

A Knowledge-based Interactive Idea Categorizer for Electronic Meeting Systems

Jae Kyeong Kim

Department of Management Information Systems, Kyonggi University, Suwon, Korea

Jae Kwang Lee

Graduate School of Management, KAIST, Seoul, Korea

Abstract

Research on group decisions and electronic meeting systems have been increasing rapidly according to the widespread of Internet technology. Although various issues have been raised in empirical research, we will try to solve an issue on idea categorizing in the group decision making process of electronic meeting systems. Idea categorizing used at existing group decision support systems was performed in a top-down procedure and mostly by participants' manual work. This resulted in tacking as long in idea categorizing as it does for idea generating, clustering an idea in multiple categories, and identifying almost similar redundant categories. However such methods have critical limitation in the electronic meeting systems, we suggest an intelligent idea categorizing methodology which is a bottom-up approach. This method consists of steps to present idea using keywords, identifying keywords' affinity, computing similarity among ideas, and clustering ideas. This methodology allows participants to interact iteratively for clear manifestation of ambiguous ideas. We also developed a prototype system, IIC (intelligent idea categorizer) and evaluated its performance using the comparison experiment with other systems. IIC is not a general purposed system, but it produces a good result in a given specific domain.

Key Words: electronic meeting systems, group decision making, idea organization, intelligent idea categorizer

1. Introduction

Recently, many major decisions in organizations are being made by groups. Groups or work groups refer to two or more individuals whose mission is to perform some task and who act as one unit (Turban, 1993). Information technology has been used increasingly for the support of group cooperative work. Computer-based systems appear under several names for group support, such as groupware, electronic meeting systems(EMS), collaborative systems, computer-supported cooperative work(CSCW), and group decision support systems(GDSS). Generally, many electronically supported meetings follow the standard template of idea generation, idea organization and idea prioritization (Turban 1993). Idea generation is a divergent phase where groups typically engage in some kind of brainstorming activity. This phase is characterized by creativity, free-thinking, lack of critical analysis and lack of restrictions or controls. When ideas were generated, idea organization activity involves taking ideas, refining, rearranging, and consolidating the items of idea based on some organizing principle or logical framework. In this phase, groups must organize or categorize the results of the idea generation phase. The final phase of idea prioritization and action plan development requires that the output from the idea organization phase is a summarized list of the issues or topics that are most important or most relevant.

Among the three phases, idea organization is often a long and painful process for a group. The output of an electronic brainstorming session is typically extremely "dirty" or noisy. Usually many of the ideas are not expressed in complete phrases. Ungrammatical phrasing is common. Time constraints and a wide variety of typing and spelling skills commonly result in many misspelled words and typographical errors. The diverse and noisy nature of the idea generation output makes the synthesis part of a convergent task especially difficult. Idea categorizing as used in an existing system was performed

in a top-down procedure and mostly by participants' manual work. This resulted in tacking as long in idea categorizing as it does for idea generating, clustering an idea in multiple categories, and identifying almost similar redundant categories. An AI-Categorizer (AIC) is developed at the University of Arizona (Aiken et al, 1996), which automatically categorizes the output of GroupSystems V brain storming session. The tool's categorization of the ideas consists of the following three steps; automatic indexing, cluster analysis, and Hopfield net classification. In spite of many advantages, AIC is not a domain dependent tool, which results the categories generated by AIC may be too generic or may have different levels of abstraction

In this research, we suggest an interactive idea categorizing methodology, which uses a domain dependent knowledge in the form of *affinity network*. The procedure of suggested methodology consists of the following steps: presenting ideas using keywords, identifying keywords' affinity, computing similarity among ideas, and clustering ideas. The suggested methodology allows participants to interact iteratively for clear manifestation of ambiguous ideas. Furthermore, the methodology organizes the ideas without the burden of participants in a given domain. We develop a prototype system, an interactive idea categorizer (IIC) to implement and evaluate this methodology. The stored digit information can be used as an *organizational memory* in knowledge management. Compared with manual work, our suggested methodology can handle information overload problem. Compared that the AI-categorizer can fit a general domain, our suggested methodology can handle well specified domain problems. We have evaluated IIC with an example set, and compared the result with those of other methodologies. In this research, we assume that the idea organization activity began after idea generation was finished.

The scope of the research is organized as follows. The related research into idea categorizing is briefly surveyed

in section 2. In section 3, an interactive idea categorizing methodology is explained with an illustrative example. In section 4, implementation and experiment of IIC is explained. Finally, concluding remarks and further research areas are discussed in section 5.

