인터넷 게시판 질문 분류를 위한 인터랙티브 접근방법에 관한 연구

An Interactive Approach to Categorize Questions on the Internet BBSs

이재광(Jae-Kwang Lee)*, 노성호(Seong-Ho Noh)*, 류옥현(Ok-Hyun Ryou)*

초 록

전통적인 고객지원방법에서는 콜센터와 서비스 센터가 고객의 질문과 요구 사항을 접수하고 응대하는 기능을 담당해왔다. 최근 인터넷의 급속한 확산에 따라 전화, 우편, 방문 등의 전통적인 고객과의 의사소통 수단이 전자우편과 인터넷 게시판과 같은 웹기반의 고객지원시스템으로 전환되고 있다. 인터넷 게시판은 기본적으로 고객의 질문에 관리자가 응답하는 시스템이므로 고객이 응답을 받는데 시간이 걸리는 제약이 있다. 이러한 시간적 제약을 해결하기위하여 고객이 인터넷을 통하여 고객지원시스템에 접속하여 미리 구축된 지식 데이터베이스로부터 원격에서 질문에 대한 응답을 받을 수 있도록 공통적인 질문과 응답을 FAQ와 같은 형태를 제공한다. 그리고, 인터넷 게시판에 다양한 내용과 형태의 질문이 혼재되어 사용됨으로써 응답과 관리상의 어려움이 많다. 따라서 질문들을 체계적으로 분류하여 FAQ를 만들고, 인터넷 게시판의 관리작업을 지원하기위한 도구의 필요성이 대두되고 있다. 본 연구에서는 키워드와 키워드들간의 친밀도를 이용하여 벡터형태로 표현한 질문들간의 유사도를 계산하여 질문들을 클러스터링하는 방법을 제안한다. 제안한 방법은 기본적으로 자동으로 질문들을 분류하지만, 내용이 애매모호한 질문의 경우 사용자가 상호작용을 통하여 사용자의 판단을 받아들일 수 있도록 개발되었다. 그리고, 제안한 방법의 성능을 평가하기위하여 프로토타입 시스템을 개발하고 제한된 상황하에서 실험을 수행하였다.

ABSTRACT

In a traditional customer support environment, mainly call centers or service centers are responsible for receiving inquiries from their customers via telephone calls. Due to the rapid growth of Internet with its widespread acceptance and accessibility, means of communication with customers in the traditional customer support center, such as telephones, letters, and direct-visiting, have been replaced by e-mails and bulletin board systems (BBSs) using the Internet constantly. BBSs are basically question and answer systems, they require some lead time to get answer from administrator. To reduce lead time, BBSs enable remote customers or users to log on and tap into a knowledge database that is generally formatted in the form of Frequently Asked Questions (FAQs) that provide answers and solutions to the common problems. And, many different types of the questions are mixed on the BBS. It is a burden to administrator. To build FAQs and to support BBS adminstrator, a supporting tool which is to categorize questions is helpful. In this research, we suggest an interactive question categorizing methodology which consists of steps to present question using keywords, identifying keywords' affinity, computing similarity among questions, and clustering questions. This methodology allows users to interact iteratively for clear manifestation of ambiguous questions. We also developed a prototype system, IQC (interactive question categorizer) and evaluated its performance using the comparison experiments with other systems. IQC is not a general purposed system, but it produces a good result in a given specific domain.

키워드 : 질문분류, 전자게시판, 웹기반 고객지원시스템,
Question Categorizing, Question Classification, BBS, Web-based Customer Support Systems

* 한국산업기술대학교 e-비즈니스학과

1. Introduction

Recently, customer support in organizations is one of the important business improvement themes to improve their business competences. Many firms have realized, as their marketplaces have become more global and service oriented, that customer support is critical to their competitiveness [11]. Therefore, most of all firms currently provide web-based customer support access or online help desk [10]. For example, Microsoft's Web-based customer support gets over 100,000 unique customer visit per day. By handling this volume of customer support online, Microsoft has been able to maintain a constant level of phone support during a period of sales growth. At Novell, web-based customer support has reduced phone support by 45%. At Network Associates and Great Plains, Web-based customer support has reduced phone call volume by 37 and 20%, respectively [16]. Many firms are taking advantage of the Web-based customer technologies to give customers direct access to their customer support knowledge base or his agent [5]. The typical form web-based customer support system is the bulletin board systems (BBSs).

