Document Retrieval Using Knowledge-Based Fuzzy Information Retrieval Techniques

Shyi-Ming Chen, *Member, IEEE,* and Jeng-Yih Wang

***Abstract*— A knowledge-based approach for fuzzy information retrieval is proposed, where interval queries and weighted-interval queries are allowed for document retrieval. In this paper, knowledge is represented by a concept matrix, where the elements in a concept matrix represent relevant values between concepts. The implicit relevant values between concepts are inferred by the transitive closure of the concept matrix based on fuzzy logic. The proposed method is more flexible than the ones presented in [6] and [11] due to the fact that it has the capability to deal with interval queries and weighted-interval queries.**

I. Introduction

The PRIMARY purpose of establishing an information retrieval system lies in assisting the users to efficiently acquire desired information. Current models of information retrieval systems may be classified into the following categories [18]:

1) Boolean logic models.
2) Vector space models.
3) Probabilistic models.
4) Fuzzy set models.

Most commercial information retrieval systems currently still adopt the Boolean logic model. However, the information retrieval systems based on the Boolean logic model are rather restricted in applications since these systems are unable to represent uncertain information. If there is uncertain information, the query processing of these systems is not handled properly.

Several fuzzy information retrieval methods based on fuzzy set theory [21] have been proposed for improving the disadvantage of the Boolean logic model which is incapable of handling uncertain information, such as [6], [11]–[15], [19], [20], and [22]. However, either efficiency or effectiveness of these methods are not satisfactory. In this paper,we propose a knowledge-based fuzzy information retrieval method to deal with documents retrieval, where the concept matrices are used for knowledge representation. The elements in a concept matrix represent relevant values between concepts. The implicit relevant values between concepts can be inferred by the transitive closure of the concept matrix based on fuzzy logic [21]. The proposed method allows the system's users to perform interval queries and weighted-interval queries. Efficient retrieving capability and flexible user's queries are consequently provided for.

The rest of this paper is organized as follows. In Section II, we briefly review the theory of fuzzy sets. In Section III, the concepts of concept networks are introduced. In Section IV, the concepts of concept matrices and the transitive closure of the concept matrices are presented for knowledge representation. In Section V, we present the query processing techniques for fuzzy information retrieval. The conclusions are discussed in Section VI.

II. Fuzzy Set Theory

The theory of fuzzy sets was proposed by Zadeh in 1965 [21]. Let U be the universe of discourse, $U = \{u_1, u_2, \cdots, u_n\}$, and let A be a fuzzy set in U, then the fuzzy set A can be represented as:

$$A = \{(u_1, f_A(u_1)), (u_2, f_A(u_2)), \cdots, (u_n, f_A(u_n))\}, \quad (1)$$

where $f_A, f_A : U \rightarrow [0, 1]$, is the membership function of the fuzzy set A: $f_A(u_i)$ indicates the degree of membership of u_i in A.

If the universe of discourse U is a finite set, then the fuzzy set A can be expressed as follows:

$$A = f_A(u_1)/u_1 + f_A(u_2)/u_2 + \cdots + f_A(u_n)/u_n. \quad (2)$$

If the universe of discourse U is an infinite set, then the fuzzy set A can be expressed as:

$$A = \int_U f_A(u_i)/u_i, \quad u_i \in U. \quad (3)$$

Let f_A and f_B be the membership functions of the fuzzy sets A and B, respectively. The basic operations of fuzzy sets A and B are shown as follows:

1) Intersection:

$$f_{A \cap B}(u_i) = \mathrm{Min}\,(f_A(u_i), f_B(u_i)), \quad \forall u_i \in U. \quad (4)$$

2) Union:

$$f_{A \cup B}(u_i) = \mathrm{Max}\,(f_A(u_i), f_B(u_i)), \quad \forall u_i \in U. \quad (5)$$

3) Complement:

$$f_{\overline{A}}(u_i) = 1 - f_A(u_i), \quad \forall u_i \in U. \quad (6)$$

Manuscript received August 27, 1993; revised July 3, 1994. This work was supported by the National Science Council, Republic of China, under Grant NSC 83-0408-E-009-041.

The authors are with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, Republic of China.

IEEE Log Number 9409221.

0018–9472/95$04.00 © 1995 IEEE

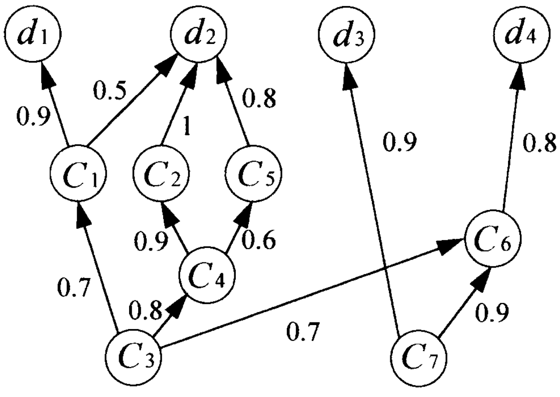


Fig. 1. A concept network.

III. Concept Networks

In [11], concept networks have been proposed for fuzzy information retrieval. A concept network includes nodes and directed links. Each node represents a concept or a document. Each directed link connects two concepts or directs from one concept C_i to one document d_j and is labeled with a real value between zero and one. If $C_i \xrightarrow{\mu} C_j$, then it indicates that the degree of relevance from concept C_i to concept C_j is μ, where $\mu \in$ [0, 1]. If $C_i \xrightarrow{\mu} d_j$, then it indicates that the degree of relevance of document d_j with respect to concept C_i is μ, where $\mu \in$ [0, 1]. Fig. 1 shows a concept network which is adapted from [11], where $C_1, C_2, \cdots$, and C_7 are concepts; d_1, d_2, d_3, and d_4 are documents. From Fig. 1, we can see that document d_2 can be expressed as a fuzzy subset of concepts, where

$$d_2 = \{(C_1, 0.5), (C_2, 1), (C_5, 0.8)\}.$$

Let C be a set of concepts, $C = \{C_1, C_2, \cdots, C_n\}$. A concept network is assumed to consist of n nodes and some directed links. Let the value associated with the directed link form concept C_i to concept C_j be denoted by $F(C_i, C_j)$, where F is a mapping function, $F : C \times C \rightarrow [0, 1]$, and $F(C_i, C_j) \in [0, 1]$. If the relevant value from concept C_i to concept C_j is $F(C_i, C_j)$, and if the relevant value from concept C_j to concept C_k is $F(C_j, C_k)$, then based on the transitivity of link relationships, the relevant value from concept C_i to concept C_k can be obtained by the following expression:

$$F(C_i, C_k) = \mathrm{Min}\,(F(C_i, C_j), F(C_j, C_k)). \tag{7}$$

Similarly, if $F(C_1, C_2), F(C_2, C_3), \cdots$, and $F(C_{n-1}, C_n)$ are known, then based on the transitivity of relationships, we can get

$$F(C_1, C_n) = \mathrm{Min}\,(F(C_1, C_2), F(C_2, C_3), \cdots, F(C_{n-1}, C_n)). \tag{8}$$

Each document has a different relevant value with respect to each concept. The document descriptor for the document d_j is defined as a fuzzy subset of the collection of concepts by the following expression:

$$d_j = \{(C_i, f_{d_j}(C_i)) | C_i \in C\},$$

where $f_{d_j}(C_i), f_{d_j} : C \rightarrow [0, 1]$, represents the degree of relevance of document d_j with respect to concept C_i.

