ELSEVIER

Decision Support Systems 35 (2003) 149–166

Decision Support Systems

www.elsevier.com/locate/dsw

Automatic discovery of similarity relationships through Web mining

Dmitri Roussinov [a,*], J. Leon Zhao [b]

[a] *School of Accountancy and Information Management, SAIM, College of Business, Arizona State University, Box 873606, Tempe, AZ 85287-3606, USA*

[b] *Department of MIS, School of Business and Public Administration, University of Arizona, Tucson, AZ 85721, USA*

Abstract

This work demonstrates how the World Wide Web can be mined in a fully automated manner for discovering the semantic similarity relationships among the concepts surfaced during an electronic brainstorming session, and thus improving the accuracy of automated clustering meeting messages. Our novel *Context Sensitive Similarity Discovery* (CSSD) method takes advantage of the meeting context when selecting a subset of Web pages for data mining, and then conducts regular concept co-occurrence analysis within that subset. Our results have implications on reducing information overload in applications of text technologies such as email filtering, document retrieval, text summarization, and knowledge management.

Keywords: Data mining; Context sensitive similarity discovery; Empirical study; Group decision support systems; Internet; Machine learning; Organizational concept space; Text clustering; Web mining

1. Introduction

Many researchers and practitioners believe that the World Wide Web is a gold mine filled with useful information. Indeed, such vast amount of textual and multimedia information was not available for researchers before the mid 90s. According to Lyman and Varian [18], the Web currently contains more than 2.5 billion of pages, consisting of at least 10 terabytes of textual information. What also makes it a novel and exciting opportunity is its massive hyperlink structure and the ability to trace people's browsing patterns. Although the term ''Web mining'' started to appear frequently in journals and magazines, not many successfully tested applications have been reported yet. This paper presents our empirically proven Web mining approach that is capable of discovering semantic relationships between specified concepts, and as a result, helps to organize messages produced during electronic meetings supported by Group Decision Support Systems.

Electronic meeting support has been proven to have great impact on productivity of group discussions [6,20]. However, it has been reported that Group Decision Support Systems (GDSS) meetings (or computer-mediated meetings in general) very often result in *information overload* [2,11,14]. Indeed, due to anonymity, the ability to contribute in parallel and automated record keeping, the number of text messages generated by a dozen of participants can exceed

* Corresponding author.

E-mail addresses: dmitri.roussinov@asu.edu (D. Roussinov), lzhao@bpa.arizona.edu (J.L. Zhao).

0167-9236/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0167-9236(02)00102-1

a thousand within an hour. Consequently, it becomes very cumbersome to sort the text messages before moving to the decision-making stage. As a result, better techniques for clustering meeting messages are very important to improve the effectiveness of electronic meetings.

Several studies have explored automated summarization of meeting messages, for example by representing them with a list of most representative topics [2], or using concept maps [21], or clustering messages into semantically homogeneous groups [23]. The common belief behind those approaches is that automated processing techniques can reduce the cognitive load of meeting participants even if manual post-processing is still required.

All those approaches inherently rely on the algorithms representing content of the text messages through a vector space model [25], in which each message is represented by a vector. The coordinates of a vector are determined by the words or phrases (called *terms*) contained in the message. To perform the required processing, those algorithms compute the similarities between the documents based on their vector representations. The more common terms documents share, the more similar their vector representations are and the more similar the documents are believed to be by the algorithm.

This vector space model has apparent limitations and the results are not always accurate. For example, as the experiment in Ref. [23] indicates, the message "`Effective transmission of video over networks`" was typically placed by human experts into the same cluster with the message "`bandwidth concerns—impact of remote collaboration`" presumably since both relate to networking issues. However, because those two comments (messages) do not have any common terms, the vector space model would never detect any similarity between them, and thus the algorithm would unlikely place them into the same cluster or within proximity in a generated semantic map.

In short, the common approach does not take similarities between different words and phrases into account. As another example, `meeting` and `brainstorming` would be treated as different words by the algorithm, although in the context of collaborative computing they would be nearly synonyms from the participants' perspective.

We believe that a potential solution to this problem is to discover and to take into account the semantic similarity relationships among the terms in the specific context such as an organization, a project or a meeting. We extend the framework of *Organizational Concept Space* (OCS) introduced recently by Zhao et al. [29]. OCS captures similarities between different concepts in the context of organizational workflow by representing similar concepts with nodes in a graph, called *similarity networks*. The Web mining algorithm for automatic discovery of similarity relationships complements the OCS approach by making it more powerful in large business applications.

Our research objectives were to explore the possibility of mining semantic similarity relationships from the World Wide Web, to validate the use of OCS as possible representation of those relationships, and to demonstrate the usefulness of applying the "mined" OCS to automated clustering of meeting messages. Specifically, our investigation consisted of the following steps.

(1) We conducted experiments in which subjects familiar with the meeting were asked to identify the relationships between the most frequently occurring concepts during the meeting. We established that the results generated by the subjects were consistent, and thus concluded that the similarity relationships can be captured by an organizational concept space (OCS) or some other framework. We also established that this consistency of similarity relationships is context specific.

(2) We asked the subjects to cluster manually the meeting messages and compared the output of automated clustering with/without OCS. Our results indicated that OCS (if manually created by subjects familiar with the meeting) indeed makes clustering closer to what subjects would expect. Again, we also found that this result was context specific.

(3) We designed, implemented and tested a Web mining algorithm that created OCS in a fully automated manner based on the contents of web pages semantically close to the contents of the meeting. We verified that the automatically created OCS (called *I-OCS*) resembles the manually created OCS (called *M-OCS*) and can help improve the accuracy of automated clustering as well. We also demonstrated that taking the meeting context into consideration improved the accuracy of message clustering.

Although our study focuses on resolving the information overload problem in computer-supported meetings, the approach developed in our study is useful to more general textual information processing tasks such as search/retrieval, filtering, categorization, and summarization.

The next section briefly reviews the pertinent literature. Section 3 presents the research questions and the research methodology. Section 4 gives the details of the algorithms and their implementation used in our experiments. Section 5 contains the results of our experiments and their interpretation and implication. Finally, Section 6 concludes the paper and outlines several directions for future research.

2. Literature review

2.1. Clustering and summarizing electronic meeting messages

Several previous studies have explored automated detection and summarization of structures in meeting messages by identifying and listing the most important concepts [2], representing the messages with semantic maps [21], or clustering the messages into automatically created subsets of topics [23]. These studies established the ability to organize the meeting contents and to help prepare the meeting participants for the decision making phase. However, the automated message clustering techniques do not always offer satisfactory accuracy. When compared with the results of manual clustering, researchers have found that the outputs of computer algorithms do not always match human expectations [21,23].

As we wrote in Introduction, these automatic message clustering approaches, along with other widely accepted text technologies, inherently rely on automated indexing, by which each message is represented with a vector, while the vector coordinates are determined by the words or phrases (called *terms*) in the message. To perform clustering or to build semantic maps, the algorithms rely on the computation of similarity between the messages. While the particular scoring formulas may be different (cosine, Jaccard, Euclidean, etc.), all of them just implement simple intuitive consideration that the more common words the messages share the more similar they are. Messages that do not share any common words or phrases would be considered the least similar.

The weakness of this approach is that it treats each word or phrase as a unique feature. Thus, the words `meeting` and `brainstorming` are not considered as semantically similar features by the algorithm, and a pair of messages, one containing the word `meeting`, and the other `brainstorming` would never be treated as similar in this framework unless they share some other words. Therefore, they may end up in different clusters or in distant areas of a semantic map, certainly not that would the participants expect.

