(1) Name of the Workshop: Workshop on Artificial Intelligence

(2) Title of the Paper: A New Method for Fuzzy Information Retrieval Based on Document Terms Reweighting Techniques

(3) Abstract: In this paper, we propose a new method for fuzzy information retrieval based on terms reweighting techniques to modify the weights of terms in document descriptor vectors based on the user's relevance feedback. After modifying the weights of terms in document descriptor vectors, the degrees of satisfaction of relevant documents with respect to the user's query will increase, and the degrees of satisfaction of irrelevant documents with respect to the user's query will decrease. The modified document descriptor vectors then can be used as personal profiles for future query processing. The proposed method can make fuzzy information retrieval systems more flexible and more intelligent to deal with documents retrieval. It can increase the retrieval effectiveness of the fuzzy information retrieval systems for document retrieval.

(4) Authors: Yih-Jen Horng (Ph.D. Student),
Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, R. O. C.
E-mail: yjhorng@cis.nctu.edu.tw
Tel: (02) 27333141 ext. 7212

Professor Shyi-Ming Chen,
Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R. O. C.
E-mail: smchen@et.ntust.edu.tw
Tel: (02) 27376417

Professor Chia-Hoang Lee,
Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, R. O. C.
E-mail: chl@cis.nctu.edu.tw
Tel: (03) 5712121 ext. 56619

(5) Contact Author: Professor Shyi-Ming Chen,
Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R. O. C.
E-mail: smchen@et.ntust.edu.tw
Tel: (02) 27376417

(6) Keywords: Document Descriptor Vectors, Fuzzy Information Retrieval, Personal Profile, Relevance Feedback, Terms Reweighting.

A New Method for Fuzzy Information Retrieval Based on Document Terms Reweighting Techniques

Yih-Jen Horng[*], Shyi-Ming Chen[**], and Chia-Hoang Lee[*]

***Department of Computer and Information Science**
National Chiao Tung University
Hsinchu, Taiwan, R. O. C.

**** Department of Computer Science and Information Engineering**
National Taiwan University of Science and Technology
Taipei, Taiwan, R. O. C.

Abstract

In this paper, we propose a new method for fuzzy information retrieval based on terms reweighting techniques to modify the weights of terms in document descriptor vectors based on the user's relevance feedback. After modifying the weights of terms in document descriptor vectors, the degrees of satisfaction of relevant documents with respect to the user's query will increase, and the degrees of satisfaction of irrelevant documents with respect to the user's query will decrease. The modified document descriptor vectors then can be used as personal profiles for future query processing. The proposed method can make fuzzy information retrieval systems more flexible and more intelligent to deal with documents retrieval. It can increase the retrieval effectiveness of the fuzzy information retrieval systems for document retrieval.

Keywords: Document Descriptor Vectors, Fuzzy Information Retrieval, Personal Profile, Relevance Feedback, Terms Reweighting.

1. Introduction

Automatic terms weighting is an important aspect of modern information retrieval systems [1]. Since different index terms have different degrees of importance in a document, an importance indicator (i.e., the term weight) is associated with each index term. Two main components that affect the importance of a index term in a document are the term frequency factor (*tf*) and the inverse document frequency factor (*idf*) [5], [17]. The term frequency factor indicates that if a term occurs frequently in a document, then this term should be important for this document. On the other hand, the inverse document frequency factor indicates that if a term occurs in most of the

collected documents, then its representative importance for any documents should be low. However, the terms weighting methods based on these two statistic factors may be not suitable enough from the human's point of view. Therefore, the degrees of similarity between relevant documents and the user's query may not be large enough. As a result, the retrieval effectiveness of the information retrieval system is not good enough. In order to increase the retrieval effectiveness, some "terms weighting" methods have been proposed [9], [17]. In [9], Jung et al. proposed a terms weighting scheme which not only considers occurrence terms, but also absence terms in finding the degrees of similarity among document descriptor vectors, where the absence terms are negatively weighted. In [17], Singhal et al. proposed a document length normalization method in which the term weights in the document descriptor vectors are normalized according to the document length, and the documents with different lengths can be fairly retrieved.

Another approach to increase the retrieval effectiveness is by modifying the user's query [3], [7], [10]. In [3], Chan et al. proposed a query expansion method which apply clustering techniques to the initial search results to provide concept-based browsing and help the user to reduce the browsing labor. In [7], we proposed a method for query expansion based on the cluster centers of the document clusters. In [10], Kim et al. proposed a query term expansion and reweighting method which consider the term co-occurrence within the feedbacked documents. Among these methods, the most used one is the "relevance feedback" method which requires the users to provided the relevance judgment of the retrieved documents. It modifies the user's query based on the set of relevant documents and the set of irrelevant documents among the retrieved documents. Although most of the relevance feedback methods are used for user's query expansion, the effect of the relevance feedback methods is only restricted to deal with the current query processing and it can't to deal with the following queries. If the user's relevance feedback can be recorded and used for the following queries, then the users of the information retrieval systems do not need to perform the process of relevance feedback in the future.

In this paper, we propose a new method for fuzzy information retrieval based on terms reweighting techniques to modify the weights of terms in document descriptor vectors based on the user's relevance feedback. After modifying the weights of terms in document descriptor vectors, the degrees of satisfaction of relevant documents with respect to the user's query will increase, and the degrees of satisfaction of irrelevant

documents with respect to the user's query will decrease. The modified document descriptor vectors then can be used as personal profiles for future query processing. The proposed method can make fuzzy information retrieval systems [12], [13] more flexible and more intelligent to deal with documents retrieval. It can increase the retrieval effectiveness of the fuzzy information retrieval systems for document retrieval.

The rest of this paper is organized as follows. In Section 2, we present an automatic terms weighting method and a query processing method for document retrieval. In Section 3, we present a method to derive modified document descriptor vectors based on the user's relevance feedback. In Section 4, we present the experimental results. The conclusions are discussed in Section 5.