Each user's query can be represented by a query descriptor Q expressed as a fuzzy subset of the collection of concepts by the following expression:

$$Q = \{(C_i, f_Q(C_i)) | C_i \in C\},$$

where $f_Q(C_i), f_Q : C \rightarrow [0, 1]$, represents the relevant value of the query descriptor Q with respect to the concept C_i.

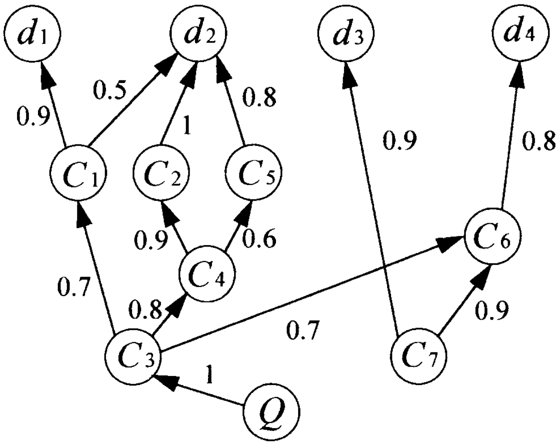


Fig. 2. A concept network of Example 3.1.

Example 3.1: Assume that the concept network shown in Fig. 2 consists of 4 documents d_1, d_2, d_3, d_4, and 7 concepts $C_1, C_2, \cdots, C_7$.

If the query descriptor Q is:

$$Q = \{(C_3, 1.0)\},$$

where 1.0 represents the relevant value of the query descriptor Q with respect to the concept C_3, then the relevant value of document d_2 with respect to concept C_3 can be calculated. From Fig. 2, we can see that there are three different routes which can be applied for determining the relevant value of document d_2 with respect to the concept C_3.

1) The first route is: $C_3 \rightarrow C_1 \rightarrow d_2$.

Based on [11], the relevant value of document d_2 with respect to concept C_3 can be determined as follows:

$$\mathrm{Min}\,(0.7, 0.5) = 0.5.$$

2) The second route is: $C_3 \rightarrow C_4 \rightarrow C_2 \rightarrow d_2$.

Based on [11], the relevant value of document d_2 with respect to concept C_3 can be evaluated as follows:

$$\mathrm{Min}\,(0.8, 0.9, 1) = 0.8.$$

3) The third route is: $C_3 \rightarrow C_4 \rightarrow C_5 \rightarrow d_2$.

Based on [11], the relevant value of document d_2 with respect to concept C_3 can be evaluated as follows:

$$\mathrm{Min}\,(0.8, 0.6, 0.8) = 0.6.$$

Then, based on [11], we can see that the relevant value of the document d_2 with respect to the concept C_3 is:

$$\mathrm{Max}\,(0.5, 0.8, 0.6) = 0.8.$$

The reasoning procedure should be repeated n times if there are n documents. If compound queries (i.e., queries with AND connectors or OR connectors) are used instead of simple queries, the reasoning process would slow down and become inefficient. As a result, once practically implemented, the concept network approach presented in [11] would have its limitations and would consequently be unable to satisfy the requirements of most users in terms of speed. Furthermore, in a concept network presented in [11], relevant values associated with directed links must be real values between zero and one. However, if we can allow them to be represented by real intervals between zero and one, then there is room for more flexibility. In this paper, we allow relevant values associated with directed links in a concept network to be represented by real intervals between zero and one.

IV. Concept Matrices

In this section, the definitions of concept matrices are presented to model the concept networks presented in [11]. The definitions of concept matrices and the transitive closure of the concept matrices are described as follows.

Definition 4.1: Let C be a set of concepts, $C = \{C_1, C_2, \cdots, C_n\}$. A concept matrix M is a fuzzy matrix [8]; $M(C_i, C_j)$ represents the relevant value from the concept C_i to the concept C_j, where $M(C_i, C_j) \in [0, 1]$.

A concept matrix M has the following properties:

1) Reflexivity,

$$M(C_i, C_i) = 1, \quad \forall C_i \in C.$$

2) M may not be symmetric,

$$M(C_i, C_j) \neq M(C_j, C_i).$$

3) Transitivity,

$$M(C_i, C_k) \geq \underset{C_j \in C}{\mathrm{Max}}\, \mathrm{Min}\,[M(C_i, C_j), M(C_j, C_k)].$$

Definition 4.2: Let M be a concept matrix,

$$M = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix}$$

where n is the number of concepts, $f_{ij} \in [0, 1], 1 \leq i \leq n$, and $1 \leq j \leq n$, and let (see (9) at the bottom of the page), where "$\vee$" represents the Max operation, and "$\wedge$" represents the Min operation. Then, there exists an integer $p \leq n - 1$, such that $M^p = M^{p+1} = M^{p+2} = \cdots$ (please see [8, p. 117]). Let $T = M^p, T$ is called the transitive closure of the concept matrix M.

Sometimes, the relevant value between concepts may hardly be represented by a crisp real value between 0 and 1. In this case, the relevant value between concepts can be represented by a real interval $[f_{ij}^l, f_{ij}^h]$ between 0 and 1 to describe the relevant value from concept C_i to concept C_j, where $0 \leq f_{ij}^l \leq f_{ij}^h \leq 1$.

Definition 4.3: Let M be a concept matrix,

$$M = \begin{bmatrix} [f_{11}^l, f_{11}^h] & [f_{12}^l, f_{12}^h] & \cdots & [f_{1n}^l, f_{1n}^h] \\ [f_{21}^l, f_{21}^h] & [f_{22}^l, f_{22}^h] & \cdots & [f_{2n}^l, f_{2n}^h] \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ [f_{n1}^l, f_{n1}^h] & [f_{n2}^l, f_{n2}^h] & \cdots & [f_{nn}^l, f_{nn}^h] \end{bmatrix},$$

where $[f_{ij}^l, f_{ij}^h]$ indicates the relevant value from concept C_i to concept $C_j, 0 \leq f_{ij}^l \leq f_{ij}^h \leq 1, 1 \leq i \leq n, 1 \leq j \leq n$, and n is the number of concept, and let (see (10) at the bottom of the next page), where "$\vee$" represents the Max operation and "$\wedge$" represents the Min operation.Then, there exists an integer $p \leq n - 1$, such that $M^p = M^{p+1} = M^{p+2} = \cdots$. Let $T = M^p, T$ is called the transitive closure of the concept matrix M.