This problem has also been noticed in a more general domain of text technologies [25] and traditionally known as vocabulary problem [10]. However, there has not been an effective solution to it. Since natural languages are very ambiguous and diverse, solving this problem would require knowing semantic relationships between all possible words and phrases. This task is believed to be "AI-complete," [17] which means solving it would require solving all the other Artificial Intelligence (AI) tasks such as natural language understanding, common sense reasoning and logical thinking.

Nevertheless, we believe that some progress in the right direction can be made. Our study has explored one step towards alleviating this problem. In particular, we have studied how the relationships of semantic similarity can be represented and can be mined from such a large knowledge repository as the World Wide Web.

2.2. Organizational concept space

A *concept space* approach [1,3,5] has been proposed to create meaningful and understandable domain-specific networks of terms and weighted associations, which are used to represent the underlying information spaces, i.e., documents in different text collections. The concept space approach consists of (1) acquiring complete and recent collections of documents as the sources of vocabularies, (2) automatically indexing all the terms in the documents, (3) clustering the documents based on the term frequency and the document frequency, and (4) organizing the documents based on the multi-term associations. The research on conceptual clustering and concept spaces indicates that techniques exist that are capable of

automatically creating conceptual networks of millions of domain-specific terms [3].

Zhao et al. [29] recently extended concept space techniques by incorporating organizational information directly in a data structure called *Organizational Concept Space*. An organizational concept space (OCS) stores the user interests in an organization via an *interest matrix* and a *similarity network*. A similarity network is a collection of similarity sets that form a network based on levels of generalization/specialization. Fig. 1 illustrates a simple similarity network.

A *similarity set* is a collection of concepts that are closely related semantically. For instance, "group systems" and "electronic meeting systems" are two concepts that are used interchangeably by many people in certain contexts. The concepts can be associated to the similarity set with a membership value between 0 and 1 denoting the degree of association (not shown in Fig. 1 for simplicity). An *interest matrix* is a two-dimensional matrix with "concept" as one dimension and "user" as the other. The combination of similarity sets and a user interest matrix is used to determine useful information objects for each user in a corporation, thus reducing information overload. Zhao et al. [29] described the algorithms to build the Organizational Concept Space (OCS) from employee feedback and a corporate information repository, but did not report any implementation and empirical results of the research.

The notion of OCS is relevant to our study because it provides an elegant way of capturing the organizational context of the messages we want to cluster automatically. OCS is different from a traditional manually built thesaurus in two important aspects: (1) OCS is specific to particular organization (or even to a particular context within organization such as meeting topic, project, etc.) and (2) OCS represents a degree of association between concepts by a number (e.g., ranging from 0 to 1) whereas a traditional, manually built thesaurus, is typically limited to binary (0 or 1) associations (e.g., synonymic, antonymic, generalization/specialization, "part-of" or other semantic relationships).

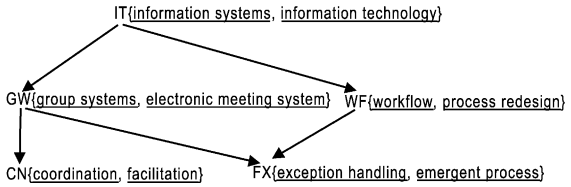


Fig. 1. A generic similarity network (adapted from [Ref. 29]).

Although the OCS framework also includes another data structure called *the user interest matrix*, we have not utilized it in this work and did not attempt to mine user interests. Without the interest matrix, an OCS framework is very similar to an automatically build thesaurus [1,8,12]: both represent relationships between concepts as weighted graphs. However, since OCS was specifically introduced for business workflow applications, we decided to follow OCS formalism in this study since we targeted GDSS applications.

2.3. Semantic mining and the Web

It has been known for a long time that the relationships between concepts (words or phrases) can be discovered by their co-occurrence in the same documents or in the vicinity of each other within documents. Firth [9], a leading figure in British linguistics during the 1950s, summarized the approach with the memorable line: "You shall know a word by the company it keeps." The classical work of van Rijsbergen [26] initiated the use of co-occurrence information for text retrieval and categorization. Until the 1990s, the studies exploring co-occurrence information in automated query expansion (adding similar words to the user query) resulted in mixed results [19,22], a good review can be found in Houston [15]. In the 1990s, the research scale increased dramatically, from hundreds to millions of documents, sometimes even using available supercomputer power [3]. The researchers also started to explore interactive systems, which only suggest concepts to the user for possibly extending his/her search instead of making that decision on the user's behalf. In the latter scenario, convincing positive results were reported [15].

We believe that the earlier difficulties resulted from several major factors: (1) there was only a small amount of data available for mining, and (2) the context in which the relationships were mined was not taken into account. In this study, we used the entire Web for mining a set of similarity relationships, thus providing enough "ore." Furthermore, we also mined the rela-

tionships in a specific context. Since the World Wide Web is very diverse, a surfer or a computerized agent can usually find a subset of pages related to a particular context and mine them for semantic relationships. We hoped to find those context specific relationships more reliable. By following Cardinal Richelieu's advice, we turn our "enemy" (the sheer volume of the Web) into our "friend" (ability to find the specified context).

According to Cooley et al. [7], the term Web mining has been used in two distinct ways. The first refers to Web content mining as the process of discovering useful resources from millions of sources across the World Wide Web. The second refers to the Web usage mining as the process of mining the Web access logs or other user information such as browsing and access patterns on one or more Web localities. In this study, we are interested in mining semantic associations between the specified concepts from the content of pages available in the entire Web, so our approach would approximately fall in the first category. For a comprehensive review of the Web mining literature, refer to Cooley et al. [7].

3. Research design

In this section, we first set up the research questions that we examine in this study. Then, we discuss the experiments designed to answer our research questions followed by the explanation of the metrics we used and our hypothesis in the operationalized form.

3.1. Research questions

We pose four research questions that will accomplish our research objective aforementioned:

Q1: What is the degree of the agreement among subjects with respect to the perceived relationships between concepts mentioned in the meeting?

This was a basic and necessary question. If there was no agreement among the meeting participants with respect to the relationships between concepts, then the entire idea of trying to represent them using any framework would likely to be futile. On the other hand, if the agreement was high, then we may try to capture them by an OCS or a similar framework. This leads to our second question:

Q2: Can Organizational Concept Space capture the relationships between concepts mentioned in a meeting in such a way that it would help to cluster those messages into more meaningful topics?

As presented in Ref. [29], OCS can be used to modify vector representation of text documents (including brainstorming messages) so that similar concepts are added even if they are not originally presented in the document. For example, the message "bandwidth concerns—impact of remote collaboration" originally represented by concepts BANDWIDTH, CONCERNS, IMPACT, REMOTE and COLLABORATION would also receive the concept NETWORKS if OCS captured semantic similarity between the concepts of NETWORKS and BANDWIDTH (as shown in Algorithms and implementations and Fig. 3). We tested if such OCS-based modification improves accuracy of clustering, i.e., grouping the messages in the way closer to what people would expect. Later sections in the paper provide more details on our experiment.

Q3: Can Organizational Concept Space be successfully mined from the Web?

Large-scale OCSs are difficult and time consuming to derive manually and therefore lacking automatic means may hinder the application of OCS and diminish its practical value. Consequently, an experiment of deriving OCS from the Web automatically is significant. Since a negative answer to this question would require an exhaustive evaluation of all possible mining algorithms, we deliberately look for a positive answer by testing if our heuristic mining algorithm discovers similarity relationships between the concepts mentioned during the meeting in such a way that the mined result (I-OCS) resembles the manually created OCS (M-OCS).

Q4: Does the Organizational Concept Space mined from the Web help cluster messages in a more meaningful way?

Q4 was similar to Q2 as both questions address the effectiveness of OCS. The difference between the two lies in the way the OCS was generated. Q4 was a special version of Q2 because it focuses on similarity relationships created automatically by a Web mining procedure.