In a traditional customer support environment, mainly call centers or service centers are responsible for receiving inquiries from their customers via telephone calls. Due to the rapid growth of Internet with its widespread acceptance and accessibility, means of communication with customers in the traditional customer support center, such as telephones, letters, and direct-visiting, have been replacing by e-mails and bulletin board systems (BBSs) using the Internet constantly [4].

BBSs have brought out some issues. First, as BBSs are basically question and answer systems, they require some lead time to get answer from administrator [4]. Customers are dissatisfied with late resposes of the BBSs. The response lead time of the BBSs is in contrast to telephones. To reduce lead time, BBSs enable remote customers or users to log on and tap into a knowledge database that is generally formatted in the form of Frequently Asked Questions (FAQs) that provide answers and solutions to the common problems [6]. To build FAQs, we have have to categorize and analize previes questions in the BBS. Second, many different types of the questions are mixed on the BBS. Especially a large sized company which has several divisions and deals with lots of products is always distressed by the problem. For example, BBS of the company many different type questions such as recruiting, product information, IR information, after service and so on. The categorizing questions into right category are difficult and time-consuming work. To categorize questions, an automatic tool is helpful. An automatic question routing systems using machine learning is suggested, which automatically categorizes the questions on the BBSs [4]. The study discussed some categorization method based on machine learning. In spite of many advantages, the study shows low accuracy of classification result, 52% from 63%.

In this research, we suggest an interactive question 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 questions using keywords, identifying keywords' affinity, computing similarity among questions, and clustering questions. The suggested methodology allows administrator to interact iteratively for the clear manifestation of ambiguous questions. Furthermore, the methodology organizes the questions without the burden of BBS administrator in a given domain. We develop a prototype system, an interactive question categorizer (IQC) to implement and evaluate this methodology. Compared with manual work, our suggested methodology can handle information overload problem. We have evaluated IQC with an example set, and compared the result with manual work. In this research, the main focus is the question categorizing on the BBSs.

In addition, the stored questions can be used as the voice of customer (VOC) in customer relationship management. To gather VOC, many companies have tried to use BBS in their efforts. However, to discover knowledge about customer such as customer needs and new business opportunities, categorizing of questions is required from BBS. The suggested methodology can be used as customer knowledge discovery.

The scope of the research is organized as follows. The related research into question categorizing is briefly surveyed in section 2. In section 3, an interactive question categorizing methodology is explained with an illustrative example. In section 4, the system implementation and the result of the experiment 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 questions on the BBSs. The questions posted on BBSs are similar to the documents. So we developed an interactive methodology based on the existing document classification or document clustering methodologies. In this section, we reviewed the previous document classification methodologies in brief.

2.1 Vector-space model

Vector-space model of Salton retrieves a specific document by a predefined similarity evaluating a given query and documents set with stopping values [14]. 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:

$$D_i = (a_{i1}, a_{i2}, \ldots\ldots, a_{it}) ,\ \text{and}$$
$$Q_j = (q_{j1}, q_{j2}, \ldots\ldots, q_{jt}) ,$$

where the coefficients a_{ik} and q_{jk} represent the values of term k in document D_i and query Q_j, respectively [12, 13, 14]. 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

$D_r = \sum_{i=1} a_{ri} T_i$, where the ari 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,

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

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

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

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 have 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 [2, 8]. 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 simiarities to be assembled into a common cluster. Alternatively, heuristic methods can be used which do not require the computation of pairwise similarities [9, 14].

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 [7]. 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 [14]. The hierarchical clustering strategies are based on prior knowledge of all pairwise similarities between items [9]. 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 paper [14] for a more detail.