V. Query Processing Techniques

Let P be a set of documents, $P = \{d_1, d_2, \cdots, d_m\}$, and C be a set of concepts, $C = \{C_1, C_2, \cdots, C_n\}$. A document in a document retrieval system is generally described by a set of concepts with each concept representing a topic. The

$$\begin{aligned} M^2 &= M \otimes M \\ &= \begin{bmatrix} \bigvee\limits_{i=1,\cdots,n} (f_{1i} \wedge f_{i1}, & \bigvee\limits_{i=1,\cdots,n} (f_{1i} \wedge f_{i2}) & \cdots & \bigvee\limits_{i=1,\cdots,n} (f_{1i} \wedge f_{in}) \\ \bigvee\limits_{i=1,\cdots,n} (f_{2i} \wedge f_{i1}) & \bigvee\limits_{i=1,\cdots,n} (f_{2i} \wedge f_{i2}) & \cdots & \bigvee\limits_{i=1,\cdots,n} (f_{2i} \wedge f_{in}) \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \bigvee\limits_{i=1,\cdots,n} (f_{ni} \wedge f_{i1}) & \bigvee\limits_{i=1,\cdots,n} (f_{ni} \wedge f_{i2}) & \cdots & \bigvee\limits_{i=1,\cdots,n} (f_{ni} \wedge f_{in}) \end{bmatrix} \end{aligned} \tag{9}$$

relations between documents and concepts can be represented by a document descriptor matrix D shown as follows:

$$D = \begin{array}{c} \\ d_1 \\ d_2 \\ \cdot \\ \cdot \\ \cdot \\ d_m \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_n \end{array} \\ \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1n} \\ t_{21} & t_{22} & \cdots & t_{2n} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ t_{m1} & t_{m2} & \cdots & t_{mn} \end{pmatrix} \end{array}.$$

where m is the number of documents, n is the number of concepts, t_{ij} represents the degree of relevance of document d_i with respect to concept $C_j, t_{ij} \in [0,1], 1 \leq i \leq m$, and $1 \leq j \leq n$.

In a document descriptor matrix D, the degree of relevance of each document with respect to a specific concept is determined by experts. However, an expert may possibly neglect the degree of relevance of certain documents with respect to some specific concepts. Because concepts may be not independent from each other, the transitive closure T of the concept matrix M can be used to evaluate the implicit relevant values of each document with respect to specific concepts to improve this. Let $D^* = D \otimes T$, where D is the document descriptor matrix and T is the transitive closure of the concept matrix M. The document descriptor matrix D^* indicates the degrees of relevance of each document with respect to specific concepts, and is used as a basis for similarity measures between queries and documents as described later.

Based on the vector representation method [1], the query descriptor Q can be represented by a query descriptor vector $\overline{q}$, i.e.,

$$Q = \{(C_1, x_1), (C_2, x_2), \cdots, (C_n, x_n), \overline{q} = \langle x_1, x_2, \cdots, x_n \rangle,$$

where $x_i \in [0,1], 1 \leq i \leq n$, indicates the degree of strength that the desired documents contain concept C_i. In a query descriptor vector $\overline{q}$, if $x_i = 0$, then it indicates that documents desired by the user must not contain the concept C_i. Furthermore, if the user considers that certain concepts may be neglected, then the user does not have to assign the degrees of strength with respect to such concepts in the query descriptor vector $\overline{q}$. The symbol "-" is used for labeling a neglected concept. Therefore, if $x_i =$ "-", then it indicates that concept C_i is a neglected concept. In this case, the concept C_i would not be considered in the document retrieval process.

Example 5.1: Assume there are five concepts in a fuzzy information retrieval system, and assume that a query descriptor Q is specified to retrieve any document having a degree of strength of 0.6 with respect to concept C_2, a degree of strength of 0.9 with respect to concept C_4, and concept C_5 excluded, i.e., $Q = \{(C_2, 0.6), (C_4, 0.9), (C_5, 0)\}$. Then, based on the vector representation method, the query descriptor Q can be represented by a query descriptor vector $\overline{q}$ shown as follows:

$$\overline{q} = \langle -, 0.6, -, 0.9, 0 \rangle.$$

In [5], a similarity measure was proposed for calculating the degree of similarity between two real values. Let x and y be two real values between zero and one. The degree of similarity between $x$and y can be calculated by the function T, i.e.,

$$T(x, y) = 1 - |x - y|, \tag{11}$$

where $T(x, y) \in [0, 1]$. The larger the value of $T(x, y)$, the higher the degree of similarity between x and y.

Let $\overline{d_i}$ be a document descriptor vector and $\overline{q}$ be a query descriptor vector, where

$$\begin{aligned} \overline{d_i} &= \langle t_{i1}, t_{i2}, \cdots, t_{in} \rangle \\ \overline{q} &= \langle x_1, x_2, \cdots, x_n \rangle, \end{aligned}$$

$t_{ij} \in [0,1], x_j \in [0,1], 1 \leq j \leq n, 1 \leq i \leq m, n$ is the number of concepts, and m is the number of documents. Let $\overline{q(j)}$ denote the jth component of the query descriptor vector $\overline{q}$. If $\overline{q(j)} \neq$ "-", then it indicates that the concept C_j is a neglected concept with respect to the query. Then, the degree of similarity between the document descriptor vector $\overline{d_i}$ and the query descriptor vector $\overline{q}$ can be calculated as follows:

$$RS(d_i) = \frac{\displaystyle\sum_{\overline{q(j)} \neq \text{“-”} \text{ and } j=1,\cdots,n} T(t_{ij}, x_j)}{k} \tag{12}$$

where $RS(d_i) \in [0, 1]$, and k is the number of not neglected concepts in the query descriptor vector $\overline{q}$. The relevant values of neglected concepts can be eliminated from the computation

$$\begin{aligned} M^2 &= M \odot M \\ &= \begin{bmatrix} \left[\bigvee\limits_{i=1,\cdots,n} (f^l_{1i} \wedge f^l_{i1}), \bigvee\limits_{i=1,\cdots,n} (f^h_{1i} \wedge f^h_{i1})\right] & \left[\bigvee\limits_{i=1,\cdots,n} (f^l_{1i} \wedge f^l_{i2}), \bigvee\limits_{i=1,\cdots,n} (f^h_{1i} \wedge f^h_{i2})\right] & \cdots & \left[\bigvee\limits_{i=1,\cdots,n} (f^l_{1i} \wedge f^l_{in}), \bigvee\limits_{i=1,\cdots,n} (f^h_{1i} \wedge f^l_{in})\right] \\ \left[\bigvee\limits_{i=1,\cdots,n} (f^l_{2i} \wedge f^l_{i1}), \bigvee\limits_{i=1,\cdots,n} (f^h_{2i} \wedge f^h_{i1})\right] & \left[\bigvee\limits_{i=1,\cdots,n} (f^l_{2i} \wedge f^l_{i2}), \bigvee\limits_{i=1,\cdots,n} (f^h_{2i} \wedge f^h_{i2})\right] & \cdots & \left[\bigvee\limits_{i=1,\cdots,n} (f^l_{2i} \wedge f^l_{in}), \bigvee\limits_{i=1,\cdots,n} (f^h_{2i} \wedge f^h_{in})\right] \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \left[\bigvee\limits_{i=1,\cdots,n} (f^l_{ni} \wedge f^l_{i1}), \bigvee\limits_{i=1,\cdots,n} (f^h_{ni} \wedge f^h_{i1})\right] & \left[\bigvee\limits_{i=1,\cdots,n} (f^l_{ni} \wedge f^l_{i2}), \bigvee\limits_{i=1,\cdots,n} (f^h_{ni} \wedge f^h_{i2})\right] & \cdots & \left[\bigvee\limits_{i=1,\cdots,n} (f^l_{ni} \wedge f^l_{in}), \bigvee\limits_{i=1,\cdots,n} (f^h_{ni} \wedge f^h_{in})\right] \end{bmatrix} \end{aligned} \tag{10}$$

of similarity since these neglected concepts are not necessary factors for the retrieval process.

