August 15, 2001
RT0422
Computer Science 10 pages

Research Report

Discussion Mining: Knowledge Discovery from Online Discussion Records

Akiko Murakami, Katashi Nagao, Koichi Takeda

IBM Research, Tokyo Research Laboratory
IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato
Kanagawa 242-8502, Japan

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Discussion Mining: Knowledge Discovery from Online Discussion Records

Akiko Murakami, **Katashi Nagao**, **Koichi Takeda**
IBM Research, Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato, Kanagawa 242-8502, Japan
{akikom, knagao, takedasu}@jp.ibm.com

Abstract

For the last decade, text mining techniques, which discover knowledge from large amounts of text data, have been making a significant progress. While text mining techniques are useful for independent texts, they don't work well for email and BBS (bulletin board systems) discussions because of the incompleteness of each individual message. Therefore, we need a new method to extract useful information from these kinds of text fragments. This paper describes methods of structuring discussion records or logs, such as email and BBS messages, allowing knowledge discovery using various retrieval and visualization techniques. Each message mainly consists of quotations and comments, but based on the quote-comment relationships, our proposed system generates a Thread Summary which is an abstraction of the ongoing discussion on a particular topic. We consider the Thread Summary to be a coherent document and produce customized summaries to allow users to find their topics of interest more easily. Also, our system can recognize an authority, a person who has a good knowledge in a particular field based on patterns of message exchange.

1 Introduction

Text mining [8] has made a remarkable progress in the business intelligence applications. In particular, we have recognized that incoming customer email (inquiry) can be accurately categorized, Frequently Asked Questions (FAQs) can be retrieved for semi-automatic composition of a reply for the inquiry, and at the backoffice, the accumulated inquiries can be analyzed to discover emerging problems and trends of popular products/topics.

Discussion records or logs, however, have not been tackled by text mining even though they are one of the most heavily used forms of Internet and intranet communications. This is because they have subtle dependencies upon each other to form a context, and hence individual record carries a limited amount of information, which leads to a poor performance of text mining techniques.

Discussion records, such as Web BBS records and mailing lists, contain a great deal of human knowledge. In such discussion records, several subjects may be discussed in one record and each record may not have an appropriate "subject" or "title". Therefore, when we want to know the recent trends from discussion records, we cannot easily find them because each subject may be spread out over several records. We call such a possibly dispersed subject as one "topic". Another difficulty in finding knowledge from discussion records is that several aspects of a single topic may be found in various records. Therefore, we cannot retrieve the particular topic by retrieving certain discussion records in isolation. We will find a similar difficulty when fragments of several topics are mentioned in one record. In this case, we have to divide the discussion records according to the topics.

We use the quotation information to recognize the topic in the records, and reconstruct a new document called a "Thread Summary" that describes a certain branch of the topic by concatenating the fragments of discussion records. We can find individual topics and trends using search and mining techniques over the Thread Summaries. This approach is called "Discussion Mining". Since Thread Summaries also captures human (sender-receiver) relationships, we can identify "Authority" and "Hub" person from a collection of specific type of Thread Summaries.

2 Previous Work

In this section, we introduce some research for various texts. How we will focus upon text from the usability perspective. First, there is conversational data, such as a call center's log or answers to inquiries. While it is not expensive to create this kind of data, the data is not very meaningful. The research field of text mining [8] focused on extracting meaningful knowledge from such data. The researchers try to retrieve "concepts" from huge documents and try to discover trends of the topics in the documents using various data association algorithms. These algorithms do not use raw text data, but after finding a trend, one can find the raw data supporting the trend in order to understand what the key point is or to make a FAQ (Frequently Asked Questions) from the documents. On the other hand, there are formally-written documents, such as a newspaper articles, dictionaries. In this field, it is very expensive to create, however each document can be considered as a complete topic, so these kinds of text can easily be reused. Document threading [10] is to construct documents such as newspaper articles when the user wants to know the topic flow. Document annotation and semantic transcoding [7] are also methods to make well-written documents more accessible.