3. The interactive methodoloy for Questions Categorizing

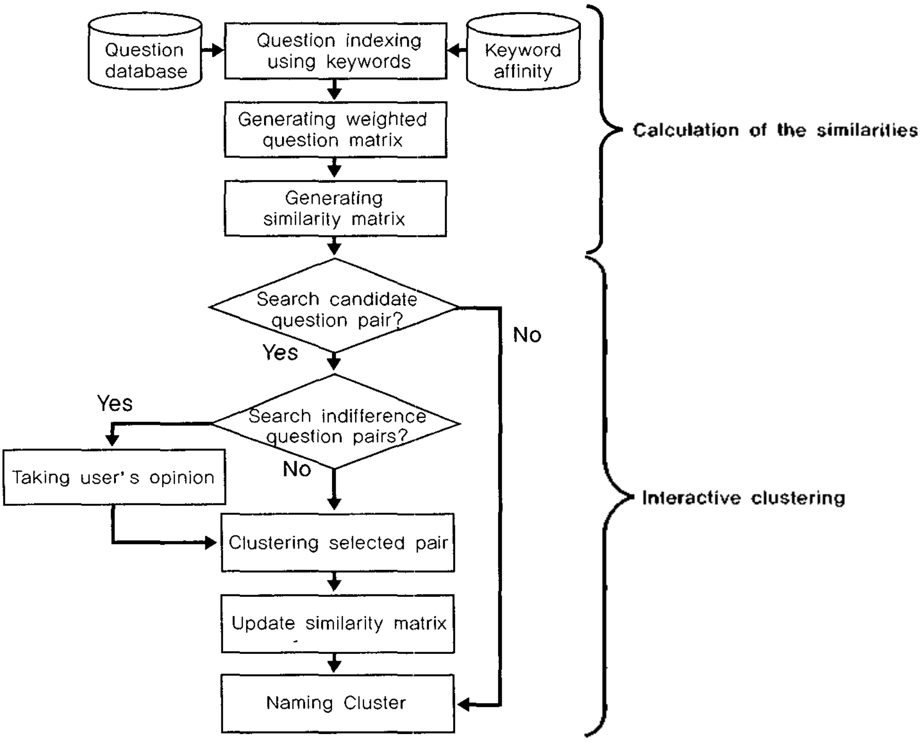
3.1 Overview of the methodology

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

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

3.2 Question indexing using keywords and its affinity

In this research, the generated n questions


Question database
Question indexing using keywords
Keyword affinity
Generating weighted question matrix
Generating similarity matrix
Calculation of the similarities
Search candidate question pair?
Yes
No
Search indifference question pairs?
Yes
No
Taking user's opinion
Clustering selected pair
Update similarity matrix
Naming Cluster
Interactive clustering

〈Figure 1.〉 The overall procedure of interactive question categorizing.

are represented as keyword vectors of the form

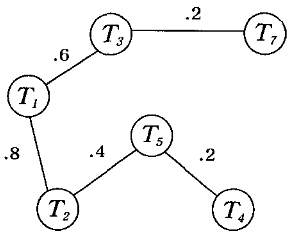
$$Q_k = (a_{k1}, a_{k2}, ..., a_{kn})$$

, where the coefficient aki represents the value of keyword i in question Q_k. Typically aki is to be 1 when keyword i appears in question Q_k, and 0 when keyword i is absent in question Q_k. 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 question matrix, where six questions are represented by nine keywords.

Keywords alone are not enough to represent the questions, so this research uses synonym to represent the questions 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 "customer" and "consumer" 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 in advance as the following Figure 3.

		Keywords								
		T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Questions	Q_1	1	0	0	1	1	0	0	0	0
	Q_2	0	1	1	0	0	0	0	0	0
	Q_3	0	0	0	0	0	1	1	1	1
	Q_4	0	1	1	1	1	0	0	0	0
	Q_5	0	0	0	1	0	1	0	1	1
	Q_6	1	0	1	0	0	1	0	0	1

〈Figure 2.〉 An example of initial question matrix



〈Figure 3.〉 Keyword affinity network for the example

After the initial question matrix is generated, a question matrix R is constructed based on the initial question 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 does not exit in keyword network, the affinity value between keywords, T_i and T_j is computed by the following equation;

$$Affinity(T_i,T_j) = Max\{Min[Affinity(T_i,T_k), Affinity(T_i,T_j)]\},\ k=1,...,n.$$

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 question 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 assumed to be 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 Figure 4 for the algorithm generating question matrix reflecting keyword affinity from initial question matrix and keyword affinity network. Figure 5 shows the question matrix reflecting keyword affinity relations.