2. Automatic Terms Weighting and User's Query Processing

In traditional information retrieval systems, the contents of documents are typically represented by some index terms extracted from the texts of the collected documents [15]. The most direct approach is to use all words appearing in the contents of the collected documents as index terms. However, since each document contains a large amount of words, these documents should be preprocessed to reduce the set of words into a manageable size for processing. The selected documents are preprocessed in two steps. Firstly, the words appearing with high frequencies in all documents are eliminated [14]. Then, the word extractor stems each remaining word to its "root form" [6]. The collection of these root-formatted words forms a set of index terms T for the document set. The formula for calculating the weight of a term in a document is based on the normalized *TF* x *IDF* (i.e., Term Frequency multiply Inverse Document Frequency) weighting method. The weight "*w_term_document*(t,d_i)" of term t in document d_i is calculated as follows [8]:

$$w\_term\_document(t,d_i)=\frac{(0.5+0.5\frac{tf_{it}}{\underset{k=1,2,\ldots,L}{Max}\ tf_{ik}})\log\frac{N}{df_t}}{\underset{j=1,2,\ldots,L}{Max}\left\{(0.5+0.5\frac{tf_{ij}}{\underset{k=1,2,\ldots,L}{Max}\ tf_{ik}})\log\frac{N}{df_j}\right\}}, \quad (1)$$

where tf_{it} denotes the frequency of term t appearing in document d_i, df_t denotes the number of documents containing term t, L denotes the number of terms contained in document d_i, and N denotes the number of collected documents. The larger the value

of *w_term_document*(t, d_i), the more important the term t to document d_i. From formula (1), we can see that the value of *w_term_document*(t, d_i) is between zero and one.

After the weight of each term in each document has been calculated, we can represent each document d_i as a document descriptor vector shown as follows:

$$\overline{d_i} = <w_{i1}, w_{i2}, \ldots, w_{is}>, \tag{2}$$

where w_{ij} denotes the weight of term t_j in document d_i, $0 \le w_{ij} \le 1$, $1 \le j \le s$, and s denotes the number of terms in the set of index terms.

Assume that the user's query q is represented by a query vector $\overline{q}$ shown as follows:

$$\overline{q} = < w_{q1}, w_{q2}, \cdots, w_{qs} >,$$

where $0 \le w_{qi} \le 1$ and $1 \le i \le s$. Then, the degree of satisfaction $DS(d_i)$ of document d_i with respect to the user's query can be calculated as follows [4]:

$$DS(d_i) = \frac{\sum_{j=1,2,\ldots,\ s} T(w_{qj}, w_{ij})}{s}, \tag{3}$$

where s is the number of terms in the set of index terms, T is a similarity function [4] to calculate the degree of similarity between two real values between zero and one,

$$T(w_{qj}, w_{ij}) = 1 - |w_{qj} - w_{ij}|, \tag{4}$$

where $0 \le w_{qj} \le 1$, $0 \le w_{ij} \le 1$, and $1 \le j \le s$. After the degree of satisfaction $DS(d_i)$ of each document d_i with respect to the user's query is obtained, we normalized the value of $DS(d_i)$ by dividing it with the maximum value among the values of $DS(d_1)$, $DS(d_2)$, …, and $DS(d_N)$, where N is the number of collected documents. The user can set a query threshold value α, where $\alpha \in [0, 1]$. The documents are retrieved only when their degrees of satisfaction with respect to the user's query q are larger than or equal to α, where $\alpha \in [0, 1]$.

3. A Method for Terms Reweighting for Fuzzy Information Retrieval

In this section, we propose a new method for terms reweighting for fuzzy information retrieval. The goal of the proposed method is to reduce the degrees of satisfaction of irrelevant documents and increase the degrees of satisfaction of relevant documents with respect to the user's query according to the user's "relevance feedback".

First, the documents and the user's query can be represented as points in a vector space as shown in Fig. 1, respectively, where each "○" means a relevant document with respect to the user's query, each "×" means a irrelevant document with respect to the user's query and "●" means the user's query.

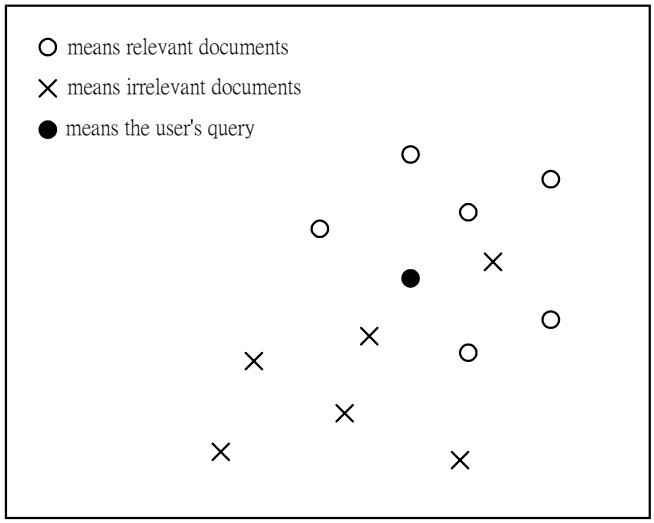

Fig. 1. Each document and the user's query represented as points in a vector space.

An intuitive idea of reducing the degrees of satisfaction of irrelevant documents and increasing the degrees of satisfaction of relevant documents with respect to the user's query, respectively, is to move each relevant document closer to the user's query q and move each irrelevant document away from the user's query q in the vector space as shown in Fig. 2.