2. Literature review

Clustering has been perceived by researchers in various domains to be a tool of discovery. It partitions a set of objects into non-overlapping subjects called clusters such that the objects inside each cluster are similar to each other and the objects from different clusters are not similar. The main focus of our research is categorizing ideas in electric meetings. The ideas generated at electronic meetings are similar to the documents. So our suggested interactive idea categorizing methodology is based on the document classification or document clustering methodology. In this section, we reviewed the document classification methodologies, a basic underlying methodology of ours.

2.1 Vector-space model

Vector-space model of Salton (1989) retrieves a specific document by a predefined similarity evaluating a given query and documents set with threshold values. The vector-space model uses an available term set to identify both stored records and information requests. Both queries and documents can be represented as term vectors of the form;

$D_i = (a_{i1}, a_{i2}, \dots, a_{it})$, and $Q_j = (q_{j1}, q_{j2}, \dots, q_{jt})$, where the coefficients a_{ik} and q_{jk} represent the values of term k in document D_i and query Q_j , respectively (Salton 1983, Raghavan 1986). Typically a_{ik} (or q_{jk}) is set equal to 1 when term k appears in document D_i (or in query Q_j), and to 0 when the term is absent from the vector. Assume a situation in which t distinct terms are available to characterize record content. Each of the t terms can then be identified with a term vector T , and a vector space is defined whenever the T vectors are linearly independent. In such a space, any vector can be represented as linear combination of the t term vectors. The r th document, D_r can be written as

$$D_r = \sum_{i=1}^t a_{ri} T_i,$$

where the a_{ri} is interpreted as the components of D_r along the vector T_i .

In vector space, the similarity between document and query is defined as,

$$D_r \bullet Q_s = \sum_{i,j=1}^t a_{ri} q_{sj} T_i \bullet T_j.$$

A similarity computation can then be used to obtain pair-wise similarity measurements between documents. Pair-wise similarity measurements between documents forming a basis for certain document-clustering systems is defined as:

$$Sim(D_r \bullet D_s) = \sum_{i,j=1}^t a_{ri} q_{sj}$$

The vector-space model can be used to obtain correlations, or similarities, between pairs of stored documents, or between queries and documents, under the assumption that the t term vectors are orthogonal, or that

the term vectors are linearly independent, so that a proper basis exists for the vector space.

2.2 Automatic document classification

The conventional document classification has been carried out manually. But the automatic approach to the classification has been tried out since late 1960's. In automatic document classification, there has been two approaches. One is to use an already fixed classification table. This is to allocate documents among the given categories. The other is to allocate documents according to the contents similarities between documents instead of a priori classification table.

2.2.1 Classification Table

The automatic document classification classifies automatically among the given categories or the generated categories by experience using a priori classification table (Hamill 1980; Borko and Bernick 1983). A disadvantage of this automatic document classification methodology, is that in many cases a priori classification table does not exist or it is difficult to build a classification category.

2.2.2 Clustering

To carry out the cluster generation, two main strategies can be used. First, a complete list of all pairwise similarities can be constructed, in that case it is necessary to employ a grouping mechanism capable of items with sufficiently large pairwise similarities to be assembled into a common cluster. Alternatively, heuristic methods can be used which do not require the computation of pairwise similarities (Salton, 1989).

When cluster generation depends on pairwise term similarities, a term-document matrix is conveniently used as a starting point, followed by a comparison of all distinct pairs of matrix rows to be used for document clustering. The pairwise comparison of matrix columns produces $N(N-1)/2$ different pairwise term similarity coefficients for the documents, where N represents the number of documents. No matter what specific clustering method is used, the clustering process can be carried out either divisively or agglomeratively. In general case, the complete collection is assumed to represent one complete cluster that is subsequently broken down into smaller pieces. In the latter, individual similar items are used as a starting point, and a gluing operation collects similar items, or groups, into larger groups (Griffiths, Robinson and Willett 1984). Several methods using graph theory have been proposed to generate several clusters. The representative methods are single-link clustering, complete-link clustering, and group-average clustering (Salton 1989). The hierarchical clustering strategies are based on prior knowledge of all pairwise similarities between items (Jardine 1971). Therefore the corresponding cluster-generation methods are relatively expensive to perform. In return, these methods produce a unique set of well-formed clusters for each set of data, regardless of the order in which the similarity pairs were introduced into the clustering process. Please refer (Salton, 1989) for a more detail.