In addition to the four questions given above, we also explored and tested a number of other supporting questions and hypothesis that we mention in Section 5.

3.2. Experiment design

In order to test our research hypotheses outlined below, we performed an experiment. We asked eleven (11) volunteers to (1) manually cluster messages recorded earlier by GDSS into semantically similar groups and (2) to specify the relationships between the most frequently mentioned concepts during the same meeting. The subjects were volunteering graduate students in the Information Management program.

We used the transcripts from the same brainstorming session as the one reported in Roussinov and Chen [23]. Thus, there was a time lapse between the actual meeting and our experiment. Since the topic of the meeting (''The Future of GroupWare'') was very general and our subjects demonstrated familiarity with the contents, we believe our subjects not being actual meeting participants does not considerably affect our findings. The original brainstorming transcript (containing 206 messages) was automatically recorded by electronic meeting support software called Groupsystems [20].

In order to address research question Q2 (effect on clustering), we compared the results of automated clustering meeting messages against clustering done by subjects. Each subject received the same text file with the meeting comments shuffled at random. We asked the same subjects to re-arrange the comments in the file using a text editor of their choice into groups (called *clusters* or *topics* throughout our paper) so that the messages in the same clusters are more similar to each other, and messages in different clusters are less similar, according to the subjects' judgment. This grouping technique resembled the process of organizing messages into topics that *Electronic Brainstorming Sessions* (EBS) participants usually do before moving to the decision making stage of the meeting [20]. The message grouping process took 40–50 min on average for each subject. To reduce the time consumption of manual message clustering, we used only a subset of 80 messages, the same as in Roussinov and Chen [23].

Upon completing the clustering task, the subjects were given a list L, consisting of the 20 most frequently mentioned concepts that were identified in the process of automated indexing of the messages as explained in Algorithms and implementations and represented in Table 1. For each concept c in the list L, the subjects had to choose three (3) other concepts from L, those that seemed to be the most semantically related to c according to the subjects' judgment. We used the data collected this way to test research question Q1 (agreement on similarity relationships) through a statistical analysis.

As one can see, some concepts are the words with the same roots, e.g., NETWORKS and NETWORK. As in prior related studies [2,21,23], we did not use any stemming algorithms while representing messages, thus treating words like NETWORKS and NETWORK as totally different terms (concepts). Although stemming sometimes helps in certain applications [25], it is not always reliable, nor desirable in a more general class of text technologies. For example, stemming results in collation of the words POLICY and POLICE—A clear case of failure. Xu and Croft [28] had suggested an algorithm that combined co-occurrence mining and linguistic stemming. We plan to study the issue of stemming in a future study. The discussion section presents some tests that we ran to isolate effects of same-root word similarity from none same-root.

Table 1
List L: 20 most frequently occurring concepts during the meeting

0	MEETINGS
1	MEETING
2	TECHNOLOGY
3	COLLABORATIVE SYSTEMS
4	INFORMATION
5	COLLABORATIVE
6	DISTRIBUTED
7	SYSTEMS
8	LINEAR THREAD MEETING
9	ENVIRONMENTS
10	NETWORKS
11	SUPPORT
12	NOTES
13	HARDWARE
14	FACILITATORS
15	TECHNOLOGIES
16	LANGUAGE
17	WIRELESS
18	NETWORK
19	BANDWIDTH

One can possibly identify several semantic "islands" of related concepts in the subset listed in Table 1, e.g., "networking" (NETWORK, NETWORKS, BANDWIDTH, WIRELESS, DISTRIBUTED), "technology" (TECHNOLOGY, TECHNOLOGIES, HARDWARE), "meetings" (MEETING, MEETINGS, FACILITATORS), "collaborative computing" (NOTES, COLLABORARIVE SYSTEMS), etc. We test through our experiment if the subjects agree on those relationships and if the relationships can help automate message clustering.

It was also obvious that the meanings of our selected concepts were very context specific. For example, NOTES actually referred to "Lotus Notes," and LINEAR THREAD MEETING is jargon standing for a distributed brainstorming meeting narrowing down to a single issue. FACILITATORS was also used to refer to a meaning specific to GDSS. One can see that even such simple examples would fail those mining approaches that do not take advantage of the context specific knowledge.

To test research question Q3 (the possibility to mine context specific similarity relationships), we ran our semantic Web mining algorithms to test if our implementation can come up with a set of relationships similar to those established by our subjects.

Finally, to test research question Q4 (the effect of the mined concept space on clustering accuracy), we constructed an OCS based on the mined relationships (called *I-OCS* or *Internet-based OCS*) and ran the same clustering comparison tests that we performed to test our second question.

3.3. Metrics

3.3.1. Q1 and Q3: overlap in concept selection

To test the consistency in the subjects' choice of the most similar concepts, we computed the average overlap. For example, given the target concept BANDWIDTH, if subject 1 selected NETWORKS, TECHNOLOGY, WIRELESS and subject 2 selected NETWORKS, TECHNOLOGY, SYSTEMS, the overlap for the concept BANDWIDTH would be 2. In order to have a metric independent of the number of subjects and the number of concepts, we normalized this overlap by the maximum number of overlaps (three from each subject–concept pair). If we had one target concept and two subjects in the above example, the overlap would be 2 and the maximum number of overlaps would be 3 (in the case of the same choices). Then our normalized overlap metric should be 2 out of 3 = 66.67%.

To test the agreement between the subjects' choice and the relationships mined from the Web, we used the same metric of a normalized overlap. We computed the overlap for the same top 20 concepts that were given to the subjects (*L*-list), and truncated the mined relationships to the top 3 most related, thus simulating subjects' actions.

3.3.2. Q2 and Q4: accuracy of clustering

We now discuss the metrics that were used to evaluate the accuracy of automated clustering. As we described in the literature review section, to measure the quality of clusters obtained automatically, we used their "closeness" to the clusters created by humans in terms of the number of wrong or missed associations. Same metrics were used and justified in Ref. [23] and resemble the so called *adjusted Rand index* [16] widely used in clustering applications in genetics. The definitions below explain the metric.

Grouping messages into non-overlapping clusters is called a *partition*. We call a partition created by a human subject the *manual partition*. The *automatic partition* is the one created by a computer. Inside any partition, an *association* is a pair of documents belonging to the same cluster. The *Incorrect associations* are those that exist in the automatic partition but do not exist in the manual partition. The *missed associations* are those that exist in the manual partition but do not exist in the automatic partition. We define a metric of *normalized clustering error* as:

$$\mathrm{NCE} = \frac{E}{A_t} \tag{1}$$

Here, E represents the total number of incorrect (E_i) or missed (E_m) associations: $E = E_i + E_m$. A_t is the total number of all associations in both partitions without removal of duplicates (associations existing in both partitions). It is computed as $A_t = A_m + A_a$, where A_m is the total number of associations in the manual partition and A_a is the total number of associations in the automatic partition. We considered only associations from clusters representing three or more documents. It is easy to verify that this measure belongs to a [0,1] interval.

Fig. 2 shows an example of manual partition (the top) and automatic partition (the bottom). In this example, the clustering algorithm made a mistake by placing document 5 with documents 1, 2, 3, 4 instead of 6, 7, 8. The incorrect associations (four total) are therefore 5-1, 5-2, 5-3, 5-4. The missing associations (three total) are 5-6, 5-7, 5-8. So, $E = 4 + 3 = 7$.

The total number of associations in the manual partition $A_m = 6 + 6 = 12$. The total number of associations in the automatic partition $A_a = 10 + 3 = 13$. So, $A_t = 12 + 13 = 25$. The normalized clustering error $\mathrm{NCE} = \frac{E}{A_t} = 7/25 = 0.28$.