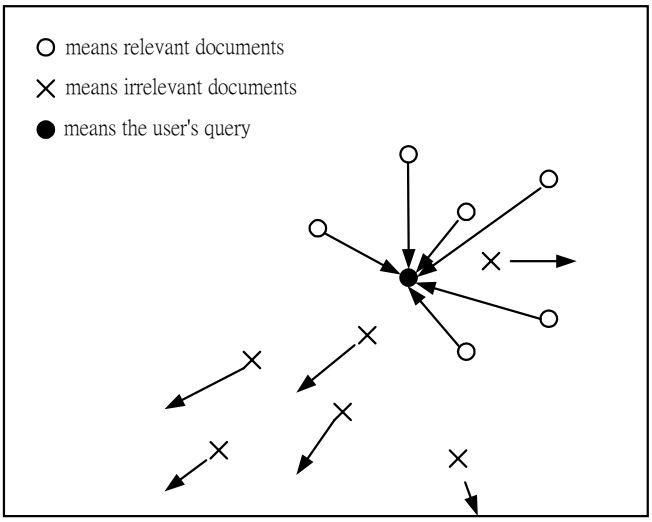

Fig. 2. Relevant documents move toward the user's query and irrelevant documents move away from the user's query in the vector space.

However, since the amount of modification for each retrieved document will be recorded for the future use, if each document has its own amount of modification, then there are lots of data have to be stored. Therefore, in this paper, we let each document move toward the same direction with the same distance. That is, we let each document has a uniformed movement. This uniformed movement of each document is transformed into a vector, which is defined as the document modification vector Δ shown as follows:

$$\Delta = <\delta_1, \delta_2, \cdots, \delta_s>,$$

where δ_i indicates the amount of modification to the ith term of each document, $1 \le i \le s$, and s is the number of terms extracted from the collected documents. When the document modification vector Δ is used to modify each document descriptor vector $\overline{d_i}$ to derive a modified document descriptor vector $\overline{d_i^*}$, where $\overline{d_i^*} = \overline{d_i} + \Delta$, most of the relevant documents will move toward the user's query q to increase their degrees of satisfaction with respect to the user's query q, and most of the irrelevant documents will move away from the user's query q to decrease their degrees of satisfaction with respect to the user's query q, as shown in Fig. 3. Therefore, when all the retrieved documents are resorted according to their new degrees of satisfaction with respect to the user's query, most of the relevant documents will be listed in front of the irrelevant documents. It will be useful to the users when browsing the retrieved documents since most relevant documents are listed in the front part of the document

list and the user can ignore the irrelevant documents by a query threshold value α, where $\alpha \in [0, 1]$.

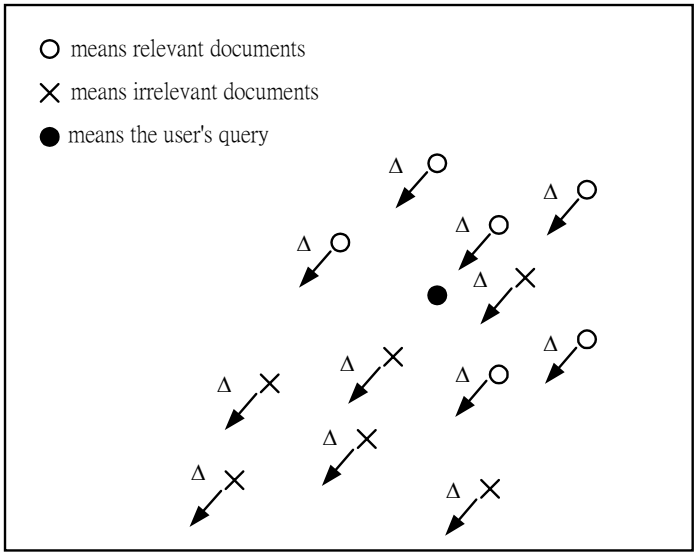

Fig. 3. Each document in the vector space is modified by a document modification vector Δ.

However, there are many candidates for the document modification vector Δ. For example, from Fig. 4 and Fig. 5, we can see that when the document modification vector Δ_1 and the document modification vector Δ_2 are applied to modify the document descriptor vector of each document, both of them will cause the most of the relevant documents to move toward the user's query q and cause most of the irrelevant documents to move away from the user's query q.

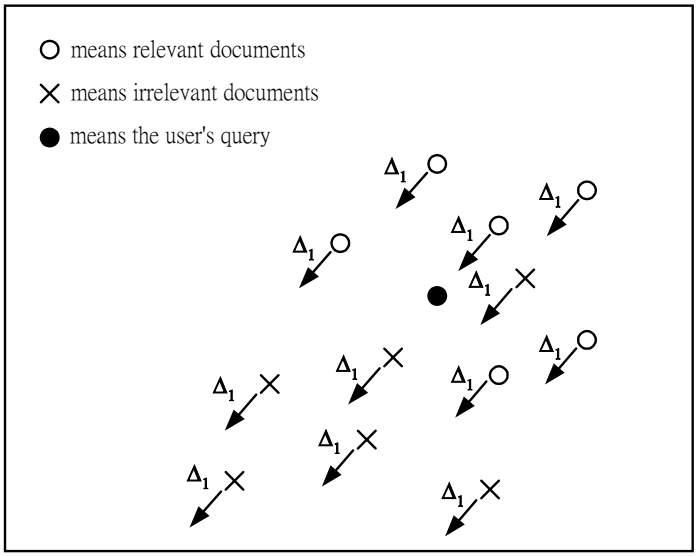

Fig. 4. Each document in the vector space is modified by the document modification vector Δ_1.