Question matrix R is represented as keyword vectors of the form

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

```
The generating algorithm is presented as follows.
      I(i,j) : the element of initial question matrix I.
      K(i,j) : the element of keyword affinity matrix K.
      R(i,j) : the element of question matrix R.
      i_n : number of questions.
      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.
```

〈Figure 4.〉 The algorithm for question matrix reflecting keywords' affinity.

$$\begin{array}{ll|ccccccccc|}
 & & \multicolumn{9}{c}{\text{Keywords}} \\
 & & T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & T_9 \\
 & Q_1 & 1 & .8 & .6 & 1 & 1 & 0 & .2 & 0 & 0 \\
 & Q_2 & .8 & 1 & 1 & .2 & .4 & 0 & .2 & 0 & 0 \\
\text{Questions} & Q_3 & .2 & .2 & .2 & .2 & .2 & 1 & 1 & 1 & 1 \\
 & Q_4 & .8 & 1 & 1 & 1 & 1 & 0 & .2 & 0 & 0 \\
 & Q_5 & .2 & .2 & .2 & 1 & .2 & 1 & .2 & 1 & 1 \\
 & Q_6 & 1 & .8 & 1 & .2 & 1 & 0 & .2 & 0 & 1
\end{array}$$

〈Figure 5.〉 Question matrix reflecting keywords' affinity

3.3 Generating weighted question matrix

Question is represented in vector form by keywords. However, the keywords of each question have different degree of importance. We represent the degree of keyword importance as weights. The keyword weight of each question is determined by the ratio of the frequency of a keyword to the sum of frequencies of keywords of the question. For example, T_1, T_2, T_3, T_4, T_5, and T_7 are the keywords with non-zero value in question Q_1 of Figure 5. 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 question Q_1 becomes 2/13. If the number of questions and keywords are m and n respectively, the keyword weights are represented by the following weight matrix W.

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

The weighted question matrix D is generated from the question matrix R multiplied by the weight matrix W.

$$D = W \otimes R, \text{ where } W = \left\| w_{ij} \right\|_{m \times n},$$

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

$$\text{and } \left\| d_{ij} \right\|_{m \times n} = \left\| w_{ij} \cdot w_{ij} \right\|_{m \times n} \; 0 \leq d_{ij} \leq 1.$$

A detailed algorithm for generating W is given at Figure 6. Figure 7 shows the weighted question matrix of the example. Please notify that it is omitted a keyword weight matrix W.

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

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

The generating algorithm is presented as follows.