A query expression can be subdivided into two types:

1) AND-connected queries: A query can be easily expressed by a query descriptor vector representing the "AND" connections of concepts. For example, consider the following query descriptor vector $\overline{q}$,

$$\overline{q} = \langle 0.6, —, 0.5, 0.7, — \rangle,$$

where the query descriptor vector $\overline{q}$ means that the user wishes to retrieve any document containing the concepts C_1, C_3, and C_4 having the degrees of strength of 0.6, 0.5, and 0.7, respectively.

2) OR-connected queries: A query expression is expressed by a number of query descriptors connected by OR connectors. For example, consider the following query expression:

$$\langle —, —, 0.8, —, — \rangle \quad \text{OR} \quad \langle —, 0.7, —, —, — \rangle.$$

It indicates that the user wishes to retrieve any document possessing the concept C_3 having a degree of strength of 0.8 or possessing the concept C_2 having a degree of strength of 0.7.

Consider the following OR-connected query:

$$\overline{q_1} \quad \text{OR} \quad \overline{q_2},$$

the degree of similarity between the query descriptor vector $\overline{q_j}$ and the documents can be expressed by a $1 \times m$ matrix RS_j, where m is the number of documents, and $1 \leq j \leq 2$. The degree of similarity between the query and the documents can be calculated as follows:

$$RS^*(d_i) = \text{Max}\,(RS_1(d_i), RS_2(d_i)). \tag{13}$$

where $RS_1(d_i)$ represents the degree of similarity between the query descriptor vector $\overline{q_1}$ and the ith row of the document descriptor matrix D^*. $RS_2(d_i)$ represents the degree of similarity between the query descriptor vector $\overline{q_2}$ and the ith row of the document descriptor matrix D^*, the retrieval status value $RS^*(d_i)$ represents the degree of similarity between the query and the document d_i, and $1 \leq i \leq m$. The information retrieval system would display every document having a retrieval status value greater than the threshold value λ, where $\lambda \in [0, 1]$, in a sequential order from the document with the highest retrieval status value to that with the lowest one.

Example 5.2: A concept matrix is assumed to be composed of seven concepts $C_1, C_2, \cdots, C_7$, and assume that there are seven documents in a fuzzy information retrieval system. Furthermore, assume that the document descriptor matrix D and the concept matrix M are shown as follows:

$$D = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.4 & 0 & 0 & 0 & 0.8 \\ 0 & 0.4 & 1 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0.9 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0.7 \\ 0 & 0.8 & 0.5 & 0 & 0.9 & 0.7 & 1 \end{bmatrix},$$

$$M = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.4 & 0 & 0 & 0 & 0.8 \\ 0 & 0.4 & 1 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0.9 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0.7 \\ 0 & 0.8 & 0.5 & 0 & 0.9 & 0.7 & 1 \end{bmatrix}.$$

In this case, the transitive closure T of the concept matrix M is:

$$T = \begin{bmatrix} 1 & 1 & 1 & 0 & 0.8 & 0.7 & 0.8 \\ 0 & 1 & 0.5 & 0 & 0.8 & 0.7 & 0.8 \\ 0 & 0.5 & 1 & 0 & 0.5 & 0.5 & 0.5 \\ 0 & 0.8 & 0.5 & 1 & 1 & 1 & 0.9 \\ 0 & 0.8 & 0.5 & 0 & 1 & 0.7 & 0.9 \\ 0 & 0.7 & 0.5 & 0 & 0.7 & 1 & 0.7 \\ 0 & 0.8 & 0.5 & 0 & 0.9 & 0.7 & 1 \end{bmatrix}.$$

The document descriptor matrix D^* can be obtained based on the document descriptor matrix D and the transitive closure T of the concept matrix M shown as follows:

$$D^* = D \otimes T = \begin{bmatrix} 0.5 & 0.7 & 1 & 0 & 0.7 & 0.7 & 0.7 \\ 1 & 1 & 1 & 0.4 & 1 & 0.7 & 0.9 \\ 0 & 1 & 0.5 & 0.5 & 0.9 & 0.7 & 1 \\ 0.6 & 0.7 & 0.9 & 0.4 & 0.7 & 1 & 0.7 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0.9 \\ 0.8 & 0.8 & 0.8 & 0.7 & 0.9 & 0.7 & 0.9 \\ 0 & 0.9 & 0.8 & 0.9 & 0.9 & 0.9 & 0.9 \end{bmatrix}.$$

Case 1: If the query descriptor vector $\overline{q}$ is:

$$\overline{q} = \langle 0.6, —, —, 0, 0.8, —, — \rangle,$$

then we can get

$$\begin{aligned} RS(d_1) &= 0.93 \\ RS(d_2) &= 0.67 \\ RS(d_3) &= 0.6 \\ RS(d_4) &= 0.83 \\ RS(d_5) &= 0.47 \\ RS(d_6) &= 0.67 \\ RS(d_7) &= 0.47. \end{aligned}$$

If the retrieval threshold value $\lambda = 0.5$, then the list of documents responding to the user is listed in a sequential order from the document with the highest retrieval status value to the one with the lowest retrieval status value, i.e., $d_1 > d_4 > d_2 > d_6 > d_3$. In this case, the documents d_5 and d_7 are not listed since their retrieval status values are less than the threshold value λ, where $\lambda = 0.5$.

Case 2: If the query expression represented by query descriptor vectors shown as follows:

$$\langle 0.6, —, —, —, —, —, — \rangle \quad \text{OR} \quad \langle —, —, —, —, —, —, 0.8 \rangle$$

then we can see that this OR-connected query expression is subdivided into two query descriptor vectors $\overline{q_1}$ and $\overline{q_2}$, i.e.,

$$\overline{q_1} = \langle 0.6, —, —, —, —, —, — \rangle$$
$$\overline{q_2} = \langle —, —, —, —, —, —, 0.8 \rangle.$$

Then, we can get

$$\begin{aligned} RS_1(d_1) &= 0.9, & RS_2(d_1) &= 0.9 \\ RS_1(d_2) &= 0.6, & RS_2(d_2) &= 0.9 \\ RS_1(d_3) &= 0.4, & RS_2(d_3) &= 0.8 \\ RS_1(d_4) &= 1.0, & RS_2(d_4) &= 0.9 \\ RS_1(d_5) &= 0.6, & RS_2(d_5) &= 0.9 \\ RS_1(d_6) &= 0.8, & RS_2(d_6) &= 0.9 \\ RS_1(d_7) &= 0.4, & RS_2(d_7) &= 0.9. \end{aligned}$$

Based on (13), we can get

$$\begin{aligned} RS^*(d_1) &= 0.9 \\ RS^*(d_2) &= 0.9 \\ RS^*(d_3) &= 0.8 \\ RS^*(d_4) &= 1 \\ RS^*(d_5) &= 0.9 \\ RS^*(d_6) &= 0.9 \\ RS^*(d_7) &= 0.9. \end{aligned}$$

If the retrieval threshold value is 0.5, then all the documents would be retrieved.