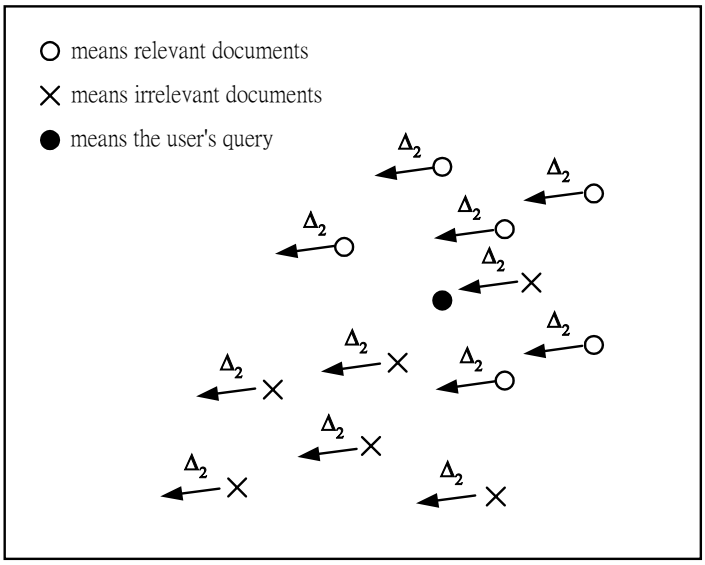

Fig. 5. Each document in the vector space is modified by the document modification vector Δ_2.

In order to choose a better document modification vector, we use the "Relevant Document Ranking Score" (*RDRS*) to judge the performance of the document modification vector based on the resorted document list, where

$$RDRS = \sum_{i=1,2,\ldots,m \text{ and } d_i \text{ is relevant to the user's query}} \frac{1}{Rank_{d_i}}, \tag{5}$$

where m is the number of retrieved documents with respect to the user's query q and $Rank_{d_i}$ denotes the rank of document d_i in the resorted document list. The maximum value of *RDRS* occurs when all relevant documents are ranked before all irrelevant documents with respect to the user's query. On the other hand, the minimum value of *RDRS* occurs when all irrelevant documents are ranked before all relevant documents with respect to the user's query. Moreover, the more the relevant documents are listed before irrelevant documents, the larger the value of *RDRS*. For example, assume that there are five documents d_1, d_2, d_3, d_4, d_5, retrieved from a fuzzy information retrieval system with respect to the user's query q. Furthermore, assume that the documents d_1, d_2 and d_3 judged by the user as relevant documents and the documents d_4 and d_5 judged by the user as irrelevant documents. Assume that these five documents are ordered according to their degrees of satisfaction with respect to the user's query q shown as follows:

$$d_1 > d_4 > d_5 > d_3 > d_2,$$

then the value of *RDRS* of the retrieved documents is $\frac{1}{1}+\frac{1}{4}+\frac{1}{5}=1.45$. Assume that a document modification vector Δ_1 is used, and the relevant documents are moved closer to the user's query q than the irrelevant documents, where the retrieved documents are reordered according to their new degrees of satisfaction with respect to the user's query q shown as follows:

$$d_1 > d_3 > d_4 > d_2 > d_5,$$

then the values of *RDRS* of the retrieved documents became $\frac{1}{1}+\frac{1}{2}+\frac{1}{4}=1.75$. Moreover, assume that another document modification vector Δ_2 is used and the relevant documents are moved closer to the user's query q than the irrelevant documents, where the retrieved documents are reordered according to their new degrees of satisfaction with respect to the user's query q shown as follows:

$$d_1 > d_3 > d_2 > d_4 > d_5,$$

then the value of *RDRS* of the retrieved documents became $\frac{1}{1}+\frac{1}{2}+\frac{1}{3}=1.83$. Therefore, based on the above *RDRS* values, we can see that the document modification vector Δ_2 is better than the document modification vector Δ_1.

The purpose of the proposed method is to provide a method to derive a document modification vector Δ which can make the value of *RDRS* of the retrieved documents as large as possible. However, since the degree of satisfaction of each document d_i with respect to the user's query q is based on the relative position of the document descriptor vector $\overline{d_i}$ and the user's query vector $\overline{q}$ in the vector space, the effect of modifying the document descriptor vector $\overline{d_i}$ of each document d_i by a document modification vector Δ as shown in Fig. 3 is equal to the effect of modifying the user's query vector $\overline{q}$ by the inverse vector "-Δ" of the document modification vector Δ as shown in Fig. 6. Since considering the modification of only one point (i.e., the user's query vector $\overline{q}$) in the vector space seems to require less effort than considering the modification of many points (i.e., all document descriptor vectors of retrieved documents) in the vector space at the same time, the method of deriving a document modification vector can start by deriving a "virtual query modification vector" which virtually move the user's query vector $\overline{q}$ to relevant document descriptor vectors as

close as possible and move as the user's query vector to irrelevant document descriptor vectors in the vector space as far away as possible. The difference between the virtually new position and the origin position of the user's query vector $\bar{q}$ in the vector space is then viewed as the virtual query modification vector. Then, the inverse vector of the virtual query modification vector is used as the document modification vector.

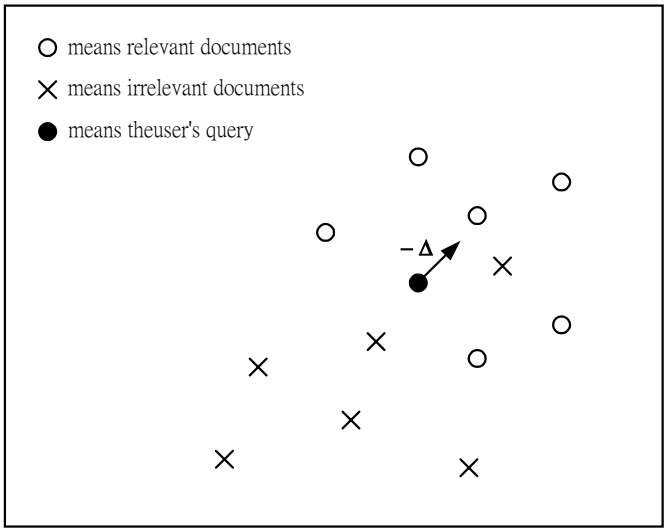

Fig. 6. The user's query vector is modified by an inverse document modification vector -Δ.

In the following, we propose an algorithm for document terms reweighting in document descriptor vectors.

Document Terms Reweighting Algorithm:

Step 1: Divide the retrieved documents into two clusters, where one cluster contains relevant documents and the other contains irrelevant documents.

