers, and not only presenters. Therefore, it is necessary to extract information that concerns unsolved issues from previous discussions. We have called statements that include future tasks “task statements.”

We developed a method of statistically determining whether statements were about future tasks, i.e., task statements [5]. Some attributes, including linguistic characteristics, structures of discussions, and speaker information, were used to create a probabilistic model.

3.4.1. Model of Task Statement

A task statement can include any of three types of content.

1. Proposals, suggestions, or requests provided during the meeting: The presenter has determined that they should be considered.
2. Problems to be solved: The presenter has determined problems that should be solved.
3. Tasks not yet carried out before the meeting: the presenter has occasionally already noted such tasks.

Candidates of task statements are fragments of a discussion chunk, which was explained earlier. A typical chunk is created from one or more of the questions and comments of the meeting participants and the presenter's responses to them. A coherent piece of discussion content related to tasks consists of questions/comments and their responses. Thus, "participants' questions/comments + presenter's responses" are a primary candidate and a target of retrieval. "Participants' questions/comments and no response" is a secondary candidate. Figure 6 has an example of candidates for task statements.

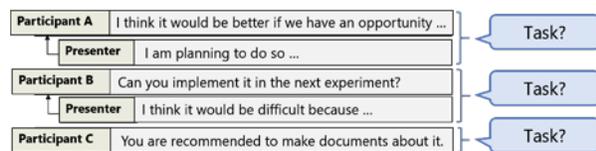


Figure 6. Candidates for task statements

3.4.2. Extraction Procedure

The method generates a probability model by using logistic regression analysis by using correct data that were manually created from past meeting content. The method calculates the probabilities of each candidate for a task statement by using a generated probabilistic model. A candidate whose probability value exceeds a certain threshold, e.g., 0.5, is extracted as a task statement.

3.4.3. Characteristic Features

Some of the past meeting content was manually analyzed to find characteristics that could be used as clues to extracting task statements. The survey data included 11 types of meeting content and 598 groups of statements (candidates). Each presenter of the meetings

manually selected task statements from each type of content.

As a result of manually extracting task statements in the survey data, 246 task statements were found, which corresponded to 41.1% of all candidates. We analyzed the characteristics of the task statements by comparing such percentages. For example, statements made by teachers had a higher overall percentage of task statements. Therefore, speaker attributes were helpful for calculating the probabilities of task statements.

As was explained earlier, presenters used their discussion commanders to mark statements that they wanted to check later during the meetings. We investigated the effectiveness of marking for discriminating task statements by calculating the percentage of marked task statements in all task statements. The percentage of task statements that were marked was 73.4%, which was higher than that of the task statements for all candidates.

We found distributions for the respective characters of a presenter's and participants' statements to examine whether there were characteristic tendencies in the number of letters (characters) in task statements. We divided the number of characters into five groups and calculated the percentages of task statements in each group. The percentage of task statements in the participants' statements increased when the number of characters increased. This was because the number of characters in their statements increased when the participants were making concrete requests and giving advice. In comparison, the number of characters with higher percentages of task statements was 20 or less for the presenter's statements. The more characters there were, the smaller the percentage of task statements. We believe that if a presenter accepted the requests or advice from participants who were present, his or her responses would have tended to be brief.

We also investigated the types of sentences included in the task statements. The percentage of task statements in the participants' statements was higher when sentences were in present tense and in declarative form (56.1%). This was due to the fact that a large amount of advice or requests was in the pattern of "should be ..." or "I want to ...". The percentage of task statements for the presenter's statements in past tense and in declarative form was low (29.2%). This was because presenters did not tend to use sentences in past tense when they talked about future tasks. In addition, the percentage of task statements made by presenters in past tense and in interrogative form was 0%.

Morphemes and collocations of morphemes in statements were also important features. We generated a morpheme bigram of nouns, verbs, adjectives, and auxiliary verbs in the survey data by calculating the number of occurrences of the morphemes. We then

determined a feature of morphemes and their bigrams of statements if their occurrences had exceeded certain thresholds. The selected nouns specifically had a percentage of occurrences that was greater than or equal to 0.5% of all nouns, and the selected verbs also had a percentage that was greater than or equal to 0.5% for all verbs. Morpheme bigrams were selected if their percentages were greater than 0.05% for all morpheme bigrams. These selected morphemes and bigrams were used as features for discriminating task statements.

Three main features were selected on the basis of these results obtained from the survey to create a prediction model.

1. Attributes of presenter
2. Features of participant's statements
 - (a) Start time and duration of statements
 - (b) Speaker type (teacher or student)
 - (c) Statement types (start-up or follow-up)
 - (d) Marking (zero or one)
 - (e) Length (no. of characters)
 - (f) Sentence types
 - (g) Morphemes and morpheme bigrams
 - (h) Responses by presenter (zero or one)
3. Features of presenter's response
 - (a) Marking (zero or one)
 - (b) Length (no. of characters)
 - (c) Sentence types
 - (d) Morphemes and morpheme bigrams

We used answers (zero or one) to five questions for the values of sentence type features.

