TURNING UNSTRUCTURED AND INCOHERENT GROUP DISCUSSION INTO DATREE: A TBL COHERENCE ANALYSIS APPROACH

Completed Research Paper

Shasha Deng\textsuperscript{1,2}
\textsuperscript{1} Department of Management Information Systems, Shanghai Jiaotong University
\textsuperscript{2} Department of Computer Science, Shanghai University of Electric Power
1. No.535 Rd.FaHuaZhen, China
2. No.2103 Rd. Pingliang, China
dengshasha@sjtu.edu.cn

Pengzhu Zhang
Department of Management Information Systems, Shanghai Jiaotong University
No.535 Rd.FaHuaZhen, China
pzzhang@sjtu.edu.cn

Yilu Zhou
Information Systems and Technology Management, George Washington University
Funger, Suite 515, USA
yzhou@gwu.edu

Abstract

Despite the rapid growth of user-generated unstructured text from online group discussions, business decision-makers are facing the challenge of understanding its highly incoherent content. Coherence analysis attempts to reconstruct the order of discussion messages. However, existing methods only focus on system and cohesion features. While they work with asynchronous discussions, they fail with synchronous discussions because these features rarely appear. We believe that discussion logic features play an important role in coherence analysis. Therefore, we propose a TCA method for coherence analysis, which is composed of a novel message similarity measure algorithm, a subtopic segmentation algorithm and a TBL-based classification algorithm. System, cohesion and discussion logic features are all incorporated into our TCA method. Results from experiments showed that the TCA method achieved significantly better performance than existing methods. Furthermore, we illustrate that the DATree generated by the TCA method can enhance decision-makers’ content analysis capability.

Keywords: Discussion logic feature, coherence analysis, asynchronous group discussion, subtopic segmentation
**Introduction**

The Internet has dramatically changed our manner of communication. With users spread around the world, the need to organize group discussion via synchronous or asynchronous methods to support business decisions has increased (Nash 2005). Many customized systems such as group support systems (GSS) allow virtual work groups to participate different discussion regardless of physical boundaries. E-mail, forums and other online communities enable us to collaboratively work to seek solutions and share expertise with colleagues and even strangers at any time. Many institutional websites like blogs, chat rooms and instant message systems allow us to converse with old friends and new people without face-to-face interaction.

In spite of the numerous benefits and convenience of group discussion through the Internet, the fact that such user-generated texts create ‘information overload’ cannot be ignored (Sheridan and Ferrell 1974). Although computers are good at processing structured data, the ability to process unstructured textual data is still a challenge. Traditional research on text summarization, text categorization and information extraction aids analysis of unstructured data to some extent. However, these discussion texts face a different challenge: they are not only unstructured but also incoherent (Zechner 2002). Scholars call this “information entropy,” meaning incoming messages are not sufficiently organized by topic and the content cannot be easily comprehended (Hiltz and Turoff 1985). Because the sequence of turns during multi-user group discussion is disturbed in online group discussions, it is more difficult/harder to understand the conversation content. Besides, intensive interactions among a large number of participants complicate the problem.

Current text-based forms of group discussions, including e-mail, newsgroups, forums, chat rooms and GSS, closely resemble spoken interaction. In contrast to spoken interaction, these systems often do not contain the coherent message relationships. Quantitative research experiments suggested that a structured group discussion would improve decision-making outcomes and efficiency (Farnham et al. 2000). Conversation trees are a manifestation of structured information that structure conversations in a more coherent way (Herring 1999). To construct such a tree structure, it is important to understand the sequence of messages and their reply-to relationships (Smith et al. 2000). Some group discussion analysis systems have been developed to address these needs (Viegas and Donath 1999; Sack 2000; Yee 2002; Eklundh and Rodriguez et al. 2004; Fu et al. 2008). However, they generally identified correct turn adjacency by using explicit system features such as header information and quotations. Very little research taps into message body text to gain social cues to improve text analysis capabilities (Fu et al. 2008). Moreover, while system features and social cues work for e-mails, newsgroups and forums, they do not apply to chat rooms and group discussions. However, little research has explored automatic analysis of online group discussion due to its unstructured and incoherent nature.

Some group discussion texts may contain very few system features and social cues that can help structure texts. They are however rich in logic cues such as argument process and utterance emotions (Raghu et al. 2001). In this study, we propose a TBL Coherence Analysis (TCA) system to turn unstructured and incoherent group discussion texts into Discussion Analysis Trees, or DATrees. TBL refers to Transfer-Based-Learning, a machine learning algorithm. DATree reorganizes unstructured discussion texts by automatically discovering subtopics and identifying reply-to relationships between messages. TCA attempts to address limitations in previous studies by utilizing a holistic feature set which is composed of not only explicit linguistic social cues and system features but also implicit discussion logic cues. Combined with various feature types and customized machine learning algorithms, the TCA system is able to capture important and implicit discussion logic cues for better performance. The TCA system is composed of feature extraction, subtopic segmentation and TBL-based classification. Our experimental results on various themes of group discussions showed that the TCA system can transfer multiple implicit features into decision rules and identify the correct reply-to relationship.

The remainder of the paper is organized as follows. We first compare different characteristics of text-based group discussion systems and provide a review of methods developed to support discussion text analysis. We then present the many facets of application in discussion text analysis and describe challenges associated with group discussion text. Subsequent sections provide an overview of our system.
design and elaborate its components. We then present experimental evaluations of the proposed system. We conclude with our research contributions and future directions.