Step 2: Let the new virtual user's query vector $\overline{vq}$ equal to the cluster center of the cluster containing relevant documents in the vector space.

Step 3: Calculate the degree of satisfaction $DS(d_i)$ of each document d_i with respect to the new virtual user's query vector $\overline{vq}$ by formula (3).

Step 4: Sort the retrieved documents according to the new value of $DS(d_i)$ of each document d_i.

Step 5: **If** all relevant documents are listed before all irrelevant documents
then Stop

else Find the irrelevant document d_{ir} which has the largest value of $DS(d_i)$ and find the first relevant document d_r next to d_{ir} in the ordered document list.

Step 6: Move the new virtual user's query vector $\overline{vq}$ across the middle line between d_r and d_{ir} in the vector space.

Step 7: Calculate the degree of satisfaction $DS(d_i)$ of each document d_i with respect to the new virtual user's query vector $\overline{vq}$ using formula (3).

Step 8: Sort the retrieved documents according to the new $DS(d_i)$ of each document d_i.

Step 9: **If** the position of d_{ir} is moved backward in the document list

then go to Step 5

else restore the former document list and find the relevant document next to d_r in the ordered document list;

if no such documents exist **then Stop**

else use it as d_r and go to Step 6.

Step 10: Calculate the difference between the new virtual user's query vector $\overline{vq}$ and the original user's query vector $\overline{q}$ and use it as the virtual query modification vector.

Step 11: Use the inverse vector of the virtual query modification vector as the document descriptor modification vector Δ.

Step 12: Use the derived document descriptor modification vector Δ to reweight document terms in the set of D document descriptor vectors of the retrieved documents, where $D = \{\overline{d_1}, \overline{d_2}, \cdots, \overline{d_m}\}$.

for $i = 1$ **to** m **do**

let $\overline{d_i} = \overline{d_i} + \Delta$

end.

The proposed document terms reweighting algorithm starts by dividing the retrieved documents into two clusters (i.e., two classes), where one cluster contains relevant documents and the other contains irrelevant documents. The idea behind this is based on the assumption that most of the relevant documents should be closer to the

cluster center of the cluster containing the relevant documents than the irrelevant documents. Therefore, the cluster center of the cluster containing relevant documents should be a good start point as the new virtual position of the user's query vector $\bar{q}$.

After virtually moving the user's query vector $\bar{q}$ to the cluster center of the cluster containing the relevant documents in the vector space to derive a new virtual user's query vector $\overline{vq}$ and calculating the degree of satisfaction $DS(d_i)$ of each document d_i with respect to the new virtual user's query, the system sorts the retrieved documents according to the new value of $DS(d_i)$ of each document d_i. However, since the two clusters are often overlapped, some irrelevant documents may be closer to the cluster center of the cluster containing relevant documents than some relevant documents as shown in Fig. 7. That is, some irrelevant documents may be listed before some relevant documents in the ordered document list. Therefore, further adjustment of the position of the new virtual user's query vector $\overline{vq}$ in the vector space is required. We do this by finding an irrelevant document d_{ir} with the highest rank comparing to other irrelevant documents and by finding the first relevant document d_r ranked after d_{ir} in the ordered document list. For example, assume that there are seven retrieved documents d_1, d_2, …, d_7 and assume that their order according to their degrees of satisfaction with respect to the new virtual user's query from high to low is $d_1 > d_2 > d_3 > d_4 > d_5 > d_6 > d_7$ as shown in Table 1. Then, document d_3 is used as d_{ir} and document d_5 is used as d_r.

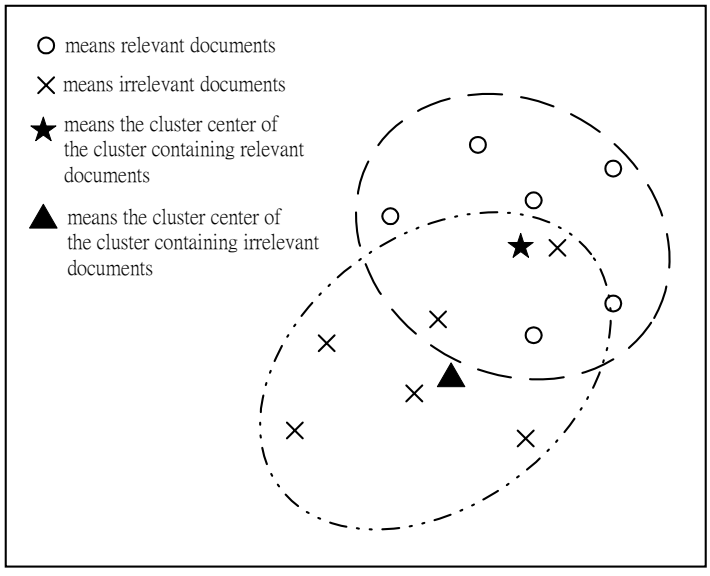

Fig. 7. The relevant cluster and the irrelevant cluster are often overlapped.

Table 1. An Ordered Document List

Ordered Document List	Relevant or Irrelevant	
d_1	Relevant	
d_2	Relevant	
d_3	Irrelevant	← d_{ir}
d_4	Irrelevant	
d_5	Relevant	← d_r
d_6	Relevant	
d_7	Irrelevant	

Then, we try to adjust the position of the new virtual user's query vector $\overline{vq}$ to make it more close to the document descriptor vector $\overline{d_r}$ than to the document descriptor vector $\overline{d_{ir}}$ in the vector space. It is achieved by deriving the middle line between $\overline{d_r}$ and $\overline{d_{ir}}$ in the vector space and move the new virtual user's query vector $\overline{vq}$ across this middle line as shown in Fig. 8. Therefore, after calculating the degree of satisfaction $DS(d_i)$ of each document d_i with respect to the new virtual user's query and resorting the retrieved documents according to the new value of $DS(d_i)$ of each document d_i, we can see that d_r can get a higher rank than d_{ir}.