keyword_frequency(j) : number of jth keyword frequency.
R(i,j) : the element of question matrix R.
W(i,j) : the element of weight matrix W.
total : total number of keyword frequencies.
i_n : number of questions.
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.
```

〈Figure 6.〉 Generating weight matrix of keywords

$$\begin{array}{cc|ccccccccc|}
 & & \multicolumn{9}{c}{\text{Keywords}} \\
 & & T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & T_9 \\
 & Q_1 & .15 & .12 & .09 & .23 & .23 & 0 & .02 & 0 & 0 \\
 & Q_2 & .12 & .15 & .15 & .05 & .09 & 0 & .02 & 0 & 0 \\
\text{Questions} & Q_3 & .02 & .02 & .02 & .03 & .03 & .1 & .05 & .1 & .15 \\
 & Q_4 & .12 & .15 & .15 & .23 & .23 & 0 & .02 & 0 & 0 \\
 & Q_5 & .02 & .02 & .02 & .15 & .03 & .1 & .01 & .1 & .15 \\
 & Q_6 & .13 & .1 & .13 & .04 & .19 & 0 & .01 & .1 & .19
\end{array}$$

〈Figure 7.〉 Weighted question matrix of the example

3.4 Generating similarity matrix

A similarity matrix represents the degree of similarity between the questions. The basic question is that if two questions have similar keywords and their frequencies are also similar, then it is concluded that two questions are similar and is grouped into a same cluster. The similarity degree between two questions is generated from a weighted question matrix.

Questions

		Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
	Q_1	·	.79	.26	.98	.48	.72
	Q_2	.79	·	.23	.85	.26	.76
Questions	Q_3	.26	.23	·	.26	.87	.58
	Q_4	.98	.85	.26	·	.47	.74
	Q_5	.48	.26	.87	.47	·	.53
	Q_6	.72	.76	.58	.74	.53	·

〈Figure 8.〉 Similarity matrix of the example.

In this research, one question is represented by n-dimensional vector, Di, where i = 1,⋯ m, and m is the number of questions. So a question vector set D becomes a set whose elements are m question vectors. Therefore, a similarity matrix S, represented by $m \times n$ matrix, is computed as follows;

$D = \{D_i\}_{i=1,m}$ $D_i = (d_{ij})_{j=1,n}$ where d_{ij} is the jth value of question vector D_i.

$S = \| sim(D_i, D_j) \|$,

where $Sim(D_i, D_j) = \frac{D_i \cdot D_j}{|D_i| \cdot |D_j|} = \cos\theta\, 0 \le \theta \le \frac{p}{2}$.

Hence, the similarity degree or similarity value between questions is represented by the cosine value of vector D_i and vector D_j. Figure 8 presents the similarity matrix.

3.5 Interactive clustering

Based on a complete list of all pairwise similarities, our suggested interactive question clustering methodology groups questions with sufficiently large pairwise similarities into one cluster. The basic question of our methodology is as follows: First, a question 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 question pairs as indifferent question pairs, and an interactive procedure is occurred to determine which pair is to be selected regarding administrators' domain specific knowledge. Third, a clustering procedure is continued until the similarity values of remaining question pairs are below the stopping value. Therefore, two strategies are possible about selecting next question pair. One is an automatic procedure that relies on similarity values only and the interaction with the facilitator 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 is discussed at next section.

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

Step (1) Initializing indifference value: A facilitator decides a stopping value and an 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 question pairs: If a question pair should be a candidate pair, the similarity value of the pair should be larger than the stopping value. Candidate question pairs consist of the question pairs with the highest similarity value, and the other 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 question pair: Candidate question pairs are suggested to the facilitator, and one question is selected.

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