By performing random permutations in the partitions generated during our experiment, we also empirically verified that this metric has the desired "smoothness." The automatic partition identical to the manual partition resulted in the metric value of 0. If some clusters were split or merged, the increase in the metrics was approximately proportional to the number of changes ("splits" or "merges"). As the number of permutation grew, the metrics approached the value of 0.9, the case of an entirely random partition. It never achieved 1, because even entirely random partition still correctly "guessed" some associations. After the simulations, we accepted 0.9 as the upper limit of the error.

3.4. Hypothesis

Using the metrics discussed above, we operationalized our research questions Q1–Q4 by the following null hypothesis:

H1$_0$. The average normalized overlap between the subjects' selections was not different from the random normalized overlap. The alternative would be that the overlap was higher than the one created by the random selection and would imply that there was agreement between the subjects with respect to the perceived similarity relationships.

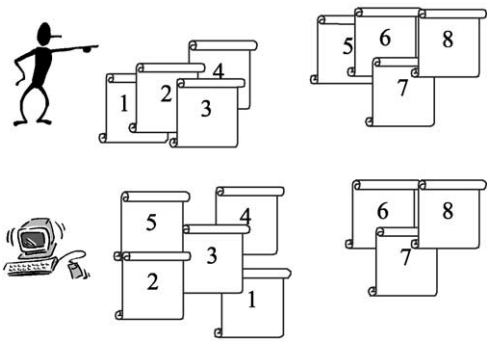


Fig. 2. An example of normalized clustering error (adapted from Ref. [23]).

H2$_0$. Clustering with the use of the manually created OCS (M-OCS) had the same normalized cluster error as clustering without using M-OCS. The alternative was that the error was smaller when M-OCS was used, and would indicate that M-OCS positively affected the accuracy of clustering.

H3$_0$. The average normalized overlap between the subjects' selections and the mining algorithm selections was not different from the average normalized overlap between the subjects' and random selection. The alternative was that there was a non-random agreement between the mined similarities from the Web and those indicated by the subjects. It would indicate that the mined OCS resembles the manually built OCS.

H4$_0$. Clustering with the use of OCS created by Web mining (I-OCS) has the same normalized cluster error as clustering without using I-OCS. The alternative hypothesis was that the error was smaller when I-OCS was used, and would indicate that I-OCS positively affected the accuracy of clustering.

4. Algorithms and implementations

4.1. Message clustering

For automated clustering, we used the same algorithm as reported in Roussinov and Chen [23]. It consisted of the following steps:

(1) automatic indexing
(2) vector space representation
(3) applying clustering technique.

We briefly explain each step below.

- *Automatic Indexing*: The general purpose of automatic indexing is to identify the contents of each textual document automatically in terms of associated features, i.e., words or phrases. Automatic indexing first extracts all words and possible phrases in the document. Then it removes words from a "stop-

word'' list to eliminate non-semantic bearing words such as ''the'', ''a'', ''on'', and ''in''. Our automatic indexing program also created phrases from adjacent words. In this study, we used the automatic indexing technique described in Orwig et al. [21]. We did not attempt any modifications of the indexing program for our study and deliberately did not change any parameters or the stop-word list once we had started our experiments.

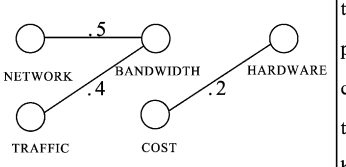
• *Vector Space Representation*: Each coordinate in the vector space corresponds to a term. If a term is present in the document, the coordinate is set to 1, otherwise to 0. We also normalized the vectors to unit length. Prior research [2,21,23] suggested that this scheme is adequate for representing electronic meeting messages. Although normalization is generally optional, it was necessary in our case because the size of a message varied several fold, and would affect the use the similarity metric that we used—negative Euclidean distance. The smaller the distance, the more similar the documents are believed to be.

For computational efficiency and accuracy of representation, we preserved only the top 100 terms (words or phrases) that were most frequently included in the collection. According to Chen et al. [2], Orwig et al. [21], Roussinov and Chen [23], this approach works best with small collections consisting of short text messages, since it provides the greatest overlap in representations. Table 1 shows the list of the 20 most frequently appearing terms in the collection. The average number of terms preserved in the document representations was 3.7.

• *Clustering*: We used Ward's [27] hierarchical agglomerating clustering technique in this particular study, similar to Roussinov and Chen [23]. The algorithm starts with each document in a cluster of its own and iterates by merging the two most similar clusters until all the documents are merged into a single cluster. By keeping track of all the mergers, the algorithm produces a balanced binary tree called a *dendrogram*. Since for our study we needed non-overlapping partitions of messages into topics (clusters), we converted the dendrogram into a partition simply by traversing the tree in pre-order stopping when the nodes were representing the clusters smaller or equal in size than the specified threshold. Since the subjects typically did not create clusters larger than 10, we also set our threshold to 10. This way, the distribution of cluster size among the automatically generated clusters was the most similar to those produced manually. Although our accuracy metrics were not very sensitive to the cluster size, we still wanted to approximate manual output as closely as possible.

4.2. Application of organizational concept space

In this study, we used a simplified version of OCS, namely a similarity network represented with a symmetric matrix. Each row or a column represented a term (word or phrase). Each cell contained a numerical value representing the similarity between the corresponding two terms, ranging from 0 (no similarity) to 1 (synonyms). Fig. 3 shows an example of a



	network	traffic	problem	current	technology	bandwidth	cost	Hardware
network	1	0	0	0	0	.5	0	0
traffic	0	1	0	0	0	.4	0	0
problem	0	0	1	0	0	0	0	0
current	0	0	0	1	0	0	0	0
technology	0	0	0	0	1	0	0	0
bandwidth	.5	.4	0	0	0	1	0	0
cost	0	0	0	0	0	0	1	.2
hardware	0	0	0	0	0	0	.2	1

Fig. 3. An example of similarity network and its matrix representation.

similarity network as used in our study. Each node in the figure corresponds to a concept, e.g., BANDWIDTH. If a concept is similar to another concept, they are connected by a weighted arc. There are three similarity arcs in Fig. 3: (NETWORK, BANDWIDTH, 0.5), (TRAFFIC, BANDWIDTH, 0.4), and (COST, HARDWARE, 0.2). For large-scale implementations in the future, we may need to take advantage of the sparseness of the similarity matrix in order to keep the processing time under control, similarly as it was done in Roussinov and Chen [23].

To obtain an OCS, we averaged the choices made by the subjects. For example, if 5 out of 11 subjects picked BANDWIDTH as one of the three most closely related concepts to NETWORK, the averaged similarity between them would be recorded as $5/11 = 0.46$.

This simple implementation suffices our need in this study because: (1) we want to test if a valid OCS could be obtained and was useful without trying to optimize its overall effect, and (2) a simple implementation is much easier to replicate in subsequent studies. We plan to explore more fine-grained implementations in the future.

Our objective of applying OCS was to utilize the similarity relationships between terms, so that the vector space representation of the messages can be enhanced by the similar terms. For example, if a message has a term BRAINSTORMING and the OCS tells us that BRAINSTORMING has a non-zero similarity to the term MEETING, then the coordinate corresponding to the term MEETING in the message vector would appear even if it was not originally there.

We used the most simple and straightforward implementation of the above enhancement. We modified each message vector $\boldsymbol{V}$ by the following linear transformation written in a matrix form:

$$\boldsymbol{V} = \boldsymbol{V} + a\ \mathbf{S} \times \boldsymbol{V},$$

where $\mathbf{S}$ is the matrix representing the similarity network, and a is the adjustment factor. The product $\mathbf{S} \times \boldsymbol{V}$ is the vector $\boldsymbol{V}$ multiplied by the matrix $\mathbf{S}$ defined by a common linear algebra. Depending on the range of values in $\mathbf{S}$ and the adjustment factor a, the changes in the representations of the messages may be very small or dramatic. In our experiment, the values S_{ij} ranged from 0 to 1. To perform our hypothesis testing, prior to the experiment, we chose $a = 1$, just for the sake of simplicity. We also analyzed the impact of different values of a afterwards.