While there is an area of research called information extraction, it is not directly applicable to such discussion records. One message is a part of a larger topic, so when one message is taken out as a result of the information extraction, we cannot understand the whole topic without referring to other messages. When the data from discussion records are considered as a collection of documents, text clustering such as the e-classifier [?] approach can be used for the extraction of a topic. However, it is not guaranteed that clustering performed properly when the entire topic is never described in any single message.

For the text of discussion records' area, various approach are focused to find the main subject in the records. Murakoshi [5] discussed how to separate topics from discussion records using quotation. They thought the discussions continue when quotations are occurred, and the continuances of the quotation are thread of the subjects.

3 Structured Communication

In the introduction, we noted that one topic can spread out in several messages in the discussion records, mailing lists and BBSs. The message is continuity as a *thread* was formed by replay to previous messages, however these *thread* may not represent one topic, and new topics are usually found in the thread. Moreover, the thread can split into several topics, and because of not replying the messages sometimes same topics exist in different threads.

Hence we define the relationship between two messages using *quotation* and *commentonaquotation*, replacing the relationship of $Reply-To$, and using this information we make a graph structure of the discussion records. We call it $DiscussionGraph$. From this, we can make a summary of the topic, a $ThreadSummary$. In this section we describe how to make a Thread Summary and how to visualize a Discussion Graph.

3.1 Quotation

In the *thread* structure, two messages may be related to each other, however actually only parts of the messages construct topics. So it is possible that a quotation and a comment (on the quotation) are linked together. In a mail message, the previous message is usually quoted with a special character, like ">". Figure 1 shows a pattern of quotation in a message. By recognizing the quotation characters, we can decide the location of previously quoted segments. Moreover the next paragraph just after a quotation is usually regarded as the comment on it. We also define types of comment, for instance, *Question* and *Answer*.

For this perspective, a comment on a quotation expresses the connection between messages. However, due to the differences among the mail programs, the symbol ">"may not necessarily indicate a quotation. Recently the form of attaching the previous messages at the end of message has become very popular. This makes it difficult to judge what part is a quotation or a comment in the message. In the worst case, there may not be any quotations.

3.2 Discussion Graph

When a quotation from a previous message exists in a message, it will form a *link* between the two messages. Each link in a Discussion Graph starts and ends at some point in the messages, indicating the quotation and the comment for the quotation. So we can consider many patterns of the links in the discussion records:

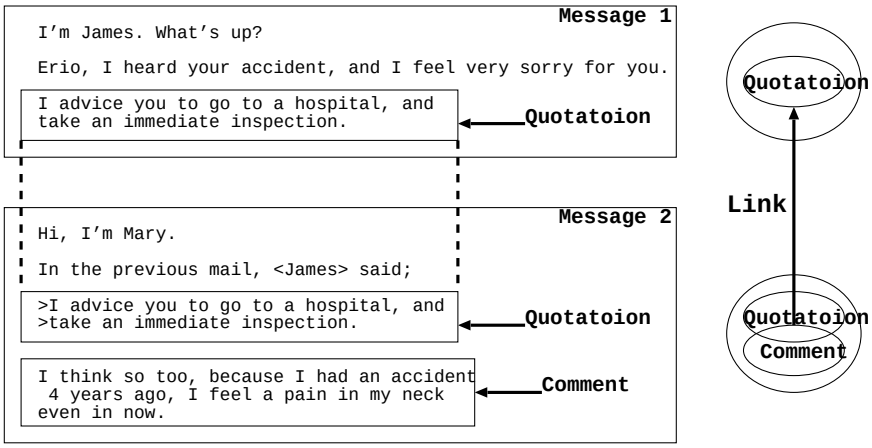

Figure 1: Quotation pattern of emails

3.2.1 Two messages have several links

in this case, two messages have several links

3.2.2 One message is quoted by several messages

in this case, we have a tree structure with links from a referenced messaged to each of the referring message

3.2.3 Several messages are quoted by one message

in this case, there will be a link from each referenced message to the referring message

A Discussion Graph can be expressed by these three link types, so the discussion records are regarded as graph structures like those shown in Figure 2. In this figure, the square-text-fields are message node, and the links are formed between the message nodes. The subjects of the messages are shown within the message nodes, and the links have attributes, like *Question*, *Answer*, *Opinion*...and so on.