3. The interactive methodology for idea categorizing

3.1 Overview of the methodology

In this research, an interactive methodology is suggested to categorize the ideas generated at brainstorming session.

The procedure can be explained in two parts, calculation of the similarity between ideas and interactive clustering of the ideas. The process of calculating similarities between ideas composed of the following three steps. First, the ideas from electronic brainstorming are included according to the affinity between keywords. Second, weighted idea matrix is constructed, where the weight of idea is determined from the keyword frequencies. Finally, the similarities between all ideas are computed based on the weighted idea matrix. After the similarity matrix is constructed, the clustering process is performed. Single linked clustering algorithm performs the clustering process automatically, but if an idea pair has a value in predefined *indifference level*, it is necessary an interactive iteration procedure which obtains the opinion of facilitators. Figure 1 presents the overall procedure.

Figure 1. The procedure of interactive idea clustering.

Next sections contain the key algorithms such as idea indexing based on keyword and its affinity, generating a weighted idea matrix, generating a similarity matrix, and interactive clustering with an illustrative example.

3.2 Idea indexing using keywords and its affinity

In this research, the generated n ideas are represented as keyword vectors of the form

$I_i = (a_{i1}, a_{i2}, \dots, a_{in})$, where the coefficient a_{ik} represents the value of keyword k in idea I_i . Typically a_{ik} is to be 1 when keyword k appears in idea I_i , and 0 when keyword k is absent in idea I_i . We regard all words except grammatical function words such as “and”, “of”, “or”, and “but” in the composition of written text as *keywords*. Figure 2 shows an example of initial idea matrix, where six ideas are represented by nine keywords.

		Keywords								
		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
I	1	1	0	0	1	1	0	0	0	0
	1									
d	1	0	1	1	0	0	0	0	0	0
	2									
e	1	0	0	0	0	0	1	1	1	1
	3									
a	1	0	1	1	1	0	0	0	0	0
	4									
s	1	0	0	0	1	0	1	0	1	1
	5									
l	1	1	0	1	0	0	1	0	0	1
	6									

Figure 2. An example of initial idea matrix.

Keywords alone are not enough to represent the ideas, so this research uses synonym to represent the idea exactly. The affinity value among synonyms has a number between 0 and 1, and it is stored at a network-type knowledge base. For example, the “school” and “collage” may be used as a synonym, and the affinity exists between the two words. The keyword affinity network for the above example is assumed to be stored at knowledge base as the following Figure 3.

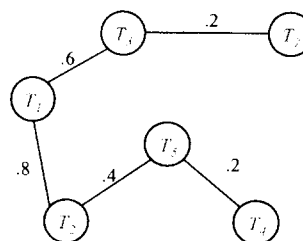
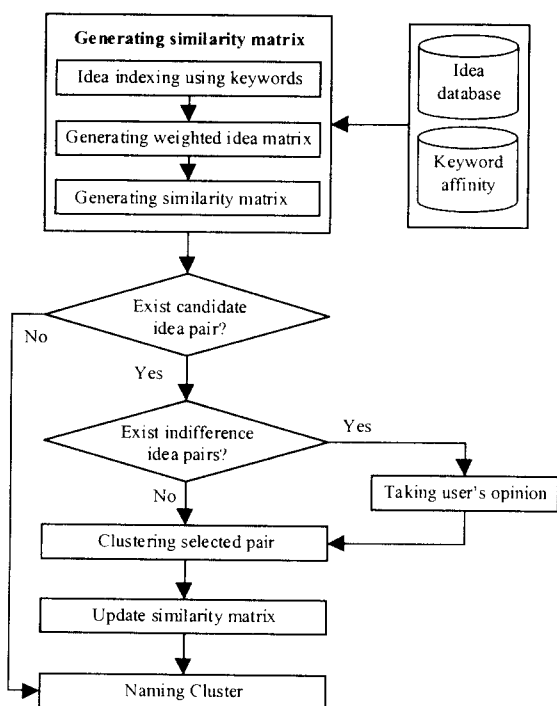


Figure 3. Keyword affinity network for the example.