Related Work

Although the Internet provides us with an unprecedented opportunity for remote group discussions, many studies have expounded upon the significance of analyzing group discussion text to understand online discourse patterns (Chia 2000; Wilson and Peterson 2002; Abbasi and Chen 2008). They emphasized that accurate use of analytic and flexible tools can greatly help people cheaply and handily grasp valuable information from gigantic and wild digital data (Paccagnella 1997). Unfortunately, the unstructured and incoherent nature of discussion texts is a big obstacle to such research. In this section, we describe the core obstacles to analyzing unstructured discussion text, review previous coherence analysis research and analyze the benefits of tree structure in text presentation.

Importance of Coherence in Unstructured Group Discussion Text

In spoken and face-to-face conversations, the sequential structure of turn-taking facilitates coordination among multiple speakers (Clark 1991). However, in online group discussion, the sequence of individual messages is determined by the server’s receiving time. The order of turn-taking is disrupted because there are considerable time lags between when a message is sent and when it is responded to (Cherny 1995). With the greater openness of group discussion, the gap becomes bigger and the unstructured degree is higher (McGrath 1990; Kuechler 2007). Herring (1999) pointed out two properties of this medium: “lack of simultaneous feedback” and “disrupted turn adjacency.”

Resolving the disrupted turn adjacency problem remains an arduous yet vital endeavor (Fu et al. 2008). In a multi-participant conversation, the “adjacency pairs” structure will contribute to discussion text coherence (Schegloff 1968). Hence, once turn adjacency is disrupted, users may have difficulty tracking message sequence in a computer-mediated environment. Plenty of previous research has observed this phenomenon. Nash (2005) manually analyzed 1,099 turns from Yahoo! Chat and found the gap between a message and its response can be as many as 100 turns. Herring and Nix (1997) safely concluded nearly half of all turns were “off-topic.” McDaniel et al. (1996) were surprised at how often thread confusion occurred.

Additionally, group discussions often encompass two or more parallel topics in a conversation. Besides topic fragmentation over time, multiple competing new directions (subtopics) are also characteristic of group conversation (Herring 1999). Group discussion systems define the order of the messages as they appear in the discussion window. As a result, some subtopics are twisted and mixed together confusing participants about which previous message is being referenced (Holmer et al. 2009). Lambiase’s (2010) study showed that topic delay exists even in group discussions with a high control of being ‘on-topic’. This problem is more severe in synchronous group discussions than asynchronous ones.

Coherence Analysis for Group Discussion Text

Text comprehension involves constructing a coherent mental representation of situations described by the text. Coherence is often used to refer to representation of mental relationships while cohesion represents the textual indications of coherent representations (Louwerse 2002). While cohesion seeks the answer in overt textual signals, a coherence approach considers connectedness to be of a cognitive nature (Sanders and Maat 2006). For web discourse, “coherence is represented in terms of coherence relations between text segments, such as elaboration, cause and explanation” (Barzilay and Elhadad 1997). Fu et al. (2008) further pointed out that coherence of online discourse is represented in terms of reply-to relationships between messages. Thus, coherence analysis in group discussion text can be considered as accurately identifying reply-to relationships to construct discussion structure. Various features in communication systems can contribute to coherence analysis.
Various Features in Coherence Analysis

Although almost all forms of group discussion text suffer from disrupted turn adjacency, different forms of discussion text have different degrees of incoherence which require different analysis techniques (Herring 1999). According to participant patterns, the discussion text can be divided into asynchronous and synchronous. Asynchronous discussion text includes e-mail, Usenet newsgroups, forums and blogs. Synchronous discussion text includes chat and customized group discussion systems such as GSS. Our focus is on synchronous group discussions because they are under-researched. We will review features for both asynchronous and synchronous group discussions.

System features. System features are often extracted from asynchronous discussion text. During transmission, e-mails and Usenet newsgroups are required to add header information and specify quoted sections. Using this information, Netscan extract the “contents of Subject, Date, Organization, Lines, MessageID and Reference lines” to generate the relationships in selected newsgroups (Xiong 1998). Smith (2001) found that “when an author posts a reply to a previous message, the new message indicates its parent message by citing the parent’s message ID in its own References header.” Using these references, they can construct an n-ary tree. By detecting quoted text, Zest divides each message into contiguous blocks of quoted or unquoted text (Yee 2002).

Although header information and quotations are effective in identifying reply-to relationships, it is difficult to generalize. On one hand, not all forms of discussion text contain system features, especially synchronous text. On the other hand, users do not always use the system features to refer to the previous message. Consequently, using system features alone does not solve our problem.

Cohesion features. Cohesion features include social cues and cohesion-based linguistic features. Previous research has suggested some interesting linguistic features to solve the incoherence problem (Halliday and Hasan 1976).

Direct address appears when a reply message includes the screen name of the author of the previous message. Using the direct addressing method, Coterie constructed an interaction network for conversation (Spiegel 2001). Lexical relation is defined as a “cohesive relation where one lexical item refers back to another, to which it is related by having common referents” (Nash 2005). It is divided into three subcategories: repetition, synonymy and super-ordinate. Lexical relation has been widely used in previous studies. For example, Choi (2000) used repetition of keywords to identify the message relationship. Co-reference describes the situation when a lexical item refers to another lexical item by the context instead of repeating the same item name. Nash (2005) divided co-reference into three subcategories: personal, demonstratives and comparatives. Soon (2001) adopted a machine learning approach to identify co-reference of noun phrases. Bagga (1998) built co-reference chains by CAMP and used the VSM to resolve ambiguities between people with the same names. Other cohesive features such as conjunctions, substitution and ellipsis have rarely been incorporated in previous research due to the difficulty in identification. All of these features rely on linguistic analysis. Asynchronous text is particularly rich in cohesion features, since asynchronous discussion allows users to think and edit their messages with no time pressure (Fu et al. 2008).