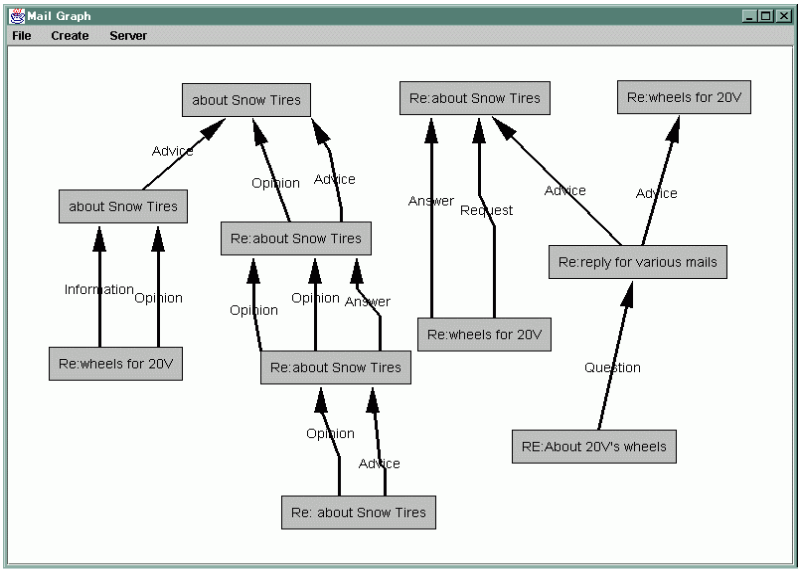

Figure 2: Discussion Graph

3.3 Thread Summary

Since the pair of a quotation and a comment forms a link, we can extract the summaries of the messages using this information, because the continuity of the quote-comment relationship is regarded as the binding to

the actual topic in the messages.

Figure 3 indicates how to make a Thread Summary from this quote-and-comment information. Because of the quotation, there is a relationship between Message A and Message B, and if in another message, the author quoted a part of the Message B's comment region, then a topic flow is recognized.

Extracting these continuations of the messages as a topic, these parts of the messages form a new document, a Thread Summary.

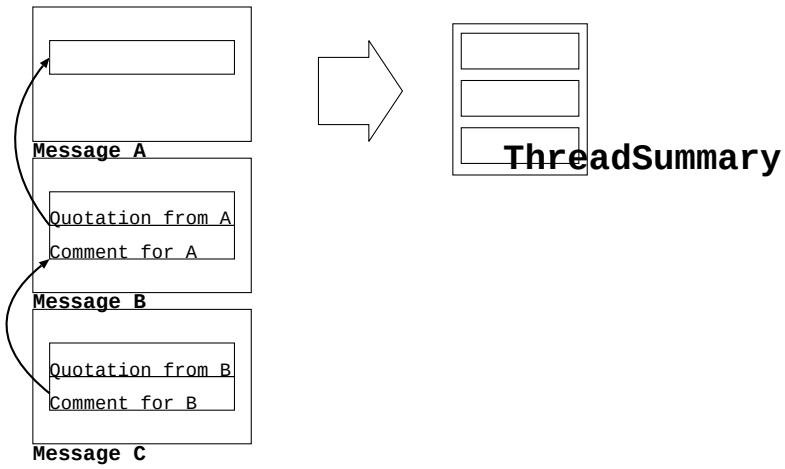

Figure 3: Thread Summary

4 Message Representation using XML

In this section we describe how to extract the quotation information out of the messages of the discussion records.

4.0.1 XML format of message

We convert the messages to an XML (Extensible Markup Language, http://www.w3.org/XML/) format to more easily extract quotation and comment information. We use structured data for the messages which includes the following meta-data.

1. Message ID in the discussion records
2. Subject
3. Date and Time
4. Author

Each message has a unique ID, such as the mail ID in the mailing list. In the body of the message, each sentence is also given a sentence ID. The quotation parts refer back to the original message using the message ID and the sentence ID. In addition, comment parts have comment-type attributes.