Consider the case of weighted queries for fuzzy information retrieval, where each query item in a weighted query can be expressed as an order pair:

$$(x_j, w_j),$$

where x_j is the degree of strength of the user's desired documents with respect to concept C_j, and w_j is the weighted value of the query with respect to concept C_j. In this case, then the query descriptor vector $\overline{q}$ can be expressed as follows:

$$\overline{q} = \langle (x_1, w_1), (x_2, w_2), \cdots, (x_n, w_n) \rangle,$$

where $0 \leq x_j \leq 1, 0 \leq w_j \leq 1$, and $1 \leq j \leq n$.

Assume that ith row of the document descriptor matrix D^* is $\langle t_{i1}, t_{i2}, \cdots, t_{in} \rangle$, where $t_{ij} \in [0, 1], 1 \leq i \leq m$, and $1 \leq j \leq n$. Then, the retrieval status value $RS_w(d_i)$ of the document d_i with respect to the query can be calculated as follows:

$$RS_w(d_i) = \frac{\sum\limits_{\overline{q(j)} \neq \text{“-”} \text{ and } j=1,\cdots,n} T(t_{ij}, x_j)}{k} \times w_j \tag{14}$$

where k is the number of not neglected concepts in the query descriptor vector $\overline{q}, \overline{q(j)}$ denotes the jth component of the query descriptor vector $\overline{q}, RS_w(d_i) \in [0, 1], 1 \leq i \leq m$, and $\Sigma_{j=1}^{m} w_j = 1$.

In the following, we present the interval query processing techniques. By using the vector representation method, a document d_j can be represented by a document descriptor vector $\overline{d_j}$ as follows:

$$\overline{d_j} = \langle [t_1^l, t_1^h], [t_2^l, t_2^h], \cdots, [t_n^l, t_n^h] \rangle,$$

where $[t_i^l, t_i^h]$ indicates the degree of strength that the document d_j contains the concept $C_i, 0 \leq t_i^l \leq t_i^h \leq 1$, and $1 \leq i \leq n$. Similarly, a query can also be represented by a query descriptor vector $\overline{q}$ as follows:

$$\overline{q} = \langle [x_1^l, x_1^h], [x_2^l, x_2^h], \cdots, [x_n^l, x_n^h] \rangle,$$

where $[x_i^l, x_i^h]$ indicates the degree of strength that the user desired documents contain the concept $C_i, 0 \leq x_i^l \leq x_i^h \leq 1, 1 \leq i \leq n$, and n is the number of concepts.

The relations between documents and concepts can be represented by a document descriptor matrix D shown as follows:

$$D = \begin{array}{c} \\ d_1 \\ d_2 \\ \cdot \\ \cdot \\ \cdot \\ d_m \end{array} \begin{array}{c} \begin{array}{cccc} C_1 & C_2 & \cdots & C_n \end{array} \\ \begin{pmatrix} [t_{11}^l, t_{11}^h] & [t_{12}^l, t_{12}^h] & \cdots & [t_{1n}^l, t_{1n}^h] \\ [t_{21}^l, t_{21}^h] & [t_{22}^l, t_{22}^h] & \cdots & [t_{2n}^l, t_{2n}^h] \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ [t_{m1}^l, t_{m1}^h] & [t_{m2}^l, t_{m2}^h] & \cdots & [t_{mn}^l, t_{mn}^h] \end{pmatrix} \end{array}$$

where m is the number of documents, n is the number of concepts, $[t_{ij}^l, t_{ij}^h]$ represents the degree of relevance of document d_i with respect to concept $C_j, 0 \leq t_{ij}^l \leq t_{ij}^h \leq 1, 1 \leq i \leq m$, and $1 \leq j \leq n$.

Let D represent the document descriptor matrix and T represent the transitive closure of the concept matrix M, where

$$T = \begin{bmatrix} [f_{11}^l, f_{11}^h] & [f_{12}^l, f_{12}^h] & \cdots & [f_{1n}^l, f_{1n}^h] \\ [f_{21}^l, f_{21}^h] & [f_{22}^l, f_{22}^h] & \cdots & [f_{2n}^l, f_{2n}^h] \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ [f_{n1}^l, f_{n1}^h] & [f_{n2}^l, f_{n2_j}^h & \cdots & [f_{nn}^l, f_{nn}^h] \end{bmatrix}.$$

Then, let (see (15) at the bottom of the next page).

The document descriptor matrix D^* indicates the degrees of relevance of each document with respect to specific concepts. D^* would be used as a basis for similarity measures between queries and documents.

The user's query expression can be represented by a query descriptor vector $\overline{q}$, i.e.,

$$\overline{q} = \langle [x_1^l, x_1^h], [x_2^l, x_2^h], \cdots, [x_n^l, x_n^h] \rangle,$$

where $[x_i^l, x_i^h]$ indicates the desired degree of strength of the concept C_i with respect to the query, $0 \leq x_i^l \leq x_i^h \leq 1, and\ 1 \leq i \leq n$. The symbol “-” is used for labeling a neglected concept so that such a concept would not be considered in a retrieval process.

In [23], a similarity measure was described to measure the distance between two real intervals. Let A and B be two real intervals contained in $[\beta_1, \beta_2]$, where $A = [a_1, a_2]$ and

$B = [b_1, b_2]$. The distance between the intervals A and B can be calculated as follows:

$$\Delta(A. B) = \frac{(|a_1 - b_1| + |a_2 - b_2|)}{2(\beta_2 - \beta_1)}. \qquad (16)$$

It is obvious that if A and B are identical intervals, then $\Delta(A, B) = 0$.

Based on (16), the degree of similarity between the intervals A and B can be measured. If A and B are both real intervals in [0, 1], where $A = [a_1, a_2]$ and $B = [b_1, b_2]$, then

$$S(A, B) = \begin{cases} 1, & \text{if } b_1 \leq a_1 \leq a_2 \leq b_2 \\ 1 - \dfrac{(|a_1 - b_1| + |a_2 - b_2|)}{2}, & \text{otherwise} \end{cases} \qquad (17)$$

where $S(A, B) \in [0, 1]$. The larger the value of $S(A, B)$, the higher the similarity between the intervals A and B.

A document descriptor vector $\overline{d_i}$ and a query descriptor vector $\overline{q}$ are assumed, i.e.,

$$\overline{d_i} = \langle [t_{i1}^l, t_{i1}^h], [t_{i2}^l, t_{i2}^h], \cdots, [t_{in}^l, t_{in}^h] \rangle$$
$$\overline{q} = \langle [x_1^l, x_1^h], [x_2^l, x_2^h], \cdots, [x_n^l, x_n^h] \rangle.$$

where $1 \leq j \leq n, 1 \leq i \leq m, n$ is the number of concepts, m is the number of documents. Let $\overline{q(j)}$ denote the jth component of the query descriptor vector $\overline{q}$. If $\overline{q(j)} =$ "-", then it indicates that the concept C_j is a neglected concept with respect to the query. Based on (17), the degree of similarity between the document descriptor vector and $\overline{d_i}$ the query descriptor vector $\overline{q}$ can be calculated as follows:

$$RSV(d_i) = \frac{\displaystyle\sum_{\overline{q(j)} \neq \text{"-"} \text{ and } j=1,\cdots,n} S([t_{ij}^l, t_{ij}^h], [x_j^l, x_j^h])}{k}. \qquad (18)$$

where the retrieval status value $RSV(d_i)$ indicates the degree of the similarity between the query and the document $d_i, RSV(d_i) \in [0, 1], 1 \leq i \leq m$, and k is the number of not neglected concepts in the query. The larger the value of $RSV(d_i)$, the higher the similarity between the query and the document d_i. The relevant values of neglected concepts can be eliminated from the computation of similarity since these neglected concepts are not necessary factors for the retrieval process.