After the initial idea matrix is generated, an idea matrix R is constructed based on the initial idea supplemented from the keyword affinity of the keyword affinity network. In that case, it is computed an *affinity value* among all keywords in index, which represents the degree of similarity. If a directed link between keywords, T_i and T_j is computed by the following equation;

$$Affinity(T_i, T_j) = \text{Max}\{\text{Min}[Affinity(T_i, T_k), Affinity(T_k, T_j)]\},$$

$$k = 1, \dots, n.$$

The affinity values between all keywords in Index are computed. If $Affinity(T_i, T_j)$ is not zero between keywords T_i and T_j , where a_{ki} (the value of keyword i in idea I_k) is zero, and a_{kj} is one, then a_{ki} is replaced by $Affinity(T_i, T_j)$. For example, assume that keyword affinity network is given like Figure 3, where the affinity value between keyword T_2 and T_1 is 0.8. At first, only keywords T_1 , T_4 , and T_5 are appeared in I_1 , but considering the keyword affinity, I_1 becomes related with keywords T_2 , T_3 , and T_7 also. It will be found that a_{12} is replaced with 0.8, $Affinity(T_1, T_2)$. Please refer appendix 1 for the algorithm generating idea matrix reflecting keyword affinity from initial idea matrix and keyword affinity network. Figure 4 shows idea matrix reflecting keyword affinity relations.

		Keywords								
		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
I	I_1	1	.8	.6	1	1	0	.2	0	0
	I_2	.8	1	1	.2	.4	0	.2	0	0
d	I_3	.2	.2	.2	.2	.2	1	1	1	1
	I_4	.8	1	1	1	1	0	.2	0	0
e	I_5	.2	.2	.2	1	.2	1	.2	1	1
	I_6	1	.8	1	.2	1	0	.2	0	1

Figure 4. Idea matrix reflecting keyword affinity.

3.3 Generating weighted idea matrix

Idea is represented in vector form by keywords. However, the keywords of each idea have different degree of importance. We represent the degree of keyword importance as weights. The keyword weight of each idea is determined by the ratio of the frequency of a keyword to the sum of frequencies of keywords of the idea. For example, $T_1, T_2, T_3, T_4, T_5,$ and T_7 are the keywords with non-zero value in idea I_1 of Figure 4.

		Keywords								
		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
I	I_1	.1	.12	.09	.2	.23	0	.0	0	0
	I_2	.5			3			2		
d	I_3	.1	.15	.15	.0	.09	0	.0	0	0
	I_4	.2			5			2		
e	I_5	.0	.02	.02	.0	.03	.1	.0	.1	.15
	I_6	.2			3			5		
a	I_7	.1	.15	.15	.2	.23	0	.0	0	0
	I_8	.2			3			2		
s	I_9	.0	.02	.02	.1	.03	.1	.0	.1	.15
	I_{10}	.2			5			1		
6	I_{11}	.1	.1	.13	.0	.19	0	.0	0	.19
	I_{12}	.3			4			1		

Figure 5. Weighted idea matrix of the example.

The frequencies of those keywords are 2, 2, 2, 3, 3, and 1 respectively. The sum of the frequencies is 13, so the weight of T_1 in idea I_1 becomes $2/13$. If the number of ideas and keywords are m and n respectively, the keyword weights are represented by the following weight matrix W ,

$$W = \|w_{ij}\|_{m \times n} \quad 0 \leq w_{ij} \leq 1.$$

The weighted idea matrix D is generated from the idea matrix R multiplied by the weight matrix W ,

$$D = W \otimes R, \text{ where } R = \|r_{ij}\|_{m \times n}, \quad W = \|w_{ij}\|_{m \times n}, \text{ and}$$

$$\|d_{ij}\|_{m \times n} = \|w_{ij} \cdot r_{ij}\|_{m \times n} \quad 0 \leq d_{ij} \leq 1.$$

A detailed algorithm for generating W is given at appendix 2. Figure 5 shows the weighted idea matrix of the example. Please notify that it is omitted a keyword weight matrix W without loss of generosity.

3.4 Generating similarity matrix

A similarity matrix represents the degree of similarity between the ideas. The basic idea is that if two ideas have similar keywords and their frequencies are also similar, then it is concluded that two ideas are similar and is grouped into a same cluster. The similarity degree between two ideas is generated from a weighted idea matrix. In this research, one idea is represented by n -dimensional vector, D_i , where $i = 1, \dots, m$, and m is the number of ideas. So an idea vector set D becomes a set whose elements are m idea vectors. Therefore, a similarity matrix S , represented by $m \times m$ matrix, is computed as follows;

$$D = \{D_i\}_{i=1,m} \quad D_i = (d_{ij})_{j=1,n}, \quad S = \|Sim(D_i, D_j)\|,$$

$$Sim(D_i, D_j) = \frac{D_i \cdot D_j}{|D_i| \cdot |D_j|} = \cos \theta \quad 0 \leq \theta \leq \frac{\pi}{2}.$$

Hence, the similarity degree or similarity value between ideas is represented by the cosine value of vector D_i and vector D_j . Figure 6 presents the similarity matrix.