Cohesion features provide more evidence than system features. However, it is not without pitfalls. First, some of these features are hard to extract automatically from the message body. Identifying the relation between the antecedent and the anaphor is a complex task in co-reference (Soon et al. 2001). Secondly, using these cohesion features alone cannot identify all reply-to relationships. Lastly, not all group discussion text contains completed cohesion features. Direct address features rarely appear in a synchronous group discussion system, when participants pay more attention to solving the problem under great time pressure (Paul and Nazareth 2010). Moreover, lexical term repetition is directly used in previous studies (Bagga and Baldwin 1998). It is not clear if two messages with reply-to relationship should be the most similar ones.

Discussion logic features. Discussion logic features are decision-making cues and context-based features. They have not been used in previous coherence analysis research but we believe they are especially important in synchronous group discussions. Toulmin’s model of discussion revealed the nature of the argument process, especially in tracing the solver’s line of discussion (Toulmin 1958). Based on Toulmin’s model, Raghu et al. (2001) modeled collaborative decision-making as a dynamic process in
which individuals assert their positions through both primitive and derivative statements. A primitive statement is a stand-alone assertion and a derivative is obtained as a strictly logical or defeasible consequence of others. Although an individual could make a primitive assertion, a cogent argument requires the assertions to be linked and organized in some logical sense. So the subsequent logical argument structure proceeds in the form of challenge/response exchanges between its proponents and opponents.

The argumentation process contains the following steps: 1) bring forth a candidate subtopic or solution; 2) demonstrate or refute the subtopic or solution; 3) raise question; and 4) provide some information or fact. This logic can be traced by multi-agent automatic systems (Carbogim et al. 2000). Furthermore, based on/using a simulation experiment for four types of decision problems, Carbogim (2000) found a group discussion can be represented in a two-step process: “first, arguments are generated; then, arguments are evaluated in terms of their acceptability.” Obviously, the inner procedure of one group decision discussion is a repetitive process of subtopic or solution-generating and its evaluation. We refer to the inner procedure as discussion logic features. How to automatically identify the proposed subtopic and its evaluation is a big challenge for discussion logic features extraction (Braak et al. 2008). We still believe that discussion logic features can be extracted to help identify the reply-to relationships.

Coherence Analysis Techniques

Using the above-mentioned features except for discussion logic features, many techniques have been used in coherence analysis, especially to construct reply-to relationships. They can be classified into four major categories: manual, linkage, heuristic-based and classification-based methods.

Nash (2005) manually identified some linguistic features, including lexical relations, direct address and co-reference, for online discussions. Barcellini (2005) manually analyzed quotation practice in the online discussions of Open Source Software design which took place in mailing list exchanges. Chesnevar (2006) described specifications for an argument interchange format to represent data exchange between various argumentation tools. Turoff (1999) and Rienks (2005) diagrammed argument process in discussions based on three models. Chesnevar (2006) described specification of an argument interchange format (AIF) to present discussion content based on linguistic words and the Toulmin model. In general this method can achieve high accuracy, but is only possible with small datasets.

Linkage methods build interaction trees with the system features and some predesigned rules (Newman 2002; Yee 2002; Smith and Fiore 2001; Sack 2000). The form of discourse that works well with this method is e-mail-based discussion lists. Some typical systems are Netscan (Usenet group analysis tool, 2001), Conversation Map (2000) and Zest (2002). Obviously, the linkage method is relatively easy to implement, but it requires that users follow system features to post messages and clearly quote messages being responded to.

Heuristic-based methods use both explicit system features and implicit cohesion features to construct a set of rules and matching algorithms to identify the interactional pattern. For example, HIC utilized both system features and linguistic features. Furthermore, several similarity-based methods have been applied to perform lexical match and even residual match (Fu et al. 2008). Murray (2007) proposed to use term-weighting in open group discussion to exploit interactional patterns. By employing pattern recognition using finite state automata, one can automatically identify patterns of interactions with both social and semantic terms in multi-person and multi-topic chat rooms (Khan et al. 2002). This method is effective in text summarization. However it is prone to being affected by unrelated sentences. Also, it only considers sentence semantics and ignores sentence structures.

In classification-based methods, coherence analysis is formulated as a binary classification problem. For example, in order to handle highly incoherent text from student online forums, Kim et al. (2010) used “speech act” features to classify discussion threads. Soon (2001) adopted a machine learning approach to identify the co-reference of noun phrases. Using statistical features and sentence structure information, they showed that machine learning is effective in binary classification. Machine learning methods need a set of handcrafted training documents. Their performances also highly depend on the features used in classification.
Previous research has provided a good foundation of coherence analysis for group discussions. We categorize these important researches in terms of domains, features and techniques in Table 1. Group discussion text is divided into asynchronous and synchronous (D1 and D2). System features, cohesion features and discussion logic features are indicated by F1, F2 and F3. Techniques include manual (T1), linkage (T2), heuristic-based (T3) and classification-based (T4).