1. Does the statement include a sentence in past tense and in declarative form?
2. Does the statement include a sentence in present tense and in declarative form?
3. Does the statement include a sentence in past tense and in interrogative form?
4. Does the statement include a sentence in present tense and in interrogative form?
5. Does the statement include a sentence of another type?

3.4.4. Verifying Extraction Results

Ten-fold cross-validation was applied to the extraction results to test and confirm the effectiveness of the proposed method. The data used for verification included 42 types of meeting content and 1,637 groups of statements (candidates). Each presenter created correct data for task statements for each type of meeting content as well as the survey data provided earlier. The data used for verification were completely different from the survey data.

We confirmed the effectiveness of the proposed method in terms of high precision (index for extraction

accuracy), recall (index for extraction leakage), and the F1 measure (harmonic mean of precision and recall).

The extracted results from the task statements with the proposed method were a precision of 75.8%, a recall of 64.2%, and an F1 measure of 69.5%. In comparison, the results for the three alternative methods of extraction were as follows. The selection of statements that were marked by the presenter had the highest precision (68.9%), and that of the statements from teachers or statements that were marked by the presenter had the highest recall (44.1%) and F1 measure (48.7%). The approach we propose obtained the highest values compared with these other extraction methods.

As was previously explained, the proposed method could calculate the probabilities of candidates for a task statement by using a generated probabilistic model. A candidate whose probability value exceeded a certain threshold was extracted as a task statement. We first set the threshold value to 0.5. It was not guaranteed that this value would be optimal. Therefore, we re-evaluated the outputs of the system by lowering the threshold by 0.1 from 0.5. As we found that the F1 measure at a threshold of 0.4 was highest (71.4%), task statements should be extracted in the future by setting the threshold to 0.4.

3.5. Improving Accuracy of Extraction by Active Learning

The discriminant model should always be updated when the amount of analysis data is increased. However, it is very difficult to label all data when increasing amounts of data are not labeled, where labeling incurs large costs. We used active learning [6–8], which has been used to attempt to improve the accuracy of extraction, for the limited data set we obtained to solve this problem.

Active learning is a technique that is frequently used in natural language processing, which may require expert knowledge concerning the assignment of teacher signals, and biostatistics, where the costs of collecting data and extracting teaching signals are very high. The technique is also suitable for improving the accuracy of task-statement extraction in which the cost of teacher-signal assignment is also very high.

A five-step procedure is repeated to apply the technique to the DM system.

1. Create a task statement extraction model by using a set of statements with teacher signals to determine whether or not they are task statements.
2. Carry out task-statement extraction with the recently created minutes after a presentation has been made at a seminar and calculate the probability value that each statement is a task statement.
3. Select the target set of statements to which teacher signals are to be assigned by active learning.

4. Present the results and request teacher-signal feedback from the user on the extension system for the selected statement set of the DM system that will be described later.

5. Add the statement set with the teacher signals that have been obtained to the training data for machine learning.

It is not only possible to improve the accuracy of extraction by increasing the amount of training data but also to constantly adapt to the feature changes in task statements over time by updating the task-statement-extraction model by repeating this five-step procedure.

Information density is a representative algorithm of the active-learning technique that takes into account the density on the feature space as a sampling reference [9]. The strategy with this algorithm is to consider that there is a large amount of information with higher density data; therefore, the benefits of providing teacher signals are considered to be great.

Since information on teacher signals is not used for density calculation, it is possible to use a large number of samples without teacher signals for learning. As the DM project has been continuing for more than a decade, the content on accumulated discussions is huge; therefore, it is very convenient to apply this method.

Although the practicality of information density is high when there is a large amount of noise data, this may adversely affect the weighting of model parameters. Morpheme information from statements is used in the task-statement-extraction model, and extremely long or short statements can be noise.

Our proposed method used a weighting algorithm that was based on a histogram of feature information in this research to cope with such problems. Since all the features used in the task-statement-extraction model are binary variables, we can create a histogram in which features that have a value of one are classified as a class. We called this a “feature histogram.” We created a feature histogram from the past 492 meeting minutes.

Our weighting algorithm was based on the ratio of frequency to the number of samples without teacher signals as a weight calculated as:

$$\arg \max_{x \in U} \phi_x \times \left(\frac{1}{|U|} \sum_{i=1}^U \text{sim}(x, x_i) \times \frac{\text{freq}(\text{bin}(x))}{|U|} \right)$$

Here, ϕ_x is the score of sampling, U is the set of samples without teacher signals, $\text{sim}(x, x_i)$ is the cosine similarity between x and x_i , $\text{freq}()$ is the frequency of a suggested bin, and $\text{bin}(x)$ is the bin to which x belongs.