		Ideas					
		I_1	I_2	I_3	I_4	I_5	I_6
Ideas	I_1	.	.79	.26	.98	.48	.72
	I_2	.79	.	.23	.85	.26	.76
	I_3	.26	.23	.	.26	.87	.58
	I_4	.98	.85	.26	.	.47	.74
	I_5	.48	.26	.87	.47	.	.53
	I_6	.72	.76	.58	.74	.53	.

Figure 6. Similarity matrix of the example.

3.5 Interactive clustering

Based on a complete list of all pairwise similarities, our suggested interactive idea clustering methodology groups ideas with sufficiently large pairwise similarities into one cluster. The basic idea of our methodology is as follows: First, an idea pair with the highest similarity value is grouped into one cluster. Second, if the difference between the highest similarity value and the second highest value is within a given *indifference value*, we call the two idea pairs as indifferent idea pairs, and an interactive procedure is occurred to determine which pair is to be selected regarding DMs' domain specific knowledge. Third, a clustering procedure is continued until the similarity value of remaining idea pair is below the threshold value. Therefore, two strategies are possible about selecting next idea pair. One is an automatic procedure that relies on similarity values only and the interaction with the coordinator or group DMs is not occurred. The other one is an interactive procedure that depends on similarity values and a predefined indifference value. The automatic procedure is a special case of the interactive procedure when the indifference value is zero. Therefore an interactive procedure is explained hereafter and the performance between the two approaches are discussed at next section.

Figure 7. Clustering procedure of the example.

The overall procedure of the interactive idea clustering methodology is as follows:

Step 1. Initializing indifference value. A coordinator or group DMs decide a threshold value and an

	I_1	I_2	I_3	I_4	I_5	I_6
I_1		.79	.26	.98	.48	.72
I_2	.79		.23	.85	.26	.76
I_3	.26	.23		.26	.87	.58
I_4	.98	.85	.26		.47	.74
I_5	.48	.26	.87	.47		.53
I_6	.72	.76	.58	.74	.53	

Step 1

	$I_1 I_4$	I_2	I_3	I_5	I_6
$I_1 I_4$.85	.26	.48	.74
I_2	.85		.23	.26	.76
I_3	.26	.23		.87	.58
I_5	.48	.26	.87		.53
I_6	.74	.76	.58	.53	

Step 2

	$I_1 I_4 I_2$	I_3	I_5	I_6
$I_1 I_4 I_2$.26	.48	.76
I_3	.26		.87	.58
I_5	.48	.87		.53
I_6	.76	.58	.53	

Step 3

	$I_1 I_4 I_2$	$I_3 I_5$	I_6
$I_1 I_4 I_2$.48	.76
$I_3 I_5$.48		.58
I_6	.76	.58	

Step 4

	$I_1 I_4 I_2 I_3 I_5$	I_6
$I_1 I_4 I_2 I_3 I_5$.58
I_6	.58	

Step 5

indifference value considering the importance or characteristic of a given problem. The indifference value is a value between 0 and 1.0.

Step 2. Looking for candidate idea pairs. If an idea pair should be a candidate pair, the similarity value of the pair should be larger than the threshold value. Candidate idea pairs consist of the idea pair with the highest similarity value, and the pair(s) of which similarity value is greater than the highest similarity value minus indifference value.

Step 3. Stopping condition. If there are no more candidate pairs, stop it. Otherwise go to step 4.

Step 4. Selecting one idea pair. Candidate idea pairs are suggested to the group DMs, and one idea is selected based on the majority rule of group DMs.

Step 5. Linking the idea pairs. The selected idea pair are grouped into one cluster, and the similarity matrix is updated according to single linked method. Single linked method uses a higher similarity value of selected idea pair as that of the cluster. Go to step 2.

As the indifference value is close to 1.0, the interaction with DMs is occurred many times. So the knowledge or

the preference of DMs are well cooperated but the burden of DMs are increased. If the indifference value is close to 0, the procedure becomes an automatic procedure, and the knowledge or preference of DMs can not be cooperated. The threshold value influences the number of combined clusters. If a larger threshold value is used, the ideas are less clustered, that means the number of ideas are not much decreased. Otherwise, the fact that a threshold is close to 0 implies that all the ideas are close to be one idea (cluster).