<table>
<thead>
<tr>
<th>Previous Studies</th>
<th>Domain Features</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D1</td>
<td>D2</td>
</tr>
<tr>
<td>Turoff et al. 1999</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sack 2000</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Smith and Fiore 2001</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Soon et al. 2001</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Yee 2002</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Khan et al. 2002</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Newman 2002</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Barcellini et al. 2005</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Nash 2005</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Rienks et al. 2005</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Chesnavr et al. 2006</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Murray and Renals 2007</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fu et al. 2008</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Kim et al. 2010</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Tree-like Structure to Support Understanding of Group Discussions**

Disrupted adjacency turns leads to chaos in group discussion. Finding reply-to relationships can help transform unstructured group discussion texts into some kind of more structured representation. Using/Based on people’s cognition and memory, Schweickert (1993) believes the tree model helps trace the detailed argument process in discussion. Qualitative analysis illustrated that visualization of text structure can show the dynamics of collaboration, disentangle intertwined discussion subtopics and grasp the situation of group discussion at a glance. MuViChat and Mediated Chat are tree-based visualization tools that offer different possibilities to follow and understand group discussion and thereby diminish the chat confusion which often occurs in standard chat systems (Holmer et al. 2009; Fuks et al. 2006). Through analysis of existing visualization tools, such as ArguMed, Convince Me and ReasonAble, research showed that these visualization tools contributed to higher quality discussion and more coherent argument (Braak et al. 2008).

Besides tools that support construction and visualization of a tree-like structure during online group discussions, researchers have also proposed methods to analyze unstructured group discussion afterwards. By capturing connections between turns and replies, a “conversation tree” can present the basic turn-taking structure of human conversations (Smith et al. 2000). Conversation Space and typical Usenet newsgroup browsers use a schematic tree view where a message is linked with replies (Popolov et al. 2000; Smith and Fiore 2001). Conversation Map use tree-like visualization tools to facilitate understanding of social and semantic structure of VLSCs (Sack 2000). Using quotation information in mailing list exchanges, a tree model can reveal links in social structure, critical elements in discussion space and how they shape the OSS design process (Barcellini et al. 2005). Fu et al. (2008) created interaction diagrams for web forums to facilitate enhanced social network and role analysis capabilities.

**Research Gaps and Questions**

Based on our review, we have identified several important research gaps. Firstly, little coherence analysis research has been conducted for synchronous discussion text. Most existing studies focused on asynchronous discussion text. E-mail and newsgroups contain rich system features to capture user interactions, and forums contain rich cohesion features in the message body to identify reply-to relations. Secondly, previous research failed to take internal discussion logic into consideration during coherence
analysis. They focused on system features and cohesion features only. It is questionable to directly apply these features in synchronous discussions such as chat rooms and customized group discussion systems.

Finally, little work has been done to automatically turn synchronous discussion text into a tree structure. Researchers have manually analyzed such discussion text and presented them in a tree-like structure. However, they have not automatically performed coherence analysis. We raise the following research questions:

1. How can we design a system that utilizes not only system and cohesion features, but also discussion logic features in synchronized group discussion coherence analysis?
2. Will discussion logic features improve coherence analysis performance over using system features and cohesion features?
3. What is a good structured representation of synchronous group discussion texts that can potentially enhance one’s analysis capability?

**System Design**

In order to address these research questions, we designed the TBL Coherence Analysis (TCA) system to perform automatic discussion text coherence analysis. The proposed system has two major components: **Subtopic Segmentation** and **Reply-To Relationship Identification** (as shown in Figure 1). Reply-to relationship identification contains two sub-components: Message Feature Extraction and Classifier Construction. The feature extraction stage acquires various system features, linguistic features and discussion logic features from the group discussion messages. Discussion logic features rely on input from subtopic segmentation. All extracted features are passed forward to the classifier construction stage, which involves creation of a TBL classifier for candidate message pairs and then utilizes the predefined rules to determine residual coherence relationships. Finally, the unstructured group discussion messages are formatted to a discussion analysis Tree (DATree).

![Figure 1. TBL coherence analysis system design](image)

**Definition**

Since there are no unified names for certain phrases in GSS and the social media field, it is necessary to provide some definition to avoid confusion in following our system design description.

Definition 1: *message* is one utterance or post. One message can include many sentences posted by an individual author at one time.
Definition 2: discussion is one discussion based on a specific topic. In group discussions, there can be different discussions at the same time based on the same or different topics. A “discussion” refers to one of these group discussions.

Definition 3: subtopic defines a proposal or claim or argument in one discussion. In other words, one discussion has one or more subtopics.

**Data Processing**

Before we perform any analysis, the data processing component is designed to extract message bodies and their related meta-information from Group Support Systems. All relevant property information is extracted and tagged with corresponding symbols. In order to perform deep text analysis, the message body’s syntactic structure is parsed by a tool called Language Technology Platform (LTP) developed by the Harbin Institute of Technology (http://ir.hit.edu.cn/demo/ltp/Sharing_Plan.htm). The parser is good at lexical analysis including word segmentation and part-of-speech (POS) tagging based on dependency grammar. The parser has also showed good performance in distinguishing noun compounds which could greatly influence information extraction accuracy. Finally, all group discussion messages are labeled by property information and lexical structures, such as author screen name, time stamps and other lexical relationships. This step prepares data for system and cohesion feature extraction as well as subtopic segmentation.