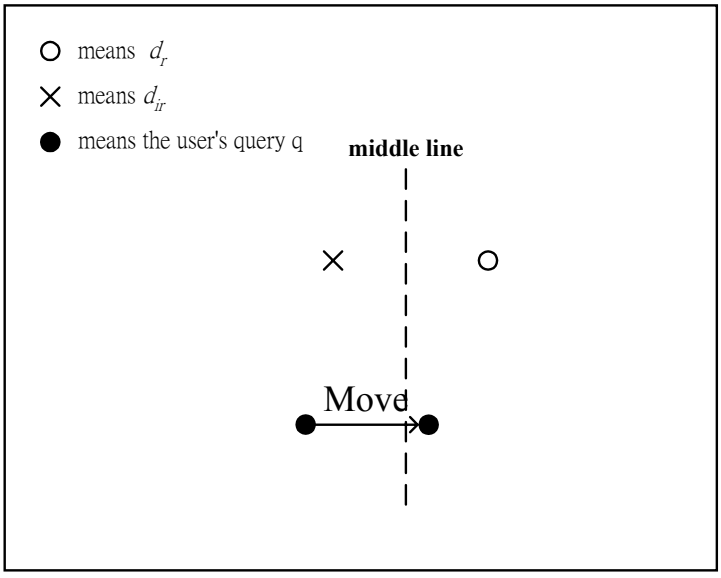

Fig. 8. The new virtual user's query vector is derived by virtually move across the middle line between d_r and d_{ir}.

However, sometimes virtually move the new virtual user's query vector $\overline{vq}$ to across the middle line between d_r and d_{ir} may cause other relevant documents to decrease their degrees of satisfaction with respect to the new virtual user's query. Therefore, although the degrees of satisfaction of d_r with respect to the new virtual user's query vector $\overline{vq}$ is larger than the one of d_{ir}, the rank of d_{ir} may still move forward in the document list when all retrieved documents are reordered according to their new degrees of satisfaction with respect to the new virtual user's query. When this happens, the former ordered document list should be restored, and the relevant document next to d_r in the ordered document list should be used as d_r. The operation repeats again until no further appropriate d_r is found. Then, the difference between the new virtual user's query vector $\overline{vq}$ and the original user's query vector $\overline{q}$ is calculated and we use the difference as the virtual query modification vector. The inverse vector of the virtual query modification vector is used as the document descriptor modification vector. Finally, the document descriptor modification vector is used to reweight document terms in document descriptor vectors of the retrieved documents.

After reweighting the terms in the document descriptor vectors by the proposed algorithm, most of the relevant documents will be listed in front of the irrelevant documents. For example, assume that there are three retrieved documents d_1, d_2 and d_3 with respected to the user's query q. The original document descriptor vectors of the three retrieved documents are:

$$\overline{d_1} = < 0.4, 0.6, 0.1, 0 >,$$

$$\overline{d_2} = < 0.7, 0.6, 0, 0.2 >,$$

$$\overline{d_3} = < 0.9, 1, 0.1, 0 >,$$

and the user's query descriptor vector $\overline{q}$ is

$$\overline{q} = < 0.5, 0.8, 0, 0 > .$$

Based on formula (3), we can get

$$DS(d_1) = \frac{(1-|0.4-0.5|)+(1-|0.6-0.8|)+(1-|0.1-0|)+(1-|0-0|)}{4}$$

$$= \frac{0.36}{4}$$

$$= 0.9,$$

$$DS(d_2) = \frac{(1-|0.7-0.5|)+(1-|0.6-0.8|)+(1-|0-0|)+(1-|0.2-0|)}{4}$$

$$= \frac{0.34}{4}$$

$$= 0.85,$$

$$DS(d_3) = \frac{(1-|0.9-0.5|)+(1-|1-0.8|)+(1-|0.1-0|)+(1-|0-0|)}{4}$$

$$= \frac{0.33}{4}$$

$$= 0.825.$$

Therefore, the order of the three retrieved documents according to their degrees of satisfaction with respect to the user's query q from high to low is $d_1 > d_2 > d_3$. Assume that document d_1 and document d_2 are judged by the user as irrelevant documents and document d_3 is judged as a relevant document. Then, according to the proposed document terms reweighting algorithm, we can get a document descriptor modification vector Δ shown as follows:

$$\Delta = <-0.2, -0.1, 0, 0>.$$

Therefore, the modified document descriptor vectors of the three retrieved documents are as follows:

$$\overline{d_1} = < 0.2, 0.5, 0.1, 0 >,$$

$$\overline{d_2} = < 0.5, 0.5, 0, 0.2 >,$$

$$\overline{d_3} = < 0.7, 0.9, 0.1, 0 > .$$

Based on formula (3), we can get

$$DS(d_1) = \frac{(1-|0.2-0.5|)+(1-|0.5-0.8|)+(1-|0.1-0|)+(1-|0-0|)}{4}$$

$$= \frac{0.33}{4}$$

$$= 0.825,$$

$$DS(d_2) = \frac{(1-|0.5-0.5|)+(1-|0.5-0.8|)+(1-|0-0|)+(1-|0.2-0|)}{4}$$

$$= \frac{0.35}{4}$$

$$= 0.875,$$

$$DS(d_3) = \frac{(1-|0.7-0.5|)+(1-|0.9-0.8|)+(1-|0.1-0|)+(1-|0-0|)}{4}$$

$$= \frac{0.36}{4}$$

$$= 0.9.$$

Therefore, the order of the three retrieved documents according to their degrees of satisfaction with respect to the user's query q from high to low is $d_3 > d_2 > d_1$. We can see that the only one relevant document (i.e., document d_3) is moved from the last place of the original document list to the first place of the new document list.