Figure 7 shows an interactive clustering procedure based on the similarity matrix of Figure 6. In this example, it is assumed that the threshold value is 0.6, and indifference value is 0.05. I_1 and I_4 are selected as a candidate idea pair because they have the highest similarity value (0.98) and the next one (0.85) is below the highest value minus indifference value. I_1 and I_4 are grouped into one cluster, and similarity matrix is updated by recalculating the similarity values of the cluster ($I_1 I_4$) and others. For example the similarity value between $I_1 I_4$ and I_2 is 0.79 at first time. After the clustering, the similarity value between ($I_1 I_4$) and I_2 becomes 0.85, because the similarity value of I_1 and I_4 is greater than I_2 and I_4 . The updated similarity matrix is given at (b). At (b), two pairs (I_3, I_5) and ($I_1 I_4, I_2$) becomes candidate idea pairs, because the similarity values of both pairs are greater than the threshold value (0.6), and the next highest value (0.85) is greater than the highest similarity value (0.87) minus indifference value (0.05). In this example, it is assumed that ($I_1 I_4, I_2$) is selected by the group DMs. So ($I_1 I_4, I_2$) becomes one cluster represented as (c). This procedure is continued until there are no more candidate pairs. The final result of our procedure is represented at (e), which shows that the six ideas are grouped into two clusters, (I_1, I_2, I_4, I_6) and (I_3, I_5).

4. Prototyping and experiments

4.1 Architecture of IIC

The interactive approach based on knowledge base proposed in this paper is implemented as a prototype system called IIC (Intelligent Idea Categorizer). IIC intends to aid the facilitator of electronic meetings. IIC consists of three major modules, Database, and Knowledge Base. The modules are User Interface, Similarity Calculator, and Interactive Cluster Generator. Figure 8 shows the system architecture and the relationships among these system components.

The User Interface module provides interactive question and answer functions. It takes the threshold value and indifference level, presents indifferent idea pairs, asks user opinions about indifferent idea pairs, accepts the answer, shows the result of idea clustering, and takes the name of the cluster. The interaction between the participants and IIC is performed at User Interface. The Similarity Calculator performs the suggested subalgorithms described at section 3.2 through 3.4. Idea indexing, idea matrix generation reflecting keyword affinity, weighted idea matrix generation, and similarity matrix generation are performed at this module. This module interacts with Database and Knowledge base. The overall procedure of Interactive Cluster Generator module is described at section 3.5. The Interactive Cluster Generator adopts a single linked clustering methodology and uses the indifference level to find indifferent idea pairs.

Ideas, keywords, and categories are stored at Database. The schema of the Database consists of three entities and three relationships. The entities are idea entity, category entity, and keyword entity. The relationship of idea and category entities has many to one cardinality. The relationship of the idea and keyword entities has many to many cardinality. The keyword entity has a recursive relationship that represents an affinity between keywords. The Knowledge Base includes domain-specific affinity between keywords. Affinities are represented by graph.

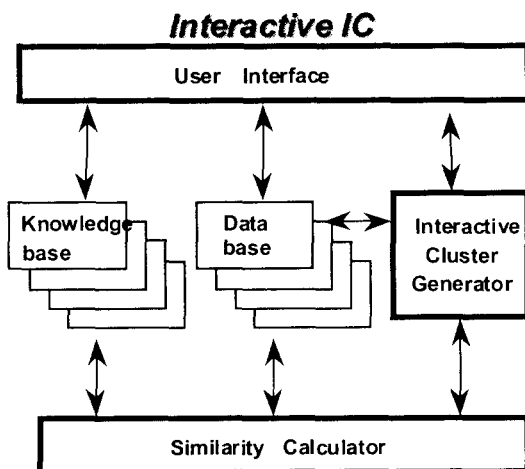
Figure 8. Architecture of IIC

4.2 Experiments

A laboratory experiment was conducted to investigate the effectiveness and efficiency of our suggested IIC (intelligent idea categorizer). The experimental plan and the number of subjects who completed the experiment is summarized in Table 1. The methodology was straightforward and similar to many previous studies. Experiment was done twice in two groups: the experiment 1 group and experiment 2 group. The subjects in each group were five graduate students, who were trained together. Two experimental topics were used during the experiments: Material purchasing problem for the experiment 1 group and Parking lot problem for the experiment 2 group. Each experiment group conducted brainstorming process about 20 minutes using the Electronic Brainstorming (EB) in GroupSystems V. The ideas elicited from the brainstorming process are used to the idea categorizing experiment. Material purchasing problem for the experiment 1 group is about purchasing research equipments such as PCs, Workstations, Overhead Projectors, and Network facilities. Experiment 1 group gathers ideas about how to purchase the equipments and the key factors influencing the decision to buy the equipments. Parking lot problem for the experiment 2 group is generating ideas to solve the insufficiency of parking lot area of the graduate school.