**Message-Level Feature Extraction**

The extraction stage involves derivation of structural, linguistic and discussion logic features. Table 2 provides a detailed description of such categories. System features capture static attributes such as current message ID (curMsgID), message’s author (viewer) and message post time (viewTime). Cohesion and discussion logic features are more complicated and are derived from the message body. We describe them in more detail.

<table>
<thead>
<tr>
<th>Group Category Type Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>curMsgID Number The current message’s ID number</td>
</tr>
<tr>
<td>viewer Number The message’s author</td>
</tr>
<tr>
<td>viewTime Number The message post time</td>
</tr>
<tr>
<td>MsgSimList Vector List the similarity between the current message and the other messages</td>
</tr>
<tr>
<td>msgSentiment True/False True if the message includes user’s subjective sentiment. False otherwise.</td>
</tr>
<tr>
<td>subTopicID Vector List all the subtopics of one discussion</td>
</tr>
</tbody>
</table>

**Cohesion Feature Extraction**

Two cohesion features are extracted here: similarity feature (MsgSimList) measures the similarity between the current message and all other messages and sentiment feature (msgSentiment) identifies if the message is subjective or objective.

**Message Semantic Similarity Measure.** Being able to identify semantic similarity between every two messages serves two purposes. It is used as a cohesion feature to be included later in the classification algorithm. Meanwhile, it is also used in the subtopic segmentation component. In order to measure the semantic similarity between messages, we propose a Revised cosine Similarity Algorithm (RSA). RSA improves cosine similarity measurement in three ways. First, it incorporates POS tagging information into a Vector Space Model. Research has shown that noun phrases and verb phrases carry most of the important meaning in a sentence, while conjunction, adverbs and adjectives are less important. Thus, we define meaningful keywords to be noun, noun compound, named entity, verb and verb phrase. Instead of taking into consideration every single keyword, we only focus on these meaningful keywords. We represent all keywords in a message using the Vector Space Model (VSM), one of the most popular methods to identify lexical-level similarity (Salton and McGill 1986). Secondly, a traditional VSM uses
term frequency count to represent a document. We improve it by using term frequency and inverted
document frequency: $\text{tfidf}$. $\text{tfidf}$ is a measure derived from Information Retrieval research. It measures the
importance of a term to a message in an entire collection. This will downgrade some of the common words
that are less important. Finally, in group discussion text, users tend to use different words to express the
same meaning (Nash 2005). A traditional VSM will treat these synonyms as two different entries and this
will result in a zero similarity value. We further modify VSM by calculating a term-level similarity value
$\text{coef}_k$ and incorporate the value into our $\text{tfidf}$ matrix. When all the terms in two messages are different,
but there are synonyms or terms that are semantically related, these two messages will still have some
level of similarity. The similarity score between a pair of messages $X$, $Y$ is computed using RSA similarity,
shown in Equation 1.

$$
R \text{Sim} (X, Y) = \frac{\sum_{k=1}^{n} S_{x_k} \cdot S_{y_k}}{\sqrt{\sum_{k=1}^{n} S_{x_k}^2 \cdot \sum_{k=1}^{n} S_{y_k}^2}}
$$

Where

$$
S_{w_{y_k}} = tf_{y_k} \cdot idf_k \cdot \text{coef}_k
$$

### Sentiment Detection and Computation

During group discussion, it is very popular for a user to express
his opinion about the previous message with simple sentences such as “Yes, I agree with you” and “No,
your point is wrong.” Such a message often does not include enough useful words for deep semantic
analysis. However, identifying the sentiment of these sentences will help to find the reply-to relationship.
Therefore, in the sentiment detection and computation step, each term in a message is compared to a set
of sentimental keywords from a sentiment dictionary based on HOWNET. Using a simple sentiment
detection method, messages with obvious subjective sentiment can be identified and stored in the
system’s msgSentiment feature.

### Discussion Logic Feature Extraction and Subtopic Segmentation

It is obvious that disrupted turn adjacency is caused by the high degree of subtopics intertwined in
discussions. Identifying subtopics in a discussion not only helps to structure the discussion text, but also
serves as input for discussion logic features. Subtopic segmentation assumes that a subtopic is always
proposed in advance of discussions of that subtopic. Our rationale is that a subtopic is a message that has
low relevance to all previous messages in a previous subtopic and has high relevance to all following
messages till the end of that subtopic. Relevance of two messages is measured by the MsgSimList feature
described in the previous section. Figure 2 illustrates a subtopic segmentation scenario and definition of
variables. The steps for our subtopic segmentation algorithm are provided below. Our rationale is to use
some statistical measures such as mean and variance between messages to predict a true subtopic:

1. Initialize the subtopic list
   If $\text{preAllMean} = 0$, Then set current message = new subtopic
2. Identify candidate subtopic
   If (distance from the previous subtopic to this current message <2) ($\text{preAllMean} < \text{msgMean}$),
   Then current message = candidate subtopic
If (distance from the previous subtopic to this current message >= 2) and (preMean < msgMean),
Then current message = candidate subtopic

3. Find marginal message based on candidate subtopic
   We define marginalMeanSet as collection of marginalMean.
   Select the maximum similarity score from marginalMeanSet. If its marginalMean is greater than
   residualMean, then mark the corresponding message as marginal message.