4.0.2 Extracting quotation information from messages

We want to extract the quotation information from the structured messages. This information is described in an XML compliant language which we call XM^2L, using this information, we are able to visualize the graph structure of messages.

4.0.3 Bulletin Board System(BBS) for automatic tag annotation

A BBS (Bulletin Board System) used for the asynchronous discussion, usually has the function for quoting previous messages. However, as with mail messages, recognizing the quotation and the comment can be very difficult.

```
<?xml version="1.0" encoding="Shift_JIS"?>
<XM2L>
<message id="564" file="message01.gda">
  <subject>On security of
             computer networks</subject>
  <date>Tue, 10 Nov 98 22:03:43 +0900</date>
  <author>Akiko Murakami</author>
</message>
<message id="569" file="message04.gda">
  <subject>Re:On security of
             computer networks</subject>
  <date>Wed, 11 Nov 1998 10:36:07 +0900</date>
  <author>Katashi Nagao</author>
  <quote type="Opinion" ref="564-3">
    <link ref="564"/>
  </quote>
  <quote type="Advice" ref="564-7 564-8">
    <link ref="564"/>
  </quote>
</message>
.
</xm2l>
```

Figure 4: XM^2L

To find where the quote or comment is located, we suggesting a new kind of BBS, which automatically insert information about quotations and comments into the original message in the form of annotations. Then, the system can visualize the structural information based on the annotations.

5 Discussion Mining

So far we have outlined a way to visualize topics as a graph structure of discussion records from the retrieved topics. The entire process of extraction, retrieval, visualization, and analysis of topics (Thread Summaries) is called Discussion Mining.

5.1 Searching Thread Summaries

There are three ways for searching for Thread Summaries.

5.1.1 Full text searches

This is basically a keyword search, such as a search for topics which contain both the word "A" and "B." Sometimes the keywords may be contained in separate messages. If we use a keyword match for the words "A" and "B" for each message, we may get no answer even though the relevant topic exists. When we know what to look for in the discussions, we just search the Thread Summaries because they represent the set of complete topics.

5.1.2 Searches based on link types

As mentioned earlier, each link has a comment type in its Discussion Graph. This comment type is very useful to recognize the continuation of the topic. We define six types of comments, Opinion, Question, Suggestion, Objection, Request, and Answer. For example, we can search for topics containing the word "A" and suggestion links.

5.1.3 Tag-based search

In the previous section, we mentioned that the messages are described in XML format. Using XML tag information, advanced searching is possible. For example, we can find the Thread Summaries containing a keyword in the `<SUBJECT>` tag and another keyword in the `<BODY>` tag.

For deep understanding of the semantic structures of messages, we annotate them with a new tag set. The new tag set was proposed by the GDA (Global Document Annotation) Project [3]. It is based on XML and designed to be as compatible as possible with other annotation/markup tag sets such as HTML, TEI [9], CES [1], EAGLES [2], and LAL [11].

Annotation data is semi-automatically generated using natural language processing techniques. An example of a GDA-tagged sentence is as follows:

```
<su><np rel="agt" sense="time0">Time </np>
<v sense="fly1">flies </v>
<adp rel="eg"><ad sense="like0">like </ad>
<np>an <n sense="arrow0">arrow</n></np>
</adp>.</su>
```

`<su>` means sentential unit. `<n>`, `<np>`, `<v>`, `<ad>` and `<adp>` mean noun, noun phrase, verb, adnoun or adverb (including preposition and postposition), and adnominal or adverbial phrase, respectively[1]

The `rel` attribute encodes a relationship in which the current element stands with respect to the element that it semantically depends on. Its value is called a relational term. A relational term denotes a binary relation, which may be a thematic role such as agent, patient, recipient, etc., or a rhetorical relation such as cause, concession, etc. For instance, in the above sentence, `<np rel="agt" sense="time0">Time</np>` depends on the second element `<v sense="fly1">flies</v>`. `rel="agt"` means that *Time* has the agent role with respect to the event denoted by *flies*. The `sense` attribute encodes a specific word sense.