Table 1. Summary of the experiments design

	Experiment 1	Experiment 2
Number of participants	5 people	5 people
Meeting subjective	Material purchasing problem	Parking lot problem
Brainstorming session	20 min	20 min
Categorizing session	GroupSystems CA, IIC, W/O Interactive respectively	



Each experiment group conducted categorizing experiments using three categorizing methods. The first method is using Categorizer (CA) of GroupSystems. The second one is using our suggested IIC. The last one is also using the IIC, but the indifference level is set zero, which means that the interactive process is not necessary. As a measure of categorizing success, time of completing the categorizing process and accuracy of the categorization is used. The accuracy of the categorization is measured by the comparison of each experiment result and the compromising result. The compromising result is obtained after experiment through full time discussion between each group participants. The results of each experiment are summarized at Table 2. The number of ideas resulting from each brainstorming process are 24 and 50, respectively.

Table 2. Summary of the experiments result

		Experiment 1	Experiment 2
Number of idea		24	50
Time	GroupSystems CA	11	21
	IIC	2	3
	W/O Interactive	-	-
Incorrect classified ideas	GroupSystems CA	3	7
	IIC	2	5
	W/O Interactive	5	8

Table 2 shows that IIC makes a better performance reducing the categorizing time than CA of GroupSystems. And it shows that the accuracy of IIC is similar to that of CA of GroupSystems. However the IIC without interactivity results the worst accuracy, although it results the shortest categorizing time. The comparison experiments are not conducted under a lot of subjects and experimental design, so the result is lack of generality. However based on the opinions of experiment participants, the characteristics of the categorizing methods are summarized at Table 3. IIC carries out idea categorizing using keywords affinities, which lessens the burden of meeting participants. In the case that large number of ideas are to be categorized, IIC will be more efficient than GroupSystems. So in the electronic meeting systems where many ideas are come from many participants, IIC is believed to be a promising idea categorizer.

Table 3. Comparison of the categorizing methods

	GroupSystems CA	W/O Interactive	Interactive IC
Approach	top down	bottom up	bottom up
User	participant	facilitator	facilitator
Classification method	manual	automatic	interactive
Keyword's affinity	don't use	use	use
Idea overlapping	allow	doesn't allow	doesn't allow
Category redundancy	allow	doesn't allow	doesn't allow
Knowledge	human	keyword's affinity	keyword's affinity,
Reuse	impossible	possible	possible
Interaction with user	frequently	almost never	sometimes

V. Conclusions

Research on group decisions and electronic meeting systems have been increasing rapidly according to the widespread of internet technology. Although various issues have been raised in empirical research, this research is an effort to solve an issue on idea categorizing in the group decision making process of electronic meeting systems. The previous idea categorizing is performed mostly by participants' manual works. In comparison with the existing idea categorizing method, an interactive idea categorizing methodology proposed in this paper saves about 40% the categorizing time. The quality of categorized result is also acceptable in view that there is hardly any difference compared to the manual categorizing work which was performed for enough time. The knowledge in regard to affinities among keywords are accumulated and used later as a knowledge base. Compared that the AI-categorizer can fit a general domain, our suggested methodology can handle well specified domain problems. Under the previous manual categorizing methods, participants should classify ideas manually and intuitively while looking at whole generated ideas in a top-down manner. IIC solves these burden and inconvenience effectively. Participants with IIC can update their original intentions interactively for any ambiguous ideas. It also has an important feature that even an novice can use the system without any difficulty.