4. Find subtopic
   If (preVariance > msgVariance) and (marginalMean > msgMean)
   Then candidate subtopic = new subtopic;
   ElseIf marginalMean > msgMean and the maximum in candidate subtopic's MsgSimList is not
   generated with previous message, then candidate subtopic = new subtopic

```
<table>
<thead>
<tr>
<th>Feature</th>
<th>Categorical Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistRange</td>
<td>0-5</td>
<td>The distance of the message pair</td>
</tr>
<tr>
<td>SimDegree</td>
<td>0-5</td>
<td>The similarity degree of the message pair</td>
</tr>
<tr>
<td>SentiOrNot</td>
<td>0-3</td>
<td>Whether or not two messages include sentiment polarity</td>
</tr>
<tr>
<td>SubtopicOrNot</td>
<td>0-3</td>
<td>Whether or not two messages are subtopic messages</td>
</tr>
<tr>
<td>LogicPosition</td>
<td>0-7</td>
<td>The logic position of the two messages in the discussion</td>
</tr>
</tbody>
</table>
```

**Classifier Construction**

Once all message features are extracted, we adopt a corpus-based machine learning approach to identify reply-to relationships. We model reply-to relationship identification as a binary classification problem. For each candidate adjacency message pair, they either construct a true reply-to relationship or not. The classifier construction includes three stages. We first transform message features into candidate adjacency pair features. We then present the TBL classifier. The residual matching mechanism is developed for remaining unpaired messages. It improves performance by matching messages based on similarity and discussion logic features.
The distance of each message pair is derived from a system feature, the message's ID number. These continuous numbers are dispersed from 0 to 5 degrees, denoted as DistRange. SimDegree and SentiOrNot are derived from linguistic features. The SimDegree feature is represented by six degrees from 0 to 5, with 5 being the most similar. In message-level features, msgSentiment captures the sentiment polarity of a message. To derive the SentiOrNot feature, we assign the value of a pair according to the 4 possible combinations: both contain subjective sentiment, both do not, or one contains subjective sentiment (two possibilities with either i or j).

SubtopicOrNot and LogicPosition are derived from discussion logic features. SubtopicOrNot determines whether the two messages are subtopic messages or not. With the LogicPosition feature, we assign the value based on whether the two messages exist in the same subtopic region and the number of subtopics between them. The discussion logic feature not only obtains the subtopic message but also subtopics' corresponding marginal messages. The subtopic region refers to the range from subtopic message to marginal message.

**Adjacency Pair Classifier with TBL**

The above features are fed into the transformation-based learning (TBL) classifier. TBL has been successfully applied in spoken dialogue act classification (Samuel 2000; Brill 1995). It starts with the unlabeled corpus and learns the best sequence of suitable “transformation rules” that can be applied to the training corpus to get better performance for the task. In our approach, each candidate adjacency pair is represented as a feature vector. One advantage of TBL is that the generated rules are easy to understand.

Each rule derived from TBL is composed of two parts: combination of features as the condition and reply-to relationship tag as the classification result. For example, “1 || 0 || 0 || 2 || => 1” is a rule. The five feature values are separated by “||”. This rule means: if DistRange=1 and Position=0 and SentiOrNot=0 and SubtopicOrNot=2, the resulting relationship is a true reply-to relationship.

**Assigning Reply-to Relationships for Remaining Pairs**

Not all of the messages are taken care of in rules generated by TBL. For those remaining messages that are not subtopics, we use a recursive similarity matching method to identify reply-to relationships. Prior match method is intuitive method which assigns each remaining post to the prior one. This method tends to have lower precision in GSS. We proposed a new recursive similarity matching method that considers the message’s similarity and discussion logic. The recursive process for our similarity match is provided below:

Define X as the residual message sequence in chronological order. Define Xi as the i-th element of X. Define Y-Set as the set of entire identified messages with root Y.

1) Initialize R is topic of this discussion and its first-level messages is subtopics.
2) Treat the message R as root and then split the tree structure into subTree by R message. Compute similarity value between Xi and the first-level messages. And then mark the message with the highest value as Y.
3) If the Y is a leaf or the similarity value between Xi and Y is the largest value in Xi’s MsgSimList, then Y is Xi’s antecedent message.
4) Else, mark Y as R and goto 2).

**Evaluation**

**Testbed and Dataset**

Our proposed TCA system is a language-independent method. It has several language dependent components such as POS tagging and noun phrase identification. With the development in computational linguistics, these parsing tools are available not only in English but also in other popular languages. We evaluate TCA using a Chinese group discussion testbed. Chinese poses some special challenges in Natural Language Processing tasks. From a linguistic perspective, Chinese is a language without explicit word
boundaries. It uses fewer function words and applies a looser grammar structure compared with English. These characteristics of Chinese create more ambiguities during parsing of syntax structure than English (Levy et al. 2003). Therefore, the accuracy of Chinese text parsing is often lower and Chinese parsing tools are also less developed. However, as the second most popular language on the Internet and with almost 5 times the growth rate of English, it is believed that Chinese will become the most popular language on the Internet in the future. Thus, we evaluate the TCA system on a Chinese group discussion testbed described in a prototype GSS system (Li et al. 2009). To record the true reply-to relationships, the GSS system is defined six categories according to Toulmin’s argument process: Question, Answer, Agreement, Objection, Explanation and New Topic. A new topic does not respond to any of the antecedent messages. The other five categories indicate different sentiments or argument processes in a certain subtopic. During the discussion process, users are required to select a previous message as the antecedent message and specify the purpose of the message from the five categories. If he is posting a new topic, there will be no previous message selected. We use these user-generated reply-to relationship tags as the “gold standard.”