The following example demonstrates the algorithm. Let us assume we only use the concepts NETWORK, TRAFFIC, PROBLEM, CURRENT, TECHNOLOGY, BANDWIDTH to represent meeting messages, so the vector space dimensionality is 6. The message "`Network traffic is going to be a problem with current technology`," originally would be represented as the vector (1, 1, 1, 1, 1, 0). "0" signifies that BANDWIDTH is not in the message. After applying the OCS shown in the Fig. 3, with $a = 1$, it would become (1, 1, 1, 1, 1, 0.9). Since BANDWIDTH is associated with NETWORK with the weight 0.5 and BANDWIDTH is associated with TRAFFIC with the weight 0.4, the total increment along BANDWIDTH coordinate was $0.4 + 0.5 = 0.9$ After this modification, all vectors were normalized to the unit length.

It is worth noting that even when the modifications are small, the resulting accuracy increase may be still significant. This is because without the modifications, most messages did not share any common terms and had the similarity of $-\sqrt{2}$ since we used negative Euclidean distance. Jaccard or cosine scores would result in 0 similarity in that case. This poses a problem for clustering algorithms that have to resolve many "ties" in order to form clusters. Even small changes in the coordinates can break those "ties" in the right direction, thus considerably helping the clustering algorithm to make better choices.

4.3. Mining OCS from the Web

There are many mining algorithms for discovering semantic similarities among given concepts [4,25,26]. However, we are not aware of any empirical studies specifically targeting the World Wide Web. In this study, we implemented techniques based on the co-occurrence approach [4,25,26] used earlier in non-Web text collections. Those techniques explored the observation that semantically similar concepts (words or phrases) tend to co-occur in text documents or in close proximity inside documents.

Since mining the entire Web would be cost prohibitive, an important decision was to be made in how to select a subset of Web documents as a collection for mining. We implemented a heuristic algorithm that takes advantage of the commercial web search

engines. In our study, we used AltaVista http://www.altavista.com), but most search engine would also qualify for the task.

Ideally, we would like to obtain web pages that refer to the tested used in our study and are semantically close to the topic of the meeting. For this purpose, we created a Web search spider that interfaces with AltaVista to retrieve Web pages using queries created from the top 30 concepts surfaced during the meeting. Our spider ran 30 unique queries and gathered 3272 unique Web pages. As shown in Table 2, each query consisted of 30 concepts (phrases were connected by '-', as required by AltaVista syntax), and exactly one word (phrase), different in each query, was preceded by '+' sign to request this word to be in the returned pages. For example, according to the AltaVista syntax, the query "meetings meeting +technology collaborative-systems information collaborative distributed systems linear-thread-meeting environments networks support notes hardware facilitators technologies language wireless network," required the returned pages to contain the word "technology." Other words were optional in the returned pages, but the Web pages containing them had a higher chance of being selected by the underlying search engine. This way, the queries were requesting the pages semantically similar to the meting context, which we subsequently manually verified.

Although not in our current implementation, the above query construction algorithm can be implemented in a fully automated manner. We limited the number of phrases in the query to 30 because the underlying search engine limited the size of the query string. More sophisticated implementations can be done in future studies. Our spider collected 200 pages from each query, thus the total number would be $200 \times 30 = 6000$, but it dropped to 3270 after removing the duplicates and non-existing pages. The HTML pages were converted into plain text using the "lynx" shareware program http://lynx.browser.org/) for the UNIX operating system and truncated to the first 20,000 bytes of text in order to avoid overly long web pages. Once the pages were collected, they were processed by the same automatic indexing procedure as mentioned in the preceding section and represented by vectors normalized to the unit length.

The entire downloading and conversion process was performed automatically in a sequential manner. Since conversion is a one-pass algorithm, it can be implemented relatively quickly so downloading would be the only bottleneck for performance. However, if performed in parallel, it can be sped up considerably, for example as reported in Ref. [24], so that the entire process can be implemented to run in real time.

As mentioned in Section 2, conceptual similarity mining exploits the fact that similar concepts (words or phrases) tend to co-occur in the same documents. To obtain the numerical value of similarity, we used the previously suggested [25,26] formula:

$$S_{ij} = \boldsymbol{t}_i \cdot \boldsymbol{t}_j / |\boldsymbol{t}_i| \, |\boldsymbol{t}_j|,$$

where S_{ij} is the similarity between terms i and j; $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ are vectors representing occurrences of the terms i and j in the documents in the mined collection. This

Table 2
A fragment of the list of the queries used to collect Web pages for mining

+meetings meeting technology collaborative-systems information collaborative distributed sys
meetings +meeting technology collaborative-systems information collaborative distributed sys
meetings meeting +technology collaborative-systems information collaborative distributed sys
meetings meeting technology +collaborative-systems information collaborative distributed sys
meetings meeting technology collaborative-systems +information collaborative distributed sys
meetings meeting technology collaborative-systems information +collaborative distributed sys
meetings meeting technology collaborative-systems information collaborative +distributed sys
meetings meeting technology collaborative-systems information collaborative distributed +sys
meetings meeting technology collaborative-systems information collaborative distributed syst
meetings meeting technology collaborative-systems information collaborative distributed syst
meetings meeting technology collaborative-systems information collaborative distributed syst
meetings meeting technology collaborative-systems information collaborative distributed syst
meetings meeting technology collaborative-systems information collaborative distributed syst
meetings meeting technology collaborative-systems information collaborative distributed syst

formula represents the cosine of the angle between the term vectors. This formula was has been demonstrated to work well in automated applications. For semi-automated concept space based expansion, other formulas were have been also successfully used [4]. We discarded as spurious all the co-occurrences that happened only within one web page.

To make our comparisons more objective and the mined concept spaces comparable with the concept spaces selected by the subjects, we truncated the mined relationships to only three most strongly related concepts (as in the subjects' case). A fragment of the mined Concept Space is shown in Table 3, where each concept is followed by three most closely concepts along with the similarity between them. The resulting OCS was applied in the same manner to modify vector representation of the messages as the manually created Concept Space described in the previous section. Prior to running experiments, we decided to use the same adjustment factor $a = 1$.

5. Experimental results

5.1. Analysis of results

This section presents our results using the metrics introduced earlier in Section 3 and discusses their implications.

5.1.1. H1: Consistency in subjects' choice

The average overlap in the subjects' selection was found to be 31%. Selecting related concepts at random would result in 15% overlap. We obtained this number by randomly shuffling the selected concepts in the lists obtained from the subjects. Following the Monte-Carlo technique [13], we computed the statistic significance of this deviation by testing 100 different shuffling of the selected concepts. The largest average overlap out of 100 shuffling tests was 18%. Since it was still much less than 31%, the test ensured that the difference was statistically significant at least at the level $\alpha = 1/100 = 0.01$. Thus, we have to reject $H1_0$ and conclude that there was significant degree of agreement among the subjects with respect to selecting the most relevant concepts.

5.1.2. H2: Improvement in the accuracy of clustering

Table 4 shows the normalized cluster error subject by subject, with and without M-OCS. We ran paired *t*-test to check if the mean difference still could be 0. We had to reject $H2_0$ ($p = 0.0005$) and conclude that the manually built OCS (M-OCS) improved the accuracy of clustering. Very low *p*-value was observed because the standard deviation of the difference was low, less than the standard deviation of the metrics themselves (0.07 vs. 0.10 and 0.15). This indicated that the subjects agreed on the difference more than they agreed on the accuracy of clustering itself.