Consider the following OR-connected query:

$$\overline{q_1} \quad \text{OR} \quad \overline{q_2},$$

where $\overline{q_1}$ and $\overline{q_2}$ are query descriptor vectors. The degree of similarity between the query descriptor vector $\overline{q_j}$ and the documents can be expressed by a $1 \times m$ matrix RSV_j, where m is the number of documents, and $1 \leq j \leq 2$. In this case, the degree of similarity between the query and the documents can be calculated as follows:

$$RSV^*(d_i) = \text{Max}\,(RSV_1(d_i), RSV_2(d_i)), \qquad (19)$$

where $RSV_1(d_i)$ represents the degree of similarity between the query descriptor vector $\overline{q_1}$ and the ith row of the document descriptor matrix D^*, $RSV_2(d_i)$ represents the degree of similarity between the query descriptor vector $\overline{q_2}$ and the ith row of the document descriptor matrix D^*, the retrieval status value $RSV^*(d_i)$ represents the degree of similarity of the query with respect to document d_i, and $1 \leq i \leq m$. The information retrieval system would display every document having a retrieval status value greater than the threshold value λ in a sequential order from the document with the highest degree of retrieval status value to that with the lowest one, where $\lambda \in [0, 1]$.

Example 5.3: Assume that the threshold value λ is 0.5 and assume that there are seven concepts $C_1, C_2, \cdots, C_7$, and seven documents $d_1, d_2, \cdots, d_7$. Furthermore, assume that the document descriptor matrix D and the concept matrix M have the following forms (see bottom of next page). In this case, the transitive closure T of the concept matrix M can be calculated as follows (see bottom of next page).

The document descriptor matrix D^* can be obtained based on the document descriptor matrix D and the transitive closure T of the concept matrix M as follows (see bottom of next page): if the user's OR-connected query is:

$$\langle [0.5, 0.8], —, —, [0.3, 0.7], [0.7, 1], —, — \rangle \quad \text{OR}$$
$$\langle —, [0.6, 0.9], [0.4, 0.6], —, —, —, — \rangle$$

$$D^* = D \odot T$$
$$= \begin{bmatrix} \left[\bigvee_{i=1,\cdots,n} (f_{1i}^l \wedge f_{i1}^l), \bigvee_{i=1,\cdots,n} (f_{1i}^h \wedge f_{i1}^h) \right] & \left[\bigvee_{i=1,\cdots,n} (f_{1i}^l \wedge f_{i2}^l), \bigvee_{i=1,\cdots,n} (f_{1i}^h \wedge f_{i2}^h) \right] & \cdots & \left[\bigvee_{i=1,\cdots,n} (f_{1i}^l \wedge f_{in}^l), \bigvee_{i=1,\cdots,n} (f_{1i}^h \wedge f_{in}^h) \right] \\ \left[\bigvee_{i=1,\cdots,n} (f_{2i}^l \wedge f_{i1}^l), \bigvee_{i=1,\cdots,n} (f_{2i}^h \wedge f_{i1}^h) \right] & \left[\bigvee_{i=1,\cdots,n} (f_{2i}^l \wedge f_{i2}^l), \bigvee_{i=1,\cdots,n} (f_{2i}^h \wedge f_{i2}^h) \right] & \cdots & \left[\bigvee_{i=1,\cdots,n} (f_{2i}^l \wedge f_{in}^l), \bigvee_{i=1,\cdots,n} (f_{2i}^h \wedge f_{in}^h) \right] \\ \vdots & \vdots & \cdots & \vdots \\ \left[\bigvee_{i=1,\cdots,n} (f_{ni}^l \wedge f_{i1}^l), \bigvee_{i=1,\cdots,n} (f_{ni}^h \wedge f_{i1}^h) \right] & \left[\bigvee_{i=1,\cdots,n} (f_{ni}^l \wedge f_{i2}^l), \bigvee_{i=1,\cdots,n} (f_{ni}^h \wedge f_{i2}^h) \right] & \cdots & \left[\bigvee_{i=1,\cdots,n} (f_{ni}^l \wedge f_{in}^l), \bigvee_{i=1,\cdots,n} (f_{ni}^h \wedge f_{in}^h) \right] \end{bmatrix} \qquad (15)$$

then we can get

$$\begin{aligned}
RSV_1(d_1) &= 0.83, & RSV_2(d_1) &= 0.75\\
RSV_1(d_2) &= 0.88, & RSV_2(d_2) &= 0.62\\
RSV_1(d_3) &= 0.78, & RSV_2(d_3) &= 0.88\\
RSV_1(d_4) &= 1, & RSV_2(d_4) &= 0.8\\
RSV_1(d_5) &= 0.72, & RSV_2(d_5) &= 0.63\\
RSV_1(d_6) &= 1, & RSV_2(d_6) &= 0.85\\
RSV_1(d_7) &= 0.65, & RSV_2(d_7) &= 0.85.
\end{aligned}$$

Based on (19), we can get

$$\begin{aligned}
RSV^*(d_1) &= 0.83\\
RSV^*(d_2) &= 0.88\\
RSV^*(d_3) &= 0.88\\
RSV^*(d_4) &= 1\\
RSV^*(d_5) &= 0.72\\
RSV^*(d_6) &= 1\\
RSV^*(d_7) &= 0.85.
\end{aligned}$$

The larger the value of $RSV^*(d_i)$, the more suitable the document d_i to the user's interval query, where $1 \leq i \leq 7$.

$$D = \begin{array}{cccccccc}
 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7\\
d_1 & [0.5, 0.5] & [0.7, 0.7] & [1, 1] & [0, 0] & [0, 0] & [0.6, 0.6] & [0, 0]\\
d_2 & [1, 1] & [0.6, 0.6] & [0, 0] & [0.4, 0.4] & [1, 1] & [0, 0] & [0, 0]\\
d_3 & [0, 0] & [1, 1] & [0, 0] & [0.5, 0.5] & [0.5, 0.5] & [0.4, 0.4] & [1, 1]\\
d_4 & [0.6, 0.6] & [0.5, 0.5] & [0.9, 0.9] & [0.4, 0.4] & [0, 0] & [1, 1] & [0.6, 0.6]\\
d_5 & [1, 1] & [0, 0] & [0.7, 0.7] & [1, 1] & [0, 0] & [0.5, 0.5] & [0.7, 0.7]\\
d_6 & [0.8, 0.8] & [0.4, 0.4] & [0.5, 0.5] & [0.7, 0.7] & [1, 1] & [0, 0] & [1, 1]\\
d_7 & [0, 0] & [0.9, 0.9] & [0.8, 0.8] & [0.9, 0.9] & [0, 0] & [1, 1] & [1, 1]
\end{array}$$