We recruited 80 subjects who are doctoral and master’s students of Management Information Systems at a university in Shanghai, China. They all had experience with GSS-supported electronic meetings and knew how to properly use the GSS software. They were divided into 20 groups to discuss the topic “how to address the overproduction problem for a tea bag manufacturer.” Subjects were told that they need to discuss the solutions to this overproduction problem and reach a business decision in thirty minutes. We obtained a total of 20 group discussion texts and each consists of input from four participants. The maximum number of messages in all 20 discussions is 71, the minimum is 22 and the average is 42. We observed that linguistic features such as direct address and co-reference are not popular in these messages.

**Experiment 1: Subtopic Segmentation**

**Experimental Design**

In the first experiment, we evaluated the effectiveness of our similarity-based subtopic segmentation method. Subtopic segmentation is an important component in our TCA system. It contributes to the tree structure and serves as the discussion logic feature. Our subtopic segmentation is based on our innovative RSA similarity measure, revised VSM and statistical calculation. We compare our approach with a traditional text clustering approach. The benchmark clustering approach uses the similarity scores derived by traditional VSM between these messages and the Euclidean Distance measure. Precision, recall and F-measure were used as our performance measures. They are defined as follows:

\[
\text{precision} = \frac{tp}{tp + fp}, \quad \text{recall} = \frac{tp}{tp + fn} \quad \text{and} \quad F - \text{measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

where \(tp\) is the number of correctly identified subtopics, \(fp\) is the number of identified subtopics that are not true subtopics and \(fn\) is the number of true subtopics that are missed.

**Hypothesis**

We propose the following hypothesis:

\(H1: \text{Using an innovative RSA similarity measure, our topic segmentation algorithm will improve the performance compared to ordinary clustering methods.}\)

**Experimental Results and Discussion**

Table 4 shows the experimental results. Our algorithm has better performance on the 20 discussions. Obviously, TCA subtopic segmentation achieves much higher precision, recall and F-measure than that of traditional text clustering. Our average precision is 0.653, average recall is 0.738 and F-measure is 0.677. Figure 3 shows the F-measure performance of the two methods on all 20 discussion texts. The subtopic segmentation method outperformed the clustering algorithm on every dataset. We believe that the subtopics selected by the text clustering method are highly similar to their classes in terms of lexical similarity. We believe two factors contributed to our superior performance. The traditional categorization method failed to differentiate between different types of terms that might hurt the similarity measure. Also, the benchmark categorization method performs an overall similarity measure and failed to capture
the discussion logic sequences. We also observed two discussion texts with 100% precision and recall using similarity-based subtopic segmentation. The experimental results demonstrated that considering the discussion logic process is necessary in subtopic segmentation. Our H1 is supported.

<table>
<thead>
<tr>
<th>Table 4. Experimental results for Experiment 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technique</td>
</tr>
<tr>
<td>----------------------------</td>
</tr>
<tr>
<td>TCA subtopic segmentation</td>
</tr>
<tr>
<td>Clustering algorithm</td>
</tr>
</tbody>
</table>

Figure 3. Experiment 1 F-measure performances for each discussion

**Experiment 2: Comparison of Techniques in Reply-to Relationship Identification**

**Experiments Setup**

In the second experiment, we compared our TCA classification method with several benchmark methods:

1) Linkage method based on system features obtained from message heading. Since most synchronized GSS do not provide quotation function, we only used the timestamp in each message header as system features. This method assumes that each message replies to the previous message. In a tree construction, this method will construct a tree with chronologically linked leaves.

2) Heuristic-based method that relies on cohesion features in the message body. The heuristic-based method uses three features: direct address match, lexical similarity match and residual match. Direct address match identifies reply-to relationships based on matching author screen names. The second part uses the “Xsimilarity” tool to compute lexical similarity and then obtains the similarity score between messages using traditional VSM. In the last part, we applied rules defined in (Fu et al. 2008) for messages without obvious above cues.

3) Classification-based method using syntactic features and system features. These benchmark techniques have been adopted in previous studies with web forums (Kim et al. 2010). However, no discussion logic features were used. The classification-based method represents the coherence analysis as a binary classification problem. Each message makes a pair with all previous messages. We extract four types of features from the message pairs. These features include “time_gap” and “dist” which are the intervals of time and distance between message pairs, respectively, “repeatNoun” which is the number of repeated nouns between message pairs and “viewer_timeGap” which examines the messages pairs from the same author and with time interval less than five seconds. Again, precision, recall and F-measure were used as our evaluation criteria.

**Hypotheses**

We believe that discussion logic features can play an important role when discussion text lacks system and cohesion features. Thus, our TCA method that incorporates all three types of features is likely to obtain better performance. We propose the following hypotheses:

H2a: TCA algorithm will outperform the linkage method for GSS-based coherence analysis.
H2b: TCA algorithm will outperform the heuristic-based method for GSS-based coherence analysis.

H2c: TCA algorithm will outperform the classification-based method for GSS-based coherence analysis.