Table 3
A fragment of a listing with mined similarity relationships

term:COLLABORATION	term:REMOTE
related:	related:
COLLABORATIVE 3.877262e-001	EXAMPLE 2.259506e-001
COLLABORATIVE SYSTEMS 3.381566e-001	NETWORK 2.501757e-001
SYSTEMS 2.656920e-001	SYSTEMS 2.463332e-001
term:FACILITATION	term:VOICE
related:	related:
COLLABORATION 1.341769e-001	TECHNOLOGIES 2.254068e-001
FACILITATORS 1.665471e-001	TECHNOLOGY 2.486785e-001
term:HARDWARE	WIRELESS 2.510902e-001
related:	term:WIRELESS
NETWORK 2.911064e-001	related:
SUPPORT 2.496007e-001	NETWORK 3.195135e-001
SYSTEMS 3.189309e-001	NETWORKS 3.093418e-001
term:MEETING	TECHNOLOGY 3.090011e-001
related:	term:BANDWIDTH
FACILITATORS 2.335312e-001	related:
INFORMATION 2.949944e-001	NETWORK 2.578386e-001
MEETINGS 6.949747e-001	NETWORKS 2.637258e-001
	SYSTEMS 2.067017e-001

Table 4
Normalized cluster error with and without M-OCS

Subject	No OCS	With M-OCS	Difference
1	0.63	0.50	0.13
2	0.89	0.82	0.06
3	0.84	0.77	0.07
4	0.61	0.48	0.13
5	0.60	0.47	0.13
6	0.79	0.76	0.03
7	0.84	0.89	−0.05
8	0.82	0.78	0.04
9	0.81	0.82	−0.02
10	0.74	0.73	0.02
11	0.78	0.62	0.17
Average	0.76	0.69	0.07
Standard deviation	0.10	0.15	0.07
Standard mean error	0.03	0.05	0.02
t-test, *p*-value			0.004346

Fig. 4 illustrates the error reduction effect of using OCS. In order to interpret the effect as large or small we needed to estimate the lower and the upper limits of the error. As we wrote in Section 3.3, the upper limit corresponding to entirely random clustering, was 0.9. Obviously, the low limit could not be 0, since there was no perfect agreement among the subjects themselves. We accepted the average disagreement among the subjects as the lower limit of the clustering error. In order to compute this disagreement, we took every pair of subjects and treated one subject's partition as the manual partition and the other subject's partition as the automatic partition to apply Formula (1). Then, we averaged this metric across all the subject pairs and finally obtained the lower limit to be 0.66. This number being large indicates low agreement across the subjects with respect to the clustering decisions, which does not come as a surprise considering the subjectivity of the task. However, the agreement among the subjects with respect to the improvement due to using OCS was still quite high, and as a results, the improvement itself was statistically significant.

Another interpretation of the average across subject disagreement is that it is equal to the average evaluation of each subject's clustering accuracy by the other subjects. We assumed, that the algorithm could not do better than a human subject. That is why we accepted the average across subject disagreement as the low bound of the error.

Taking the upper and low bounds into consideration, we can interpret that clustering without OCS resulted in the performance somewhere in the middle (0.76) between the worst (0.9) and the best possible (0.66), slightly closer to the best side. The *M-OCS*-based clustering (0.69) was closer to the best possible side and the reduction was quite noticeable (70% of maximum possible reduction!) even for such a simplified implementation as in our experiment. This result indicated potential of OCS framework for dramatically improving the accuracy of clustering.

5.1.3. H3: Mined relationships make sense

The average overlap between the mined relationships and those indicated by the subjects was 28%. Similar to H1, we ran Monte-Carlo tests and established that it was different from the random overlap at the level of significance <0.01. We have to reject $H3_0$ and conclude that there was a resemblance between mined relationships and those identified by the subjects. The actual overlap of 28% was remarkably high, considering that the average overlap in the subjects' selections was 31%. Since the algorithm would unlikely surpass a human, we accepted the average subject's overlap (31%) as the upper limit for the algorithm accuracy.

5.1.4. H4: Mined relationships improve the accuracy of clustering

Table 5 shows the normalized cluster error subject by subject, with and without the OCS mined from the

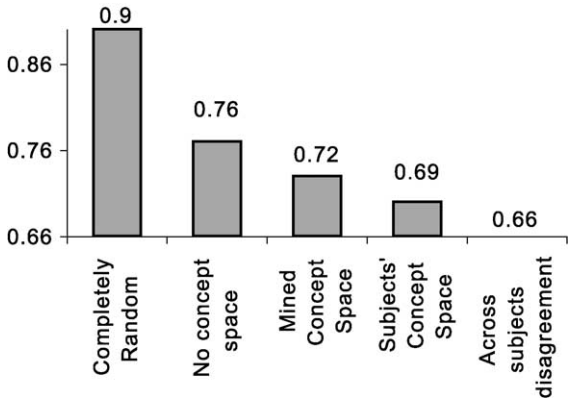


Fig. 4. Normalized cluster error for different concept spaces.

Table 5
Normalized cluster error with and without I-OCS

Subject	No OCS	With I-OCS	Difference
1	0.63	0.62	0.02
2	0.89	0.84	0.05
3	0.84	0.84	0.00
4	0.61	0.56	0.05
5	0.60	0.54	0.06
6	0.79	0.78	0.02
7	0.84	0.82	0.03
8	0.82	0.74	0.08
9	0.81	0.77	0.04
10	0.74	0.73	0.02
11	0.78	0.68	0.10
Average	0.76	0.72	0.04
Standard deviation	0.10	0.11	0.03
Standard mean error	0.03	0.03	0.01
t-test, *p*-value			0.0004489

Web (I-OCS). We ran paired *t*-test to check if the mean difference was equal to 0. Similar to H1, we detected a statistically significant ($p = 0.0005$) reduction in the cluster error due to using mined OCS: from 0.76 to 0.72. Again, as in H2, there was a good agreement among the subjects about the difference, resulting in a low standard deviation of 0.03. We have to reject $H4_0$ and conclude that the mined Concept Space (I-OCS) improved the accuracy of clustering. As Fig. 4 illustrates, the performance of using I-OCS was in the middle between the one without OCS and the one manually built from subjects' data (M-OCS), closer to the one with M-OCS. Thus, I-OCS achieved 40% of maximum possible reduction of error! This result indicated the potential of the semantic mining to dramatically improve the accuracy of clustering even in such a simplified implementation.

5.2. Discussion

This section discusses the limitations of the study and some additional findings.

5.2.1. Across subjects agreement

While the across subjects disagreement (0.66) seems to be quite high due to the subjectivity of the decisions involved in manual clustering, we believe that the overall idea of applying clustering techniques to GDSS supported meetings is still sound due to enormous information overload that the participants endure. Although we did not test the effect of using clustering on the productivity of meetings, we believe that even pre-organizing messages into clusters would help to identify the important issues and to move into the decision making phase. The participants can always clean up the clusters (topics) altogether as a group during the meeting if the initial starting point provided by GDSS does not seem to be adequate for them.

5.2.2. Stemming

By looking at the mined or manually identified relationships, we concluded that many of them were between words with the same roots, e.g., NETWORK and NETWORKS. It is noteworthy that the mining algorithm was able to discover those relationships without using any linguistic resources. However, if almost all discovered relationships were between words with the same roots, our results would be much less interesting, since those relationships still could be obtained by finding the appropriate linguistic resources without any mining involved. In order to differentiate same-root from different-root similarity relationships, we randomly shuffled all the subjects' choices, except those with the same root. This way, we re-tested H1 for only the relationships among words with different roots. Remarkably, the average overlap dropped to 22%, thus testifying that a significant proportion of the established relationships was between words with different roots, which defends the validity of our findings.