As the indifference value is close to 1.0, the interaction with facilitator is occurred many times. So the knowledge or the preference of facilitator is well cooperated but the burden of facilitator is increased. If the indifference value is close to 0, the procedure becomes an automatic procedure, and the knowledge or preference of facilitator can not be cooperated. The stopping value influences the number of combined clusters. If a larger stopping value is used, the questions are less clustered, that means the numbers of questions are not much decreased. Otherwise, the fact that a stopping is close to 0 implies that all the questions are close to be one question (cluster).

Figure 9 shows an interactive clustering procedure based on the similarity matrix of Figure 8. In this example, it is assumed that the stopping value is 0.6, and indifference value is 0.05. *Q1* and *Q4* are selected as a candidate question pair because they have the highest similarity value (0.98) and the next one (0.85) is below the highest value minus indifference value. *Q1* and *Q4* are grouped into one cluster, and the similarity matrix is updated by recalculating the similarity values of the cluster (*Q1* *Q4*) and others. For example the similarity value between *Q1* and *Q2* is 0.79 at first time. After the clustering, the similarity value between

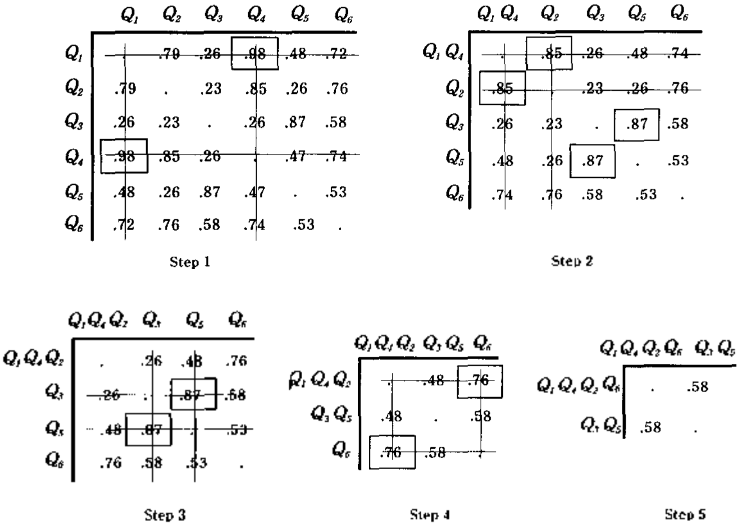
(Q_1 Q_4) and Q_2 becomes 0.85, because the similarity value of Q_1 and Q_4 is greater than Q_2 and Q_4. The updated similarity matrix is given at step2. At step2, two pairs (Q_3; Q_5) and (Q_1 Q_4: Q_2) becomes candidate question pairs, because the similarity values of both pairs are greater than the stopping 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 (Q_1 Q_4; Q_2) is selected by the facilitator. So (Q_1 Q_4 Q_2) becomes one cluster represented as step3. This procedure is continues until there are no more candidate pairs. The final result of our procedure is represented at step5, which shows that the six questions are grouped into two clusters, (Q_1 Q_2 Q_4 Q_6) and (Q_3 Q_5).

4. System Implementation and Experiment

4.1 Architecture of IQC

The interactive approach based on knowledge base proposed in this paper is implemented as a prototype system called IQC (Interactive Question Categorizer). IQC intends to aid the administrator of the BBSs. IQC consists of Database, Knowledge-base, and three major modules. The modules are User Interface, Similarity Calculator, and Interactive Cluster Generator. Figure 10 shows the system architecture and the relationships among these system components.

The User Interface module provides interactive question and answer functions. It takes the



	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
Q_1	.	.79	.26	.98	.48	.72
Q_2	.79	.	.23	.85	.26	.76
Q_3	.26	.23	.	.26	.87	.58
Q_4	.98	.85	.26	.	.47	.74
Q_5	.48	.26	.87	.47	.	.53
Q_6	.72	.76	.58	.74	.53	.

Step 1

	Q_1 Q_4	Q_2	Q_3	Q_5	Q_6
Q_1 Q_4	.	.85	.26	.48	.74
Q_2	.85	.	.23	.26	.76
Q_3	.26	.23	.	.87	.58
Q_5	.48	.26	.87	.	.53
Q_6	.74	.76	.58	.53	.

Step 2

	$Q_1 Q_4 Q_2$	Q_3	Q_5	Q_6
$Q_1 Q_4 Q_2$	.	.26	.48	.76
Q_3	.26	.	.87	.58
Q_5	.48	.87	.	.53
Q_6	.76	.58	.53	.

Step 3

	$Q_1 Q_4 Q_2$	$Q_3 Q_5$	Q_6
$Q_1 Q_4 Q_2$	.	.48	.76
$Q_3 Q_5$	.48	.	.58
Q_6	.76	.58	.

Step 4

	$Q_1 Q_4 Q_2 Q_6$	$Q_3 Q_5$
$Q_1 Q_4 Q_2 Q_6$	.	.58
$Q_3 Q_5$	.58	.

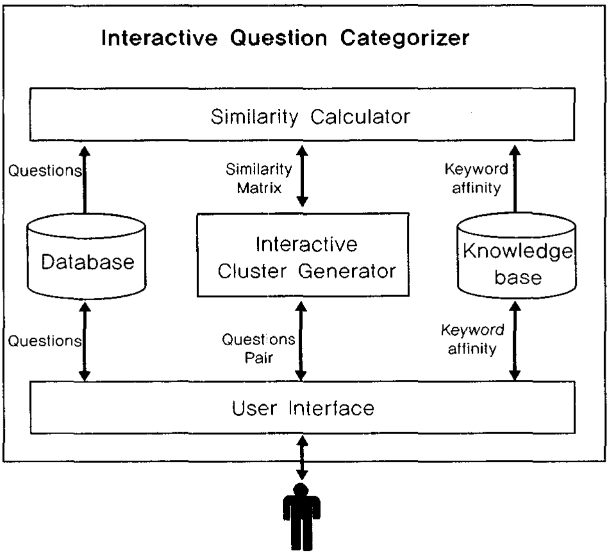
Step 5

〈Figure 9.〉 Clustering procedure of the example stopping value and indifference level, presents indifferent question pairs, asks user opinions about indifferent question pairs, accepts the answer, shows the result of question clustering, and takes the name of the cluster. The interaction between the administrator and IQC is performed at User Interface. The Similarity Calculator performs the suggested subalgorithms described at section 3.2 through 3.4. Question indexing, question matrix generation reflecting keyword affinity, weighted question matrix generation, and similarity matrix generation are performed at the 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 question pairs.

Questions, keywords, and categories are stored at Database. The schema of the Database consists of three entities and three relationships. The entities are question entity, category entity, and keyword entity. The relationship of question and category entities has many-to-one cardinality. The relationship of the question 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.