5.1.4 Scoring search results

To evaluate how a search result matches to a query, we employ a scoring function considering the occurrence count of keywords in a Thread Summary, the value for link type consistency, and the degree of structural correspondence between the query sentence and a sentence in each candidate message.

5.2 Visualization of Search Results

The visualization of search results are displayed as a Discussion Graph. From this visualization, we can get the related data from the selected Thread Summary. This visualization allows the user to easily check the relevance of the results.

5.2.1 Visualization using graph structure

A Thread Summary consists of fragments of messages (called submessages) and quote-comment links between those messages (corresponding links to the Discussion Graph). There is at least one Discussion Graph which includes the retrieved Thread Summary. The Discussion Graph sometimes includes different Thread Summaries, called "neighbor" Thread Summaries. The neighbor Thread Summaries share some submessages with the original Thread Summary. They may be very relevant to the user's interest because they have some additional information about the retrieved Thread Summary.

For instance, there are two Thread Summaries in Figure 5, which includes some submessages in common. Therefore, one is a neighbor Thread Summary of the other.

Among neighbor Thread Summaries, it is probable that the query-matching scores will be different, because the keyword occurrence counts will most likely be different. It is found in many cases that the Thread Summaries which have only a small part in common are affecting each other, but the Thread Summaries in the same Discussion Graph are related.

To perform a search using data which is reconstructed based on the dependencies, it is much more effective that we enumerate the graph itself visually as peripheral information rather than display only the items in a ranked form when we show the results. Moreover, by using the graphic view, and not the enumeration of

[1] A more detailed description of the GDA tag set can be found at `http://www.etl.go.jp/etl/nl/GDA/tagset.html`.

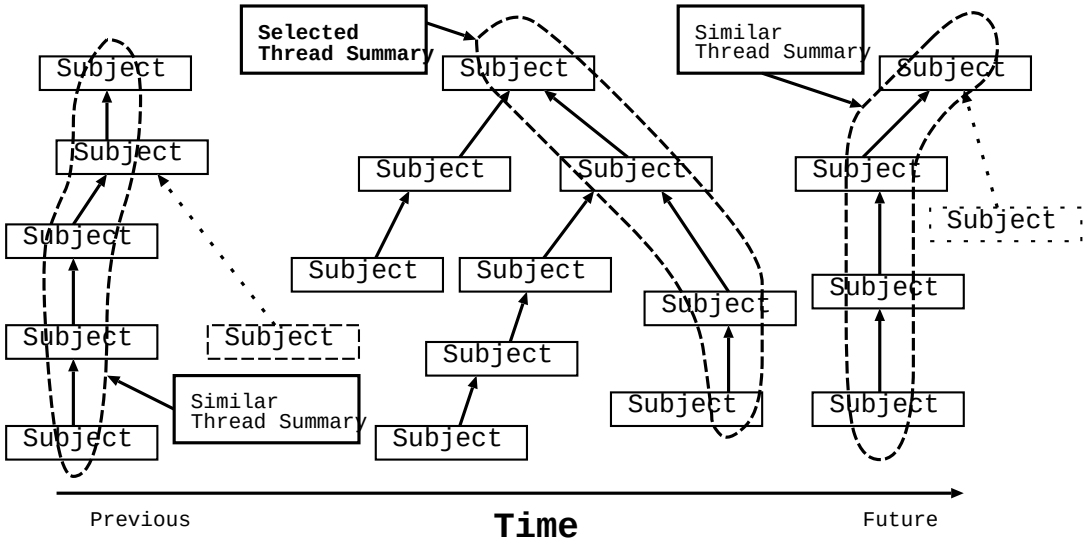

Figure 5: Example of neighbor Thread Summaries

Thread Summaries in the same graph, we can display the results more compactly. For these two reasons, when we display Thread Summaries as the results of a search, not only the Thread Summary itself, but also the whole graph in which the Thread Summary exists is displayed and linked.