5.2.3. Change in statistical distributions

One other possible fallacy to our findings was that the improvement in clustering accuracy could be due to the changes in the statistical properties of the vector representations, but not really due to the semantic similarity taken into account. Indeed, after the OCS was applied, the vectors became less sparse. What if it was only the reduction in sparsity that improved the accuracy of clustering? In order to investigate this, we randomly shuffled discovered similarities. For example, if we had only two discovered similarity pairs (BRAINSTORMING, MEETING, 0.5) and (NETWORK, BANDWIDTH, 0.6), the shuffling would change them to (BRAINSTORMING, BANDWIDTH, 0.5), (NETWORK, MEETING, 0.6). Once the similarity matrix was shuffled, no improvement in

clustering accuracy was detected under any combinations of parameters in either the manually built or the mined concept spaces. This finding supported our belief that the correctness of similarity relationships was crucial for the discovered effects.

5.2.4. Importance of knowing the context

We also investigated the importance of knowing the context for both manual and mined concept spaces. To explore the *M-OCS* case, we asked three (3) other subjects to perform the concept selection tasks, but without preceding it with the clustering tasks. Thus, those three subjects were not familiar with the context of the meeting. We built and applied the OCS built from this non-context specific data the same way as we described above. In that case the accuracy of clustering became only worse! The normalized clustering error was in 0.78–0.85 range, under any combination of parameters. This finding confirmed our conjecture that knowing the context was crucial for the effectiveness when OCS was built manually. Having only three subjects for this simple test was a limitation, but we still think it is worth mentioning this additional finding.

To test the importance of the context to the mined OCS case, we created a different collection for mining in such a way that we did not provide contextual cues to the underlying search engine. Specifically, we simplified our heuristic querying algorithm by removing all the nonrequired keywords (not preceded with '+' sign in Table 2). Each of the 30 queries consisted of exactly one of the top 30 most frequent concepts. Those queries resulted in the pages about each of the concepts taken stand-along, i.e., the top 200 pages from AltaVista about "MEETING," then the top 200 about "LOTUS NOTES," etc., ignoring the context in which the concepts were mentioned. We also manually double-checked that those pages were more general, thus as we hoped, much less context specific.

We obtained 5241 pages after removing duplicates and empty pages. We randomly selected 3270 from them, the same number as for context specific mining in order to make comparison more objective. Then, we mined the OCS from the selected documents in the same way as described above. The resulting improvement was indeed smaller than with the I-OCS using context (0.74 normalized clustering error, instead of 0.72), with the difference between them being statistically significant (p-value = 0.045), which confirmed that knowing the context was important for similarity mining on the Web.

5.2.5. Adjustment factor

Fig. 5 shows the clustering improvement as a function of the adjustment parameter a. As one can see, the error reaches an upper limit about 0.77 when a approaches 0. The effect fluctuates significantly, due to randomness involved in automated clustering especially when many similarities have to be "tie-broken." Nevertheless, it was possible to identify the improvement from the original 0.76 for a wide range of values in a.

5.2.6. Other clustering algorithms

Using only one clustering algorithm in our study was of course a limitation. However, our result was more general because all text-clustering algorithms rely on the notion of similarity between documents (messages). Since our *context sensitive similarity discovery* (CSSD) framework resulted in a more accurate similarity computation, it should improve the results under other clustering algorithms as well. Instead of testing a wide range of clustering algorithms, we directly verified that CSSD resulted in improvement in the accuracy of similarity computation as follows.

We derived the matrix of similarities (called **S**-matrix) between each pair of documents based on the fraction of the subjects who placed this pair into the same cluster. The entry value in the matrix ranged from 0 (least similar) to 1 (most similar). For example, if two out of three subjects placed the message "Effec-

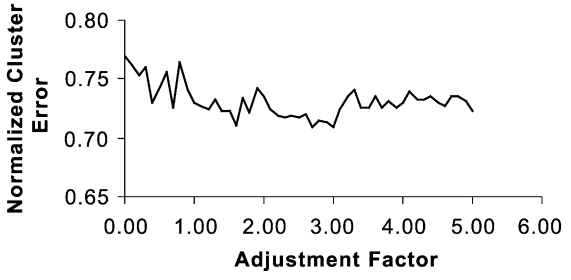


Fig. 5. Normalized clustering error as a function of the adjustment factor a.

tive transmission of video over networks" into the same cluster with the message "bandwidth concerns—impact of remote collaboration," the similarity between them recorded in **S**-matrix would be 2/3 = 0.67. If they never were placed into the same cluster, the similarity would be 0 (least similar). If all subjects placed them into the same cluster, the value would be 1 (most similar).

Then, we computed the correlation between the elements in the **S**-matrix and the automatically computed matrix of similarities (referred to as **C**-matrix). To compute the similarity in the **C**-matrix, we used negative Euclidean distance in the document vector space as Ward's (and many other) clustering algorithms are based on Euclidean distance. Since the **S**-matrix reflects the average of the subjects' evaluation of similarities between every pair of documents, the correlation between its elements and the corresponding elements in the **C**-matrix measures the accuracy of automated similarity computation. Indeed, if an element of **S**-matrix representing the similarity between a pair of documents (D1 and D2) was larger than the average, thus indicating a very similar pair, then the corresponding element in the computed similarity between (D1 and D2) also should be larger than the average, thus also indicating a very similar pair. So, an ideal algorithm should result in the correlation coefficient (Pearson's) close to 1, and conversely, a random assignment of similarities should result in a 0 correlation. In theory, the ideal correlation of 1 is only possible to achieve in the case of similarities statistically distributed in the same way, which is not usually the case. In our situation specifically, the **S**-matrix was manually constructed and the **C**-matrix was the result of the mining process. Therefore, even large positive values would still indicate good accuracy.

This metric is more fundamental than those based on clustering since it does not depend on a particular clustering algorithm, and thus can attest the accuracy of the automated similarity computation in many text-related tasks such as retrieval, clustering, filtering, categorization, etc.

Fig. 6 demonstrates the variation in the correlation coefficient (i.e., the similarity accuracy) as a function of parameter *a*. It is purely a coincidence that the maximum happens very close to $a = 1$. The improvement in the similarity accuracy was visibly significant.

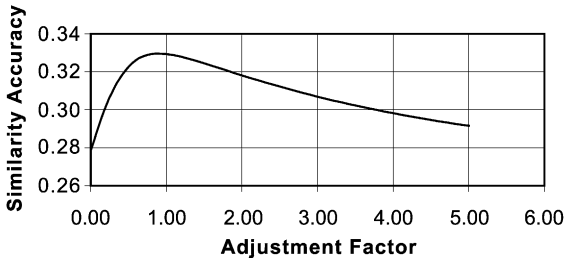


Fig. 6. Similarity accuracy measured by the correlation coefficient between computed message-to-message similarity and the one derived from subjects' data.

Since all clustering algorithms rely on similarity computation, we can generalize our findings that our CSSD method improves accuracy to all similarity-based clustering algorithms.

5.2.7. Size of messages

Testing our method with only meeting messages had certain limitation since those messages are typically short. Many text documents in other applications (e.g., research articles, web pages, CRM reports) are much longer and thus represented by more keywords. Although we believe our approach works best with smaller documents, it should be useful for longer ones, especially when there is a need to bridge vocabulary gaps: even two long documents may use different terminology (e.g., *car* vs. *automobile*) and the computed similarity between them would never be accurate without some kind of similarity discovery method.

6. Conclusions and future research

In this study, we proposed *context sensitive similarity discovery*, a new method, for computing similarity relationships among concepts surfaced during an electronic brainstorming session. Our experiment indicated that there are semantic similarity relationships between the concepts occurring in the context of an electronic meeting and demonstrated that those similarities can be captured by a specially designed data structure called Organizational Concept Space (OCS). Our test of hypotheses confirmed that those semantic similarities can be taken into consideration

while representing meeting comments and, as a result, the accuracy of clustering was greatly improved. Our experiments also demonstrated that context specific similarity relationships can be derived through Web mining in a fully automated manner, thus making the *context sensitive similarity discovering* (CSSD) method more powerful to use in a wide range of knowledge processing applications.