However, since the document descriptor modification vector Δ is obtained based on the user's relevance feedback with respect to a specific query q, the effect of utilizing the document descriptor modification vector Δ will not take place when the user submits another query. Instead, the document descriptor modification vector Δ is recorded as a personal profile with respect to the specific user's query q. When the user submit the same query q in the future, the information retrieval system will find out the document descriptor modification vector Δ with respect to the specific user's query q and use it to modify the document descriptor vectors of the retrieved documents for document retrieval.

4. Experimental Results

We have implemented the proposed terms reweighting algorithm for document retrieval on a Pentium 4 PC using Delphi Version 5.0 [5]. We choose 247 research reports [20] as the set of documents for clustering, which are a subset of collection of

the research reports of the National Science Council (NSC), Taiwan, Republic of China. Each report consists of several parts, including a report ID, a title, researchers' names, a Chinese abstract, an English abstract, …, etc. Since the proposed method intends to deal with English documents, the system grabs the English abstracts of the reports to represent the contents of the documents. The automatic document indexing method described in Section 2 is used to represent the contents of the documents by the index terms extracted from the set of documents containing 247 reports [20]. The documents are then represented by document descriptor vectors which are used for further user's query processing.

Ten queries as shown in the first column of Table 2 are submitted to the information retrieval system and assume that the query threshold value α given by the user is 0.4. Then, a set of documents are retrieved and the retrieved documents are sorted according to their degrees of satisfaction with respect to the queries. The value of *RDRS* of each set of retrieved documents are calculated and recorded as shown in the second column of Table 2. Then, for each query, the retrieved documents are judged by the user as relevant or irrelevant. Based on the user's relevance feedback, the system modifies the document descriptor vectors of the retrieved documents by using the proposed document terms reweighting algorithm presented in Section 3 and calculates their new degrees of satisfaction with respect to the query. The retrieved documents for each query are resorted according to their new degrees of satisfaction with respect to the query. The value of *RDRS* of each set of retrieved documents are calculated and recorded again as shown in the third column of Table 2. From Table 2, we can see that by modifying the document descriptor vectors using the proposed document terms reweighting algorithm, most of the *RDRS* values of the retrieved documents of the 10 queries are improved except the ones of the second query and the fifth query. It is because the *RDRS* values of the retrieved documents using the original document descriptor vectors with respect to the second query and the fifth query, respectively, are large than 3, which means that most of the relevant documents are listed in the front part of the document list, and it is no much room for further improvement.

Table 2. *RDRS* Values of the Retrieved Documents

Queries	RDRS Values of the Retrieved Documents Before Applying the Proposed Document Terms Reweighting Algorithm	RDRS Values of the Retrieved Documents After Applying the Proposed Document Terms Reweighting Algorithm
q_1 = natural language processing (The Weights of the Terms "natural", "language" and "processing" are 0.8, 0.8 and 0.7, Respectively)	2.246	2.659
q_2 = fuzzy set (The Weights of the Terms "fuzzy" and "set" are 0.8 and 0.8, Respectively)	3.010	3.010
q_3 = heterogeneous database (The Weights of the Terms "heterogeneous" and "database" are 0.9 and 0.8, Respectively)	3.018	3.060
q_4 = database management (The Weights of the Terms "database" and "management" are 0.8 and 0.7, Respectively)	1.136	2.599
q_5 = expert system (The Weights of the Terms "expert" and "system" are 0.9 and 0.7, Respectively)	3.173	3.173
q_6 = image processing (The Weights of the Terms "image" and "processing" are 0.8 and 0.7, Respectively)	1.905	2.368
q_7 = machine learning The Weights of the Terms "machine" and "learning" are 0.8 and 0.8, Respectively)	2.413	3.131
q_8 = object oriented database (The Weights of the Terms "object", "oriented" and "database" are 0.9, 0.9 and 0.8, Respectively)	2.782	3.030
q_9 = image restoration (The Weights of the Terms "image" and "restoration" are 0.8 and 0.8, Respectively)	3.111	3.141
q_{10} = multimedia database (The Weights of the Terms "multimedia" and "database" are 0.9 and 0.8, Respectively)	2.050	3.112

Since a large *RDRS* value of the retrieved document indicates that most of the relevant documents are in the front part of the ordered document list, it will result in a better retrieval effectiveness if the user only concerns the top ranked documents of the ordered document list. For the 10 queries submitted to the information retrieval system, assume that the user concerns only the top 10 documents of each set of the retrieved documents with respect to the 10 queries. A comparison of the recall rates of the top 10 retrieved documents with respect to each query using the original document descriptor vectors with that using the modified document descriptor vectors is shown in Fig. 9. A comparison of the precision rates of the top 10 retrieved documents with respect to each query using the original document descriptor vectors with that using the modified document descriptor vectors is shown in Fig. 9. From Fig. 9 and Fig. 10, we can see that when the system uses the modified document descriptor vectors, it can

get a higher or the same precision rate and recall rate regarding the top 10 retrieved documents than those using the original document descriptor vectors.

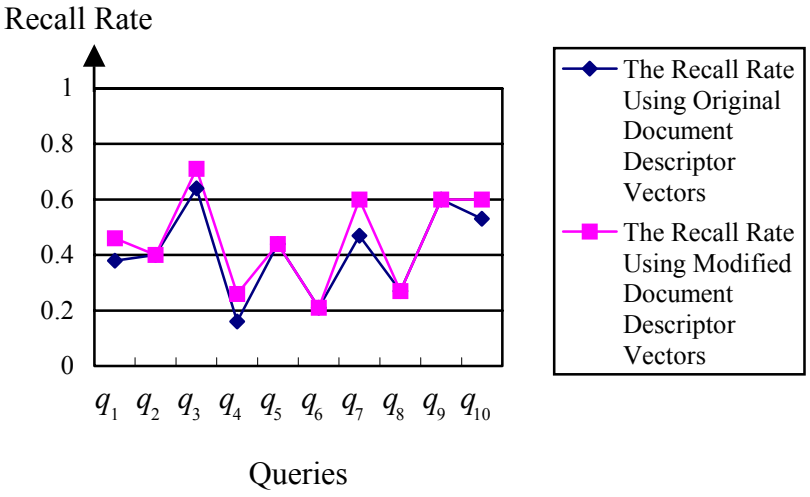

Fig. 9. The recall rate of the top 10 documents with respect to each user's query.