5.3 Summarizing Thread Summaries

When several candidates of interested Thread Summaries are retrieved, the user wants to select the most appropriate one. Therefore, our next step is to make each Thread Summary more readable. We can deal with a Thread Summary as a coherent document describing a particular topic. We apply an automatic document summarization technique to generation of a shorter version of each Thread Summary. Then, our system constructs a collection of digests of Thread Summaries. The order in the collection of summaries depends on the matching scores.

5.3.1 Document structure of Thread Summary

As mentioned earlier, messages are structured with linguistic annotations. Thread Summaries inherit their original messages' linguistic structures. A linguistically-annotated document naturally defines an intra-document network in which nodes correspond to elements and links represent the semantic relations. This network consists of sentence trees (syntactic head-daughter hierarchies of subsentential elements such as words or phrases), coreference/anaphora links, document/submessage/quotation/comment nodes, and rhetorical relation links.

5.3.2 Summarization algorithm

Our text summarization method employs a spreading activation technique to calculate the importance values of elements in the text [6]. The summarization algorithm works as follows:

1. Spreading activation is performed in such a way that two elements have the same activation value if they are coreferent or one of them is the syntactic head of the other.
2. The unmarked element with the highest activation value is marked for inclusion in the summary.
3. When an element is marked, the following elements are recursively marked as well, until no more elements are found:
   - the marker's head
   - the marker's antecedent
   - the marker's compulsory or *a priori* important daughters, the values of whose relational attributes are `agt` (agent), `pat` (patient), `rec` (recipient), `sbj` (syntactic subject), `obj` (syntactic object), `pos` (possessor), `cnt` (content), `cau` (cause), `cnd` (condition), `sbm` (subject matter), etc.

- the antecedent of a zero anaphor in the marker with some of the above values for the relational attribute

4. All marked elements in the intra-document network are generated preserving the order of their positions in the original document.

5. If a size of the summary reaches the user-specified value, then terminate; otherwise go back to Step 2.

The size of the summary can be changed by simple user interaction. Thus the user can see the summary in a preferred size by using an ordinary Web browser without any additional software. The user can also input any words of interest. The corresponding words in the document are assigned numeric values that reflect degrees of interest. These values are used during spreading activation for calculating importance scores.

6 Authority Finding using Graph Structure Information

We want to understand the knowledge existing in the discussion records, however the answer is not always described explicitly in the data. Thus, based on the data, if we can determine who is the most knowledgeable person on the topic of interested, we can get the answer from that person. The method that we use to identify the appropriate person is called authority finding.

Kleinberg et al. proposed a model of authority finding for WWW pages [4]. In this model, the authorities are not person, but World Wide Web pages. Their model is based on the relationships between the authority pages for a topic. The model also considers pages that link to many related authority pages, which are called hubs. Their model of hubs and authorities on the Web is defined by the hyperlink structure. A good hub is a page that points to many good authorities, and a good authority is a page that is connected to many good hubs.

In Discussion Graphs, a message can be considered as a node, discussion records can be converted into hypertexts and therefore we can analyze them using the same methods as for hub and authority structures. Moreover, we can get richer knowledge using nodes' attributes (such as authors) and the link types (Question, Answer, etc.).

The message is referred to several messages, this message is considered as raising an issues or issues about something, so the node has a function of a hub. On the contrary, the message that refers to several messages functions as an authority.

In the left side graph in Figure 6, the message is referred to by several messages, this message is considered as raising an issue about something, so the node has the function of a *hub*. In the right side graph in Figure 6, the message that refers to several messages functions of as an *Authority*.

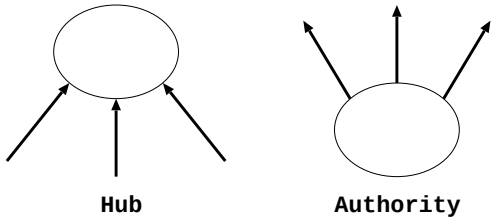

Figure 6: Message Hub and Authority

Each node is scored as an Authority N^A and as a Hub N^H. The scores are defined as follows:

$$N_H = \sum \frac{W_{in}}{d} \times type_{in}$$

$$N_A = \sum \frac{W_{out}}{d} \times type_{out}$$

$$N = N_A - N_H$$

W_in/out are the number of links which $comeinto$ (referred to by) or $gooutfrom$ (refer to) one node (message). If the node is in the Thread Summary, all of the links in the Thread Summary are counted. However, the links' distance from the node reduced its value for the scoring. The distance is expressed as d.