We discovered that taking the context of the messages into account was crucial for the consistency of the similarity relationships and for the success of the tested mining algorithms. We suggest that ignoring context in several prior studies may account for the reported failures.

Our research showed that using the discovered similarity relationships, the meeting messages can be grouped into topics (clusters) in a more meaningful way, thus reducing information overload. Since the concept of Organizational Concept Space is not limited to the GDSS domain, it may be used to capture similarities in many other applications where the notion of similarity among text documents is used. We plan to conduct studies that will test a wider range of parameters (such as the number of clusters, selection of clustering algorithms, the vector sizes, and the particular implementations of the CSSD method) in various business applications. Specific applications may include email filtering, document sharing, and knowledge distribution where the proliferation and ambiguity of vocabulary poses a potential problem to the accuracy and effectiveness of text clustering and matching.

Acknowledgements

It is our pleasure to acknowledge the help from Artificial Intelligence Laboratory and the Center for Management at the University of Arizona for providing data sets and software to support our study.

References

[1] H. Chen, K.J. Lynch, Automatic construction of networks of concepts characterizing document databases, IEEE Transactions on Systems, Man, and Cybernetics 22 (5) (1992) 885–902.

[2] H. Chen, P. Hsu, R. Orwig, L. Hoopes, J.F. Nunamaker, Automatic concept classification of text from electronic meetings, Communications of the ACM 37 (10) (1994) 56–73.

[3] H. Chen, B.R. Schatz, T.D. Ng, J.P. Martinez, A.J. Kirchhoff, C. Lin, A parallel computing approach to creating engineering concept spaces for semantic retrieval: the Illinois digital library initiative project, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8) (August 1996) 771–782.

[4] H. Chen, C. Schuffels, R. Orwig, Internet categorization and search: a self-organizing approach, Journal of Visual Communication and Image Representation 7 (1) (1996) 88–102.

[5] H. Chen, J. Martinez, A. Kirchhoff, T.D. Ng, B.R. Schatz, Alleviating search uncertainty through concept associations: automatic indexing, co-occurrence analysis, and parallel computing, Journal of the American Society for Information Science 49 (3) (1998) 206–216.

[6] T. Connolly, L.M. Jessup, J.S. Valacich, Effects of anonymity and evaluative tone on idea generation in computer mediated groups, Management Science 36 (6) (1990) 689–703.

[7] R. Cooley, B. Mobasher, J. Srivastava, Web mining: information and pattern discovery on the world wide web, in: R. Cooley, J. Srivastava (Eds.), Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), November 1997.

[8] C.J. Crouch, An approach to the automatic construction of global thesauri, Information Processing and Management 26 (5) (1990) 629–640.

[9] J.A. Firth, Synopsis of Linguistic Theory 1930–1955, Studies in Linguistic Analysis, Philological Society, Oxford, 1957, reprinted in Palmer, F. (ed. 1968) Selected Papers of J.R. Firth, Longman, Harlow.

[10] G.W. Furnas, T.K. Landauer, L.M. Gomez, S.T. Dumais, The vocabulary problem in human-system communication, Communications of the ACM 30 (11) (1987) 964–971.

[11] R.B. Gallupe, W.H. Cooper, Brainstorming electronically, Sloan Management Review 35 (1) (Fall 1993) 27–36.

[12] G. Grefenstette, Explorations in Automatic Thesaurus Discovery, Kluwer Academic Publishing, Boston, MA, 1994.

[13] J.M. Hammersly, D.C. Handscomb, Monte Carlo Methods, Methuen, London, 1964.

[14] S.R. Hiltz, M. Turoff, Structuring computer-mediated communication systems to avoid information overload, Communications of the ACM 28 (7) (1985) 680–689.

[15] A. Houston, Information classification: usability studies on two automatic approaches, Doctoral Dissertation, University of Arizona (1998).

[16] L. Hubert, P. Arabie, Comparing partitions, Journal of Classification, (1985) 193–218.

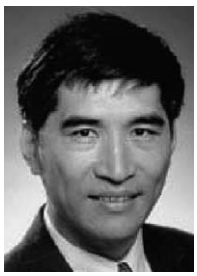
[17] N. Ide, J. Véronis, Word sense disambiguation: the state of the art, Computational Linguistics 241 (1998) 1–40.

[18] P. Lyman, H. Varian, How Much Information? A project report of the Regents of the University of California, available at http://www.sims.berkeley.edu/how-much-info (2000).

[19] J. Minker, G.A. Wilson, B.H. Zimmerman, An evaluation of query expansion by the addition of clustered terms for a document retrieval system, Information Storage and Retrieval, (1972) 329–348.

[20] J.F. Nunamaker Jr., A.R. Dennis, J.S. Valacich, D.R. Vogel, J.F. George, Electronic meeting systems to support group work: theory and practice at Arizona, Communications of the ACM 34 (7) (1991) 40–61.
[21] R.E. Orwig, H. Chen, J.F. Nunamaker, A graphical, self-organizing approach to classifying electronic meeting output, Journal of the American Society for Information Science 48 (2) (1997) 157–170.
[22] H.J. Peat, P. Willett, The limitations of term co-occurrence data for query expansion in document retrieval systems, Journal of the American Society for Information Science 42 (5) (1991) 378–383.
[23] D. Roussinov, H. Chen, Document clustering for electronic meetings: an experimental comparison of two techniques, Decision Support Systems 27 (1–2) (1999) 67–79.
[24] D. Roussinov, K. Tolle, M. Ramsey, M. McQuaid, H. Chen, Visualizing internet search results with adaptive self-organizing maps, Proceedings of ACM SIGIR, August 15–19, 1999, Berkeley, CA.
[25] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.
[26] C.J. van Rijsbergen, A theoretical basis for the use of co-occurrence data in information retrieval, Journal of Documentation 33 (2) (1977) 106–119.
[27] J. Ward, Hierarchical grouping to optimize an objection function, Journal of the American Statistical Association 58 (1963) 236–244.
[28] J. Xu, W.B. Croft, Corpus-based stemming using co-occurrence of word variants, ACM Transactions on Information Systems 16 (1) (1998) 61–81.
[29] J.L. Zhao, A. Kumar, E.A. Stohr, A dynamic grouping technique for distributing codified-knowledge in large organizations, Proceedings of the 10th Workshop on Information Technology and Systems, December 9–10, 2000, Brisbane Australia.

Dmitri Roussinov is an assistant professor at the School of Accountancy and Information Management, College of Business, Arizona State University. He received his PhD in MIS from University of Arizona in 1999 and has a prior MA degree in Economics from Indiana University, and a diploma with honors in Computer Science from Moscow Institute of Physics and Technology, Russia. Prior to joining ASU, Dr. Roussinov served 2 years on the faculty at Syracuse University, School of Information Studies. His research interests include applications of Artificial Intelligence to Knowledge Management, Group Decisions Support Systems, and Electronic Commerce.



J. Leon Zhao is an associate professor in the MIS Department, University of Arizona. He holds a PhD degree in Information Systems from University of California, Berkeley, an MS degree from University of California, Davis, and a Bachelor of Engineering degree from Beijing Institute of Agricultural Mechanization. He has previously taught in the Hong Kong University of Science and Technology and the College of William and Mary. His research work has appeared in *Management Science*, *Information Systems Research*, *IEEE Transactions on Knowledge and Data Engineering*, *Communications of the ACM*, and *Journal of Management Information Systems* among other academic journals.