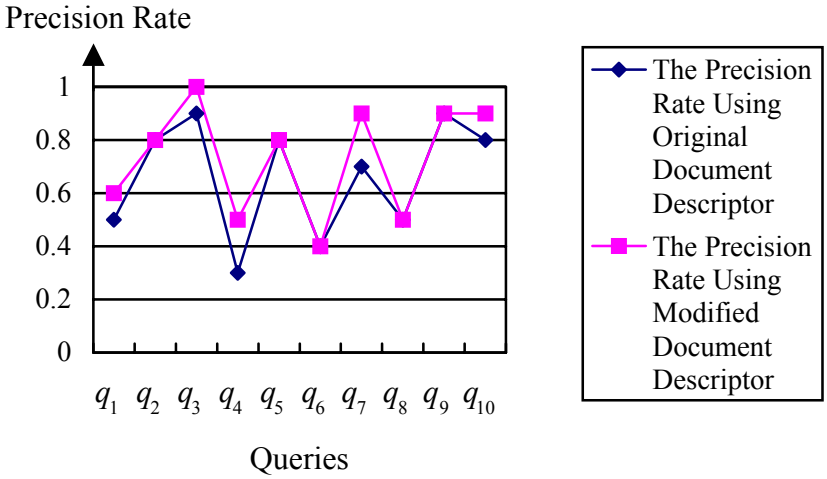

Fig. 10. The precision rate of the top 10 documents with respect to each user's query.

5. Conclusions

In this paper, we have proposed a new method for fuzzy information retrieval based on terms reweighting techniques to modify the weights of terms in document descriptor vectors based on the user's relevance feedback. After modifying the weights of terms in document descriptor vectors, the degrees of satisfaction of relevant documents with respect to the user's query will increase, and the degrees of satisfaction of irrelevant documents with respect to the user's query will decrease. The modified document descriptor vectors then can be used as personal profiles for

future query processing. The proposed method can make fuzzy information retrieval systems more flexible and more intelligent to deal with documents retrieval. It can increase the retrieval effectiveness of the fuzzy information retrieval systems for document retrieval.

Acknowledgements

This work was supported in part by the National Science Council, Republic of China, under Grant NSC-91-2213-E-011-052.

References

[1] G. Bordogna and G. Pasi, "A user-adaptive neural network supporting a rule-based relevance feedback," *Fuzzy Sets and Systems*, vol. 82, no. 2, pp. 201-211, 1996.

[2] Chris Buckley, "The importance of proper weighting methods," in *Human Language Technology*, edited by M. Bates. CA: Morgan Kaufman, pp. 349-352, 1993.

[3] C. H. Chan and C. C. Hsu, "Enabling concept-based relevance feedback for information retrieval on the WWW," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 4, pp. 595-609, 1999.

[4] S. M. Chen and J. Y. Wang, "Document retrieval using knowledge-based fuzzy information retrieval techniques," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, vol. 25, no. 6, pp. 793-803, 1995.

[5] G. Cornell and T. Strain, *Delphi Nuts & Bolts for Experienced Programmers*. CA: McGraw-Hill, 1995.

[6] W. B. Frakes, "Stemming algorithms," in *Information Retrieval: Data Structure & Algorithms*, edited by W. B. Frakes and R. Baeza-Yates. New Jersey: Prentice Hall, pp. 131-160, 1992.

[7] Y. J. Horng, S. M. Chen, and C. H. Lee, "Fuzzy information retrieval using fuzzy hierarchical clustering and fuzzy inference techniques," *Proceedings of the 13th International Conference on Information Management*, Taipei, Taiwan, Republic of China, vol. 1, pp. 215-222, 2002.

[8] Y. J. Horng, S. M. Chen, and C. H. Lee, "Automatically constructing multi-relationship fuzzy concept networks in fuzzy information retrieval systems," *Proceedings of the 10th IEEE International Conference on Fuzzy*

Systems, Melbourne, Australia, vol. 2, 2001.

[9] Y. Jung, H. Park, and D. Du, "An effective term weighting scheme for information retrieval," *Computer Science Technical Report TR008*, Department of Computer Science, University of Minnesota, Minneapolis, Minnesota, pp. 1-15, 2000.

[10] B. M. Kim, J. Y. Kim, and J. Kim "Query term expansion and reweighting using term co-occurrence similarity and fuzzy inference," *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vancouver, Canada, vol. 2, pp. 715-720, 2001.

[11] H. M. Lee, S. K. Lin, and C. W. Huang, "Interactive query expansion based on fuzzy association thesaurus for web information retrieval," *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, vol. 2, 2001.

[12] S. Miyamoto, "Information retrieval based on fuzzy associations," *Fuzzy Sets and Systems*, vol. 38, pp. 191-205, 1990.

[13] S. Miyamoto, *Fuzzy Sets in Information Retrieval and Cluster Analysis*. Netherlands: Kluwer Academic Publishers, 1990.

[14] G. Salton, *The SMART Retrieval System: Experiments in Automatic Document Processing*. New Jersey: Prentice Hall, 1971.

[15] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513-523, 1988.

[16] G. Salton and M. J. Mcgill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.

[17] A. Singhal, C. Buckley, and M. Mitra, "Pivoted document length normalization," *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, pp. 21-29, 1996.

[18] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

[19] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*. Boston: Kluwer Academic Publishers, 1991.

[20] A Subset of the Collection of the Research Reports of the National Science Council, Taiwan, R. O. C. *http://fuzzylab.et.ntust.edu.tw/NSC_Report_Database/247documents.html* (Data Source: *http://sticnet.stic.gov.tw/sticweb/html/index.htm*).