$$M = \begin{bmatrix}
[1, 1] & [1, 1] & [1, 1] & [0, 0] & [0, 0] & [0, 0] & [0, 0]\\
[0, 0] & [1, 1] & [0.4, 0.4] & [0, 0] & [0, 0] & [0, 0] & [0.8, 0.8]\\
[0, 0] & [0.4, 0.4] & [1, 1] & [0, 0] & [0, 0] & [0, 0] & [0.5, 0.5]\\
[0, 0] & [0, 0] & [0, 0] & [1, 1] & [1, 1] & [1, 1] & [0, 0]\\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [1, 1] & [0, 0] & [0.9, 0.9]\\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [1, 1] & [0.7, 0.7]\\
[0, 0] & [0.8, 0.8] & [0.5, 0.5] & [0, 0] & [0.9, 0.9] & [0.7, 0.7] & [1, 1]
\end{bmatrix}$$

$$T = \begin{bmatrix}
[1, 1] & [1, 1] & [1, 1] & [0, 0] & [0.8, 0.8] & [0.7, 0.7] & [0.8, 0.8]\\
[0, 0] & [1, 1] & [0.5, 0.5] & [0, 0] & [0.8, 0.8] & [0.7, 0.7] & [0.8, 0.8]\\
[0, 0] & [0.5, 0.5] & [1, 1] & [0, 0] & [0.5, 0.5] & [0.5, 0.5] & [0.5, 0.5]\\
[0, 0] & [0.8, 0.8] & [0.5, 0.5] & [1, 1] & [1, 1] & [1, 1] & [0.9, 0.9]\\
[0, 0] & [0.8, 0.8] & [0.5, 0.5] & [0, 0] & [1, 1] & [0.7, 0.7] & [0.9, 0.9]\\
[0, 0] & [0.7, 0.7] & [0.5, 0.5] & [0, 0] & [0.7, 0.7] & [1, 1] & [0.7, 0.7]\\
[0, 0] & [0.8, 0.8] & [0.5, 0.5] & [0, 0] & [0.9, 0.9] & [0.7, 0.7] & [1, 1]
\end{bmatrix}$$

$$\begin{aligned}
D^* &= D \odot T\\
&= \begin{bmatrix}
[0.5, 0.5] & [0.7, 0.7] & [1, 1] & [0, 0] & [0.7, 0.7] & [0.7, 0.7] & [0.7, 0.7]\\
[1, 1] & [1, 1] & [1, 1] & [0.4, 0.4] & [1, 1] & [0.7, 0.7] & [0.9, 0.9]\\
[0, 0] & [1, 1] & [0.5, 0.5] & [0.5, 0.5] & [0.9, 0.9] & [0.7, 0.7] & [1, 1]\\
[0.6, 0.6] & [0.7, 0.7] & [0.9, 0.9] & [0.4, 0.4] & [0.7, 0.7] & [1, 1] & [0.7, 0.7]\\
[1, 1] & [1, 1] & [1, 1] & [1, 1] & [1, 1] & [1, 1] & [0.9, 0.9]\\
[0.8, 0.8] & [0.8, 0.8] & [0.8, 0.8] & [0.7, 0.7] & [0.9, 0.9] & [0.7, 0.7] & [0.9, 0.9]\\
[0, 0] & [0.9, 0.9] & [0.8, 0.8] & [0.9, 0.9] & [0.9, 0.9] & [0.9, 0.9] & [0.9, 0.9]
\end{bmatrix}
\end{aligned}$$

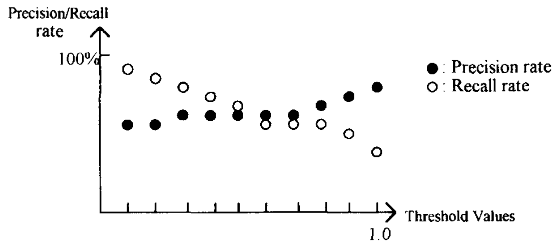


Fig. 3. The precision rate and recall rate with respect to different threshold values in a simple query.

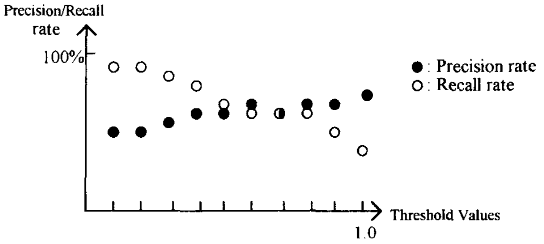


Fig. 4. The precision rate and recall rate with respect to different threshold values in an interval query.

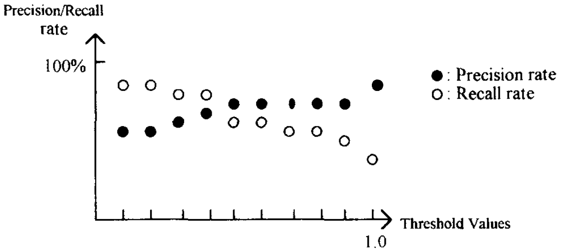


Fig. 5. The precision rate and recall rate with respect to different threshold values in a weighted query.

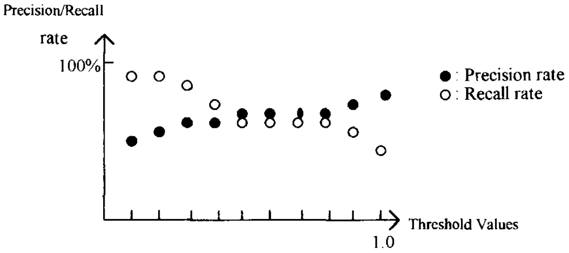


Fig. 6. The precision rate and recall rate with respect to different threshold values in a weighted-interval query.

From the above results, we can see that all of the retrieval status values of the documents are larger than the threshold value λ, where $\lambda = 0.5$. We also can see that the documents d_4 and d_6 are most suitable to the user's interval query due to the fact that they have the largest retrieval status value.

Weighted-interval queries can also be processed by our method. In weighted-interval queries, a query expression can be represented by a query descriptor vector $\overline{q}$ shown as follows:

$$\overline{q} = \langle([x_1^l . x_1^h]. w_1). ([x_2^l . x_2^h]. w_2). \cdots . ([x_n^l . x_n^h]. w_n)\rangle$$

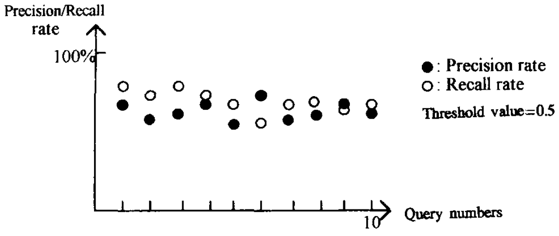


Fig. 7. The precision rate and recall rate with respect to query numbers in simple queries.

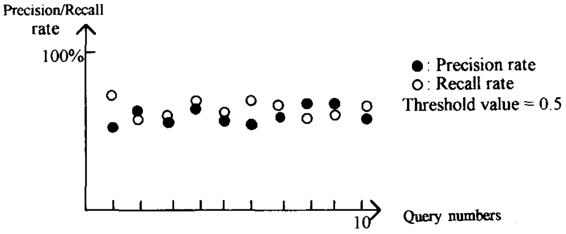


Fig. 8. The precision rate and recall rate with respect to query numbers in weighted queries.