This weighting makes it possible to assign a light weight to statements of extreme length that can have a negative effect on information density, and more effective sampling can be expected in regard to improving the accuracy of extraction.

4. Creative Activity Support

We developed another system to extract important statements and task statements and to reflect them in subsequent activities to make effective use of the meeting results. This system consisted of a function that involved writing notes and reports by quoting statements, a function that involved managing schedules concerning the execution of tasks, a function that involved disclosing and sharing notes and reports in a group, and a function that involved group members mutually evaluating the reports. We called the system a “creative activity support system.”

4.1. Creative Activity Support System

The creative activity support system is a system that guides subsequent activities from immediately after a meeting has been held based on the results obtained from discussions. As there is a possibility of forgetting content for a while after meetings, particularly for task statements, and there is a possibility of neglecting this, it is necessary to associate this with the activities as soon as possible. We implemented functions that users could easily remember to quote task statements, formulate action plans, reflect on schedules, and be appropriately reminded of the task execution plan for that reason.

It is not easy to steadily complete various tasks that arise in work and research. Indeed, there have been many arguments on how to manage tasks to achieve success, and many scheduling-support systems, such as Google Calendar, have been developed to support this.

Most conventional scheduling-support systems have only been focused on managing the schedule of plans, and not on understanding how an established schedule is processed and what state it is in before proceeding to the next task. There is no support in implementing tasks that have causal relationships over the long term, such as setting guidelines. One major issue related to graduation and completion studies is addressed by carrying out individualized tasks that have been segmentalized in a long-term time series, so the plan-do-check-act (PDCA) cycle in business execution is recommended for education as well. It is necessary to have a mechanism to support the smooth execution of a series of performance cycles of planning, executing, and evaluating tasks.

In addition, although it is necessary to be aware of the existence of all tasks to be scheduled by students as a prerequisite in conventional efforts related to task-achievement support, we aimed at advanced support that included awareness of the tasks to be carried out.

The flow of task achievement is outlined in Figure 7, and the underlined parts are executed by the system. First, the user creates notes for a performance plan to

complete the tasks for the task statements extracted by using the learned extraction model and then manages the execution schedule with the scheduling components of the system. He/she then carries out the tasks according to the planned schedule, receives evaluations from other users (support for the check step) on the basis of the results added to the performance-plan notes, and considers the evaluations to improve subsequent activities.

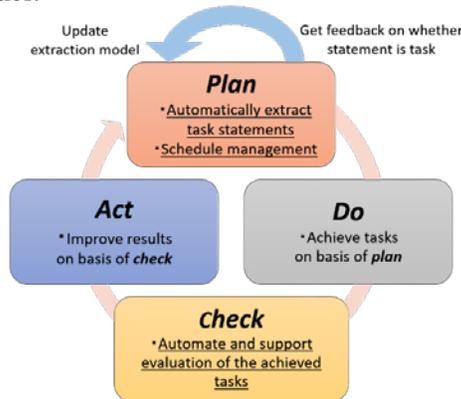


Figure 7. Flow to complete tasks

4.2. Support for Task Execution and Evaluation of Results

The most important thing in executing a task is that it achieves some goal. In other words, it is necessary to first consider what target is set, which part of the goal is associated with the task to be carried out, and what will be achieved by the solution.

The creative activity support system has a map that roughly describes the entire activity in tree structure form and can search which part of the target is related to the problem to be solved from the map.

Users can continue their activities while confirming that they are heading toward the goal by doing so, and not just executing the task in front of them.

We must begin by being aware of the problems that confront us at the moment to facilitate research activities. Task statements extracted with the proposed system are presented in Figure 8.

Each statement that is presented is determined to be a task statement with high levels of possibility, and it is necessary for the user to finally assess whether the task should be achieved. Therefore, since the statement immediately after extraction has a blue icon marked “Task?”, which can be clicked, it determines whether the statement indicates a task to be achieved.

The user creates a note citing the task statement after deciding the task and describes the details of the plan and results related to the task. By setting the attribute of this note to “Finished,” a completion icon is displayed

in the related task statement. The user can then easily be aware of which task is in a completed state.



Figure 8. Presentation of tasks

It is adverse for students with little experience in research activities to only be aware of the number of tasks because there is uncertainty as to whether the tasks can be appropriately executed. Therefore, each task should be well organized and scheduled after the existence of tasks is identified to use time efficiently. We implemented a task scheduler (Figure 9) in the proposed system that could schedule the duration of task executions.