$type_{in}$ and $type_{out}$ are the comments' type parameters. The value of a parameter changes depending on whether the link starts from it, or enters it, and depending on the links' attributes (Question, Answer, ... etc.). The parameter values are shown in Table **??**:

	link(out)		link(in)	
	Authority	Hub	Authority	Hub
Question	0	1	0	0
Answer	1	0	0	1
Opinion	0.5	0	0	0.5
Suggestion	0.5	0	0	0

Figure 7: The type parameters

Opinion and Suggestion attributes are less influential on the Hub/Authority's scores than Question and Answer attributes, so the value of such parameters halved.

Thus, we can get Hub and Authority score N using each score. This shows a Authority Node or a Hub Node or an Ordinary Node, using this score N.

Since identifying the authority for the topic is the main concern, we can find the authority of the topic by considering the nodes' Hub scores and Authority scores. The summation of the nodes' scores for each person leads to the Hub or Authority score of that person. Using this method, we can derive a profile for each author which indicates whether the author is a "Hub person" or a "Authority person" for some specific topics.

7 Concluding Remarks

We have introduced a technique for restructuring the discussion records using the information in quotations and related comments. In online discussion logs, one message sometimes has several topics, and also the topics may be distributed among several messages. This technique is useful to recognize one topic as the Thread Summary in such discussion records.

We have also shown that Thread Summaries accurately represent individual topics, so knowledge may be retrieved from discussion records by performing searches on the Thread Summaries, or by using text mining methods with the Thread Summaries as the basic document units. In addition, visualization of the Discussion Graph helps users to search for discussion records.

We also have discussed authority finding based on structural analysis of the link topology surrounding authoritive message nodes on the topic. Based on this idea, we are also planning to associate scores indicating the reliability of the information using the authors' reliability scores in the Discussion Graph.

Acknowledgements

The authors likely to acknowledge Kevin M. Squire, Matthew Hurst, and Ryuichiroh Higashinaka for their helpful comments and conversation. We also acknowledge the excellent advice of Yukiko Katagiri for the design of the graph structure.

References

[1] Corpus Encoding Standard (CES): Corpus Encoding Standard. http://www.cs.vassar.edu/CES/.

[2] Expert Advisory Group on Language Engineering Standards (EAGLES): EAGLES online. http://www.ilc.pi.cnr.it/EAGLES/home.html.

[3] Koiti Hasida: Global Document Annotation. http://www.i-content.org/GDA/.

[4] Jon M. Kleinberg: Authoritative Sources in a Hyperlinked Environment. *JACM 46(5)*, 604-632 ,1999.

[5] Hiroyuki Murakoshi and Koichiro Ochimizu: Influence of Conversational Coherency of E-mail Communication in Successful Cooperative Software Development, In Joint Workshop of ICSE98 on Human Dimensions in Successful Software Development, 1998.

[6] Katashi Nagao and Koiti Hasida: Automatic text summarization based on the Global Document Annotation. In *Proceedings of COLING-ACL'98*. 1998.

[7] Katashi Nagao et al.: Semantic Transcoding: Making the World Wide Web more understandable and usable with external annotations. *IBM Research Report*. 2000.

[8] Tetsuya Nasukawa, Masayuki Morohashi, Tohru Nagano: Customer Claim Mining: Discovering Knowledge in vast amounts of textual data, *IBM Research Report*. RT0319, 1999.

[9] The Text Encoding Initiative (TEI): Text Encoding Initiative. http://www.uic.edu:80/orgs/tei/.

[10] N. Uramoto and K. Takeda: A Method for Relating Multiple Newspaper Articles by Using Graphs, and Its Application to Webcasting, In *Proceedings of COLING-ACL'98*. 1998.

[11] Hideo Watanabe: Linguistic Annotation Language: The markup language for assisting NLP programs. *IBM Research Report RT0334*.1999.