4.2 Experiments

A laboratory experiment was conducted to



〈Figure 10.〉 Architecture of IQC

investigate the effectiveness and efficiency of the suggested IQC (interactive question categorizer). The experimental plan and the number of BBS that completed the experiment are summarized in Table 1. Three BBSs of under graduate school were used during the experiments: Dept. of e-business, Dept. of computer engineering, and Dept. of electonics. There are questions on the BBS about lecture, absence, and readmit and so on. The numbers of questions on the each BBS are 500, 1000 and 2000, respectively.

Each experiment conducted categorizing questions using three categorizing approach. The first is carried out manually. The second one is using our suggested IQC. The last one is also using the IQC, but the indifference level is set zero, which means that the interactive process is not necessary.

The most important measures for retrieval system evaluation are (1) ability of the system to retrieve wanted information, (2) ability of the system to reject unwanted information. Several evaluation studies use test methodology based mainly recall value and precision value that apply to a set of test similarities [16]. Originally recall is the proportion of relevant material actually retrieved: precision (accuracy) is the proportion of retrieved material which is relevant. However, in this research, we redefined recall value and accuracy. Recall is the proportion of relevant question actually categorized: precision (accuracy) is the proportion of categorized question which is relevant.

Recall = (Number of relevant questions categorized) / (Total number of relevant questions in BBS)

Precision = (Number of relevant questions categorized) / (Total number of questions categorized)

As a measure of categorizing success, *time* of completing the categorizing process, *recall value*, and *accuracy* of the categorization is used. The relevant category 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 all participants. The results of each experiment are summarized at Table 2.

〈Table 1.〉 Summary of the experiments design

	Experiment 1	Experiment 2	Experiment 3
BBS name	Dept. of e-Business	Dept. of Computer Engineering	Dept. of Electronics
Number of questions	500	1000	2000
Categorizing approach	Manual, IQC, W/O Interactive respectively		

〈Table 2.〉 Summary of the experiments result

		Experiment 1	Experiment 2	Experiment 3
Number of questions		500	1000	2000
Number of irrelevant questions (not categorized manualy)		57	117	249
Time	Manual	46 min	73 min	145 min
	IQC	8 min	15 min	32 min
	W/O Interactive	-	-	-
Recall value	Manual	-	-	-
	IQC	87.36%	90.37%	91.21%
	W/O Interactive	61.85%	60.82%	59.17%
Accuracy	Manual	-	-	-
	IQC	83.23%	86.27%	84.50%
	W/O Interactive	54.80%	53.70%	51.80%

Table 2 shows that IQC make a better performance reducing the categorizing time than manual approach. And it shows that the recall value and accuracy of IQC is not bad. However the IQC without interactiveness results the worst, although it results the shorttest categorizing time. The comparison experiments are not conducted under a lot of BBSs and experimental design, so the result is short of generosity. IQC carries out question categorizing using keywords affinities, which lessens the burden of administrator. In the case that large numbers of questions are to be categorized, IQC will be more efficient than manual approach. So on the BBSs where many questions are come from many customers, IQC is believed to be a promising question categorizer.

5. Conclusions