Figure 9. Task scheduler

There is a graph in the upper part of the screen of the scheduler that can roughly organize the proportion of information by the task type, e.g., surveillance, development, or experiments, and their achievement situations. The user can arrange and update the time intervals of the tasks scheduled to be executed on the time table at the bottom of the screen.

After the tasks have been carried out, we should assess the progress of the task, understand the level of achievement, and use it as a source of future activities [10]. We implemented a function to evaluate task-execution content in the proposed system by receiving other students’ assessments on the content of execution by publishing task-execution notes in the laboratory.

We confirmed that mutual evaluation became practical by referring to notes that described the content of task execution. When completing task execution, the system accessed the level of achievement in the research goals. The user could then publish related research notes and associate them with the previously shared notes to motivate others to update evaluations.

4.3. Definition and Assessment of Creativity

Although how to capture creativity is varied, we regarded it as the degree of development of ideas in this research. In other words, the extent to which ideas was refined and materialized after discussion was used as an indicator of creativity.

The ideas should appear in the statements in the discussions and extracted from the minutes and cited in the creative activity support system. Their further development will therefore be possible. Then, after the reports are published and the evaluations are received, the values of the ideas are augmented, and they can be further developed by making them the topics of the next meeting.

Creativity can be represented by using concrete data and modeled by machine learning in this way by tracing the generation of ideas through to their development.

4.4. Acceleration of Innovation

When it becomes possible to predict the creativity of ideas by machine learning, and when multiple ideas to solve the same problem are conceived, it becomes possible to rationally decide which one to preferentially develop. This is done by presenting multiple ideas at a meeting and by automatically evaluating extracted statements after discussions.

It is thought that innovation will be accelerated by repeating choices based on the creativity of the ideas. Innovation cannot be achieved unless it develops ideas to a level that can affect society, but it will be possible to efficiently use time to carry out creative activities by considering which ideas to preferentially develop.

This is presently not an era when innovation is generated from a few outstanding ideas. However, a plethora of ideas that can create innovation in the future can be created from discussions by numerous people. True innovation can be achieved by carefully developing such a profusion of creative ideas.

5. Concluding Remarks

We proposed a system of discovering creative ideas that led to future innovations and developing them

efficiently by recording and analyzing meetings in detail. We intend to clarify the true value of this research by gathering data on a large scale, building a machine learning model, and operating it in the future.

We are currently collecting and analyzing data on meetings at the IBM Tokyo Research Laboratory and Nagoya University. We are also operating the creative activity support system and collecting data on students' creative activities after holding meetings at Nagoya University. We also intend to conduct comparative analysis of the characteristics of meetings held by academia and industry in the future.

Acknowledgments

The author would like to thank Toru Nagano, Osamu Ichikawa, Gakuto Kurata, and Ryuki Tachibana of Cognitive Computing Group of IBM Tokyo Research Laboratory for their support for developing the meeting recorder. The author also thanks members of Nagoya Laboratory of Nagoya University for their efforts for developing and managing systems.

References

- [1] K. Nagao, K. Kaji, D. Yamamoto, and H. Tomobe. Discussion Mining: Annotation-Based Knowledge Discovery from Real World Activities. *Advances in Multimedia Information Processing - PCM 2004, LNCS, Vol. 3331*, pp. 522-531. Springer, 2005.
- [2] R. Barzilay, M. Lapata. Modeling Local Coherence: An Entity-based Approach, *Computer Linguistics, Vol. 34, No. 1*, pp. 1-34, 2008.
- [3] T. Schultz, A. Waibel, M. Bett, F. Metzger, Y. Pan, K. Ries, T. Schaaf, H. Soltan, W. Martin, H. Yu, and K. Zechner. The ISL Meeting Room System, *Proc. of the Workshop on Hands-Free Speech Communication (HSC-2001)*, 2001.
- [4] T. Nishida. Conversation quantization for conversational knowledge process. *Int. J. Computational Science and Engineering, 3(2)*, 134-144, 2007.
- [5] K. Nagao, K. Inoue, N. Morita, and S. Matsubara. Automatic Extraction of Task Statements from Structured Meeting Content. In *Proceedings of the 7th International Conference on Knowledge Discovery and Information Retrieval*, 2015.
- [6] B. Settles. Active learning literature survey. *Computer Sciences Technical Report 1648*, University of Wisconsin-Madison, 2010.
- [7] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference, Vol. 90*, pp. 227-244, 2000.
- [8] A. Liu, L. Reyzin, and B. D. Ziebart. Shift-pessimistic active learning using robust bias-aware prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.
- [9] B. Settles and M. Craven. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2008.
- [10] F. Watanabe, Y. Mori, and C. Kogo. Analyzing Learners' Subjective Evaluation of Peer Assessment in Japan Massive Open Online Courses. *Waseda Journal of Human Sciences, Vol. 28, No. 2*, pp. 237-245, 2015.