**Experimental Results and Discussion**

Table 5 shows the experimental results for all four methods. The TCA method achieved the best performance in terms of precision, recall and F-measure. The F-measure was 20-50% higher than the other three methods. Such improved performance was consistent across all 20 discussions, as depicted in Figure 4. The heuristic-based method performed better than the other two methods. Surprisingly, the traditional classification-based method performed worst and F-measure is less than 0.1.

<table>
<thead>
<tr>
<th>Table 5. Experimental results for Experiment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technique</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>TCA method</td>
</tr>
<tr>
<td>Linkage method</td>
</tr>
<tr>
<td>Heuristic-based method</td>
</tr>
<tr>
<td>Classification-based method</td>
</tr>
</tbody>
</table>

The TCA method performed best among the four methods for our GSS-based testbed. And the superior performance was consistent across all 20 discussions, as depicted in Figure 4. This is due to the introduction of discussion logic features and the composition of advanced linguistic features. We observed that the enhanced accuracy of subtopic segmentation helped the performance of reply-to relationship identification, as illustrated in the performance comparison chart in Figure 3. Although both the TCA method and traditional classification-based method use the same machine learning algorithm, TBL, the results from TCA were significantly better. This is due to the additional discussion logic features used in the TCA method. In the traditional classification-based method, system features and simple linguistic features do not represent complete characteristics in identifying reply-to relationships. The linkage method only linked messages in chronological order. Our results showed that in synchronous group discussions, relying on system features alone does not help structure discussion texts. Meanwhile, because discussion texts lack interactional cues, only lexical similarity contributed to the heuristic-based method. The performance is better than all three other benchmark methods, but still significantly lower than TCA. The experimental results demonstrate the impact of decision logic features and complicated linguistic features on group discussion coherence analysis as well as effective application of the two features in the TCA method.

Synchronous group discussion can support problem solution by a set of decision-makers working together as a group. The process of decision discussion focuses on analyzing the problem itself instead of interaction among participants. And the object of utterance represents the issue about discussion topic not care about who is spokesman. Consequently, personal behavior with characteristic of communication fails to identify reply-to relationship. The experimental results support this point of view.

Despite achieving superior performance by incorporating many complex and advanced computational linguistic theories into TCA method, the performance of coherence analysis in synchronous group discussion is still lower than in asynchronous group discussion. In the study by Fu et al. (2008), they achieved an F-measure of 0.76 while we only achieved 0.509. Language difference could also contribute to the lower performance. But we believe that this is mainly due to the unique challenges in synchronous group discussion. The corpus of this research includes more than 70% interaction features, which means coherence analysis in synchronous group discussion is a more difficult problem.
Experiment 3: DATree Representation

After identifying the subtopics and reply-to relationships, we can represent the original unstructured group discussion text in a discussion analysis tree structure, which we call a DATree. DATree can enhance content analysis capability. Important subtopics are floated as top branches, sequence of discussion is corrected and the phenomenon of disrupted adjacency turns can be avoided. In Figure 5 we present an example of a DATree constructed by the TCA method on the right in comparison with the original unstructured text on the left. DATree provides more clear visualization in contrast to the unstructured view. From the general tree structure, it is easy to tell the number of discussion points (subtopics), how thoroughly each topic is discussed, what topics attracted intensive discussions and what topics are not well elaborated. Discussion details are also presented as tree leaves. Our example DATree was constructed with Chinese texts, but the techniques can be generalized to other languages. Although DATree was proposed for the analysis of synchronous discussion text, it can be applied in asynchronous text analysis as well.

Conclusions

In this study we applied coherence analysis to GSS discussion. Our research has several contributions. First, we propose the TBL Coherence Analysis (TCA) method to identify reply-to relationships in synchronous GSS discussions. GSS discussions generate the most unstructured texts among all CMC environments. TCA is unique in two ways. Conceptually, this method is based on Toulmin’s model (Toulmin 1958) where internal discussion logic is considered. Discussion logic features are divided into two categories: “hard” features such as a subtopic sequence and its related messages sets, and the “soft” feature which is the soul of the TCA method. In order to automatically detect the “hard” features from unstructured discussion text, a novel subtopic segmentation algorithm is introduced. The “soft” feature places great emphasis on discussion logic. Thus, similarity match assigned the reply-to relationship not only based on the highest similarity between pairs but also based on subtopic and child node relationships.

On the technical side, to measure the similarity between two messages, we revised traditional VSM by taking into consideration tfidf, POS information and word-level similarity. We incorporated system features, cohesion features and discussion logic features in our TCA classification method. The last step in the TCA method is a recursive process to take care of any remaining message pairs.

The results showed that the subtopic segmentation algorithm outperformed the traditional clustering algorithm in terms of precision, recall and F-measure. Our TCA classification method outperformed traditional linkage, heuristic-based and traditional classification-based methods in reply-to relationship identification. The results revealed that discussion logic features can greatly improve the performance of coherence analysis. Furthermore, the use of discussion logic features is promising for extremely unstructured user-generated discussion text where system features and cohesion features are less effective.
Finally, the DATree generated from the TCA method demonstrated the advantage of turning unstructured discussion text into a structured tree format. We believe that such a visualization tool will greatly help decision-makers obtain valuable information from massive user-generated discussion texts in an effective fashion.

In the future we will test our TCA method with different topic discussion text. We are also interested in analyzing the participant behavior in group discussion. We plan to apply the method to analyze other forms of user-generated data.

### Acknowledgements

This research was supported by the National Natural Science Foundation of China under Grant 71171131 and 70533030

### References