Researches on web-based customer support systems have been increasing rapidly according to the widespread of Internet technology. Although BBSs have various advanages, they also have some issues such as long response time, the burden of admistrator, and customer knowledge discovery. This research is an effort to solve the issues on question categorizing from the BBS. As a prototype system, IQC is developed based on the methodology. The methodology proposed in this paper save 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 that was performed for enough time. IQC saves the burden and inconvenience effectively. BBS administrator with IQC can update their original intentions interactively for any ambiguous questions. It also has an important feature that even a novice can use the system without any difficulty. The experiment for the performance comparison our approach with other clustering methods such as another information retrieval methods and neural networks is a promising further research area. The transformation of the question categorizing results into a tacit knowledge about VOC will be further research area.

References

[1] Abecker, A., A. Bernadi, K. Hinkelmann, O. Kuhn, and M. Sintek, "Toward a technology for organizational memories," IEEE Interactive Systems, May/June, 1998, pp.40-48.

[2] Borko, H., and M.Bernick, "Automatic Document Classification," Journal of the Association for Computing Machinery 10(1), 1983, pp.151-162.

[3] Chen, H., P. Hsu, R. Orwig, L. Hoopes, and J.F. Nunamaker, "Automatic concept classification of text form electronic meetings," Communications of the ACM 37(10), 1994, pp.56-73.

[4] Choi, H.R., K.R. Ryu, J. Kang, J.I. Shin and C.S. Lee, "An Automatic Question Routing Systems using Machine Learning," Proceedings of Conference on Korea Interactive Information Systems, 2003, pp.313-318.

[5] Davenport, T.H, P. Klahr, "Managing Customer support knowledge," California Management Review 40(3), 1998, pp.195-208.

[6] Foo, S, S.C. Hui, P.C. Leong, and S. Liu, "An integrated help desk support for customer services over the World Wide Web - a case study," Computers in Industry 41, 2000, pp.129-145.

[7] Griffiths, A., L.A.Robinson, and Willett, "Hierarchic Agglomerative Clustering Methods for Automatic Document Classification," Journal of Documentation 40(3), 1984, pp.175-205.

[8] Hamill, K.A., and A.Zamora, "The Use of Titles for Automatic Document Classification," Journal of the American Society for Information Science (JASIS) 31(6), 1980, pp.396-402.

[9] Jardine, N., and C.J.van Rijsbergen, "The Use of Hierarchic Clustering in Information Retrieval," Information Storage and Retrieval 7(5), 1971, pp.217-240.

[10] Kay, E., "http:/www.where's myorder.com," Computerword 33 (29), 1999, pp.82-83.

[11] Negash, S., T Ryan and M. Lgbria, "Quality and effectiveness in Web-based

customer systems," Information and Management, 2029, 2002, pp.1-12

[12] Raghavan, V.V., and S.K.M.Wang, "A Critical Analysis of the Vector Space Model for Information Retrieval," Journal of the American Society for Information Science (JASIS) .37(5), 1986, pp.279-287.

[13] Salton, G., "Historical Note: The Past Thirty Years in Information Retrieval," Journal of the American Society for Information Science, 38(5), 1987, pp.375-380.

[14] Salton, G., Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison Wesley, 1989.

[15] Salton, G., and McGill, M.J. Introduction of modern information retrieval. New York: McGraw-Hill, 1983.

[16] The Association of Support Professionals (ASP), "This year's 10 best Web support site," http:/www.asponline.com, 23, 2000.

저자소개

이재광 (E-mail : jklee@kpu.ac.kr)
1993. 한국과학기술원 산업공학과
1995. 한국과학기술원 경영정보공학과(석사)
2000. 한국과학기술원 경영공학(박사)
2000. 7~2002. 2 (주)OpenTide eConsulting 사업부 Senior Consultant
2002. 2~2003. 2 SK(주) 마케팅 지원본부 책임컨설턴트
관심 분야 : 전략의사결정, CRM, 지능정보시스템, e-business전략

노성호 (E-mail : shnoh@kpu.ac.kr)
1977. 고려대학교 경제학과
1979. 고려대학교 대학원 경제학과(석사)
1984. 국립 Aix-Marseille 산업경제학(박사)
1990~1992. SOC 투자 기획단 전문위원(청와대)
1992~1993. 지역균형발전 기획단(청와대)
1997~1997. 산업연구원(연구기획조정본부장, 선임연구위원)
1997. 한국산업단지공단 정보센터소장
2000~2002. 전국산업단지기업 e-Biz화 및 디지털화 사업책임자(산업자원부)
1998~현재 산업정책평가위원회(산업자원부)
2001~산업기술개발 및 기반조성사업 평가위원(산업자원부)
관심 분야 : e-business 정책 및 전략

류옥현 (E-mail : ok-ryou@kpu.ac.kr)
1991. 서울대학교 산업공학과
1993. 한국과학기술원 산업공학과(석사)
2001. University of New Hampshire Engineering: Systems Design(박사)
1993~1998. 대우자동차 기술연구소
2001~2002. 포스데이타 e-Biz 사업부 차장
관심 분야 : 기업간 전자상거래, CAD/CAM, PDM