References

1. Aiken, M., and Judith Carlisle. (1992). "An automated idea consolidation tool for computer support cooperative work," *Information & Management* 23, 373-382.
2. Bookstein, A. (1981). "A Comparison of Two Systems of Weighted Boolean Retrieval," *Journal of the American Society for Information Science (JASIS)* 32(4), 275-279.
3. Borko, H., and M. Bernick. (1983). "Automatic Document Classification," *Journal of the Association for Computing Machinery* 10(1), 151-162.
4. Buell, D., (1982). "An Analysis of Some Fuzzy Subset Applications to Information Retrieval Systems," *Fuzzy Sets and Systems* 17(3), 35-42.
5. Dennis, A., et al. (1988). "Information Technology to Support Electronic Meetings," *MIS Quarterly* December.
6. Dennis, A., et al. (1991). "Group, Sub-Group and Nominal Group Idea Generation in an Electronic Meeting Environment," *System Science* 3, 573-579.
7. DeSanctis, G. and B. Gallupe. (1987) "Group Decision Support Systems: A new Frontier," *Management Science* May.
8. Griffiths, A., L.A. Robinson, and Willett. (1984). "Hierarchic Agglomerative Clustering Methods for Automatic Document Classification," *Journal of Documentation* 40(3), 175-205.
9. Hamill, K.A., and A. Zamora. (1980). "The Use of Titles for Automatic Document Classification," *Journal of the American Society for Information Science (JASIS)* 31(6), 396-402.
10. Huber, G.P. (1984) "Issues in the Design of Group Decision Support Systems," *MIS Quarterly* September.
11. Jardine, N., and C.J. van Rijsbergen. (1971). "The Use of Hierarchic Clustering in Information Retrieval," *Information Storage and Retrieval* 7(5), 217-240.
12. Jeong, Y. (1993). *The Theory of Information Retrieval*, Inc. kumi trade press.
13. Radecki, T. (1979). "Fuzzy Set Theoretical Approach to Document Retrieval," *Information Processing & Management* 15(5), 247-259.
14. Raghavan, V.V., and S.K.M. Wang. (1986). "A Critical Analysis of the Vector Space Model for Information Retrieval," *Journal of the American Society for Information Science (JASIS)* 37(5), 279-287.
15. Salton, G., and M.J. McGill. (1983). *Introduction to Modern Information Retrieval*, McGraw-Hill.
16. Salton, G. (1987). "Historical Note: The Past Thirty Years in Information Retrieval," *Journal of the American Society for Information Science*, 38(5), 375-380.
17. Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison Wesley.
18. Starub, D. and R. Beauclair. (1988). "Current and Future Uses of Group Decision Support System Technology: Report on Empirical Study," *Journal of Management Information Systems* 5(1), 101-116.
19. Tahani, V. (1976). "A Fuzzy Model of Document Retrieval Systems," *Information Processing & Management* 12(3), 177-187.
20. Turban, E. (1993). *Decision Support and Expert Systems: Management Support Systems*, Macmillan.
21. Ventana. (1992a). *Group Systems V: Basic Tools Manual*, Ventana Corporation.
22. Ventana. (1992b). *Group Systems V: Advanced Tools Manual*, Ventana Corporation.

Appendix 1: The algorithm for idea matrix reflecting keywords' affinity.

Idea matrix R is represented as keyword vectors of the form

$$R = \left\| r_{ij} \right\|_{m \times n}$$

The generating algorithm is presented as follows.

$I(i,j)$: the element of initial idea matrix I .
 $K(i,j)$: the element of keyword affinity matrix K .
 $R(i,j)$: the element of idea matrix R .
 i_n : number of ideas.
 k_n : number of kinds of keywords.

```

Initialize matrix R = 0.
For m=1 to i_n.
  For n=1 to k_n.
    If I(m,n) is 0 then.
      For j=1 to n_k.
        If I(m,j) less than K(n,j) then.
          R(m,j) = K(n,j).
        End Loop j.
      End Loop n.
    End Loop m.
  
```

Appendix 2: Generating weight matrix of keywords

The weight matrix of keywords is generated based on the frequency of keywords. Weight matrix of keywords R is represented as

$$W = \left\| w_{ij} \right\|_{m \times n} \quad 0 \leq w_{ij} \leq 1.$$

The generating algorithm is presented as follows.

keyword_frequency(j) : number of jth keyword frequency.

R(i,j) : the element of idea matrix R.

W(i,j) : the element of weight matrix W.

total : total number of keyword frequencies.

i_n : number of ideas.

k_n : number of keywords.

Initialize matrix W with 0.

For m=1 to i_n.

For n=1 to k_n.

If R(m,n) is not 0 then.

total = total + keyword_frequency(n).

End Loop n.

For j=1 to k_n.

W(m,j) = keyword_frequency(j) / total.

End Loop j.

End Loop m.