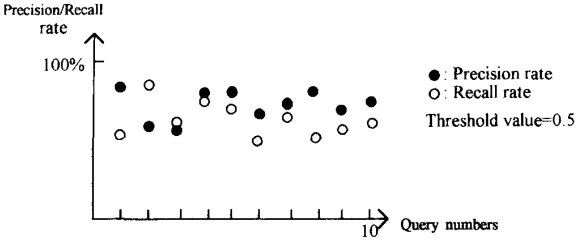


Fig. 9. The precision rate and recall rate with respect to query numbers in interval queries.

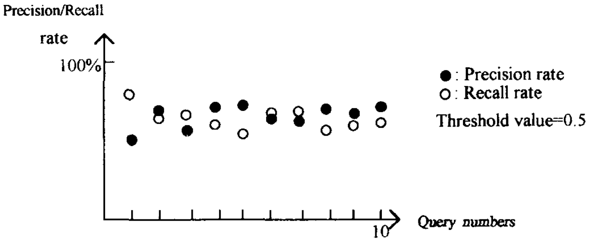


Fig. 10. The precision rate and recall rate with respect to query numbers in weighted-interval queries.

Let the ith row of the document descriptor matrix D^* be $\langle[t_{i1}^l, t_{i1}^h], [t_{i2}^l, t_{i2}^h], \cdots, [t_{in}^l, t_{in}^h]\rangle$, where $0 \leq t_{ij}^l \leq t_{ij}^h \leq 1$, and $1 \leq j \leq n$. Then, the degree of similarity between the query and the document d_i can be calculated as follows:

$$RSV_w(d_i) = \frac{\sum\limits_{\overline{q(j)} \neq \text{``-''} \text{ and } j=1,\cdots,n} S([t_{ij}^l, t_{ij}^h], [x_j^l, x_j^h])}{k} \times w_j \tag{20}$$

where the retrieval status value $RSV_w(d_i)$ indicates the degree of similarity between the query and the document $d_i, RSV_w(d_i) \in [0,1], 1 \leq i \leq m$. and $\Sigma_{j=1}^{m} w_j = 1$.

Example 5.4: Same assumptions as in Example 5.3. Let the user's query expression be represented by the query descriptor vector $\overline{q}$ shown as follows:

$$\overline{q} = \langle([0.1,0.4],0.6),—,—,([0.6,0.9],0.3),\\ ([0.5,0.7],0.1),—,—\rangle.$$

Then, the retrieval status values of the documents can be obtained shown as follows:

$$\begin{aligned} RSV_w(d_1) &= 0.625 \\ RSV_w(d_2) &= 0.41 \\ RSV_w(d_3) &= 0.75 \\ RSV_w(d_4) &= 0.69 \\ RSV_w(d_5) &= 0.44 \\ RSV_w(d_6) &= 0.64 \\ RSV_w(d_7) &= 0.82. \end{aligned}$$

If the retrieval threshold value is 0.5, then the documents d_2 and d_5 will not be retrieved due to the fact that the retrieval status values of the documents d_2 and d_5 are less than the threshold value. From the above results, we also can see that the document d_7 is the most suitable to the user's weighted interval query due to the fact that it has the largest retrieval status value.

VI. Conclusions

We have presented a knowledge-based fuzzy information retrieval method based on the transitive closure of concept matrices, where weighted queries and weighted-interval queries are allowed for document retrieval. The proposed method is more flexible than the ones presented in [6] and [11] due to theft that it has the capability to deal with interval queries and weighted-interval queries. Efficient retrieving capability and flexible user's queries are consequently provided for. We have implemented a fuzzy information retrieval system called MASTER based on the proposed method using Turbo C version 2.0 on a PC/AT, where one hundred books about computer science were acted as tested documents, and nineteen concepts were used to characterize these documents. According to [6], the performance of the implemented system can be examined through the measures of recall rate R and precision rate P defined as follows [18]:

$$R = \frac{\text{number of items retrieved and relevant}}{\text{total relevant in collection}} \times 100\% \quad (21)$$

$$P = \frac{\text{number of items retrieved and relevant}}{\text{total retrieved}} \times 100\%. \quad (22)$$

Recall rate and precision rate are defined as the capability of accepting useful documents and rejecting useless documents, respectively. In this paper, four query types (i.e., simple queries, interval queries, weighted queries, and weighted-interval queries) are examined to measure the precision rate and recall rate of the implemented system MASTER. The curves of precision rate and recall rate with respect to different threshold values are shown from Figs. 3–6.

From these curves, some phenomenon were identified. The precision rate of the simple queries is similar to the interval queries, but interval queries are more flexible than the simple queries. The precision rate of weighted queries is higher than non-weighted queries. However, users can select the query types dependent on his/her requirements when retrieving documents.

The curves of precision rate and recall rate with respect to query numbers are shown from Figs. 7–10. We can see that the experiment results of the implemented system MASTER are successful.

References

[1] S. M. Chen, "A new approach to handling fuzzy decision making problems," *IEEE Trans. Syst. Man Cyber.*, vol. 18, no. 6, pp. 1012–1016, 1988.

[2] ——, "An improved algorithm for inexact reasoning based on extended fuzzy production rules," *Cybernetics and Systems: An Int. J.*, vol. 23, no. 5, pp. 463–481, 1992.

[3] ——, "A new approach to inexact reasoning for rule-based systems," *Cybernetics and Systems: An Int. J.*, vol. 23, no. 6, pp. 561–582, 1992.

[4] S. M. Chen, J. S. Ke, and J. F. Chang, "An inexact reasoning algorithm for dealing with inexact knowledge," *Int. J. Software Engineering and Knowledge Engineering*, vol. 1, no. 3, pp. 227–244, 1991.

[5] ——, "Techniques for handling multicriteria fuzzy decision-making problems," in *Proc. 4th International Symposium on Computer and Information Sciences*, Cesme, Turkey, vol. 2, pp. 919–925, Oct. 1989.

[6] G. T. Her and J. S. Ke, "A fuzzy information retrieval system model," in *Proc. 1983 National Computer Symp.*, Taiwan, R.O.C., 1983, pp. 147–155.

[7] M. Kamel, B. Hadfield, and M. Ismail, "Fuzzy query processing using clustering techniques," *Information Processing and Management*, vol. 26, no. 2, pp. 279–293, 1990.

[8] A. Kandel, *Fuzzy Mathematical Techniques with Applications*. CA: Addison-Wesley, 1986.

[9] D. H. Kraft and D. A. Buell, "Fuzzy sets and generalized Boolean retrieval systems," *Int. J. Man-Machine Studies*, vol. 19, no. 1, pp. 45–56, 1983.

[10] C. G. Looney, "Fuzzy Petri nets for rule-based decision making," *IEEE Trans. Syst. Man Cyber.*, vol. 18, no. 6, pp. 178–183, 1988.

[11] D. Lucarella and R. Morara, "FIRST: Fuzzy information retrieval system," *J. Information Sci.*, vol. 17, pp. 81–91, 1991.

[12] T. Murai, M. Miyakoshi, and M. Shimbo, "A fuzzy document retrieval method based on two-valued indexing," *Fuzzy Sets and Systems*, vol. 30, pp. 103–120, 1989.

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

[14] T. Radechi, "Mathematical model of time effective information retrieval system based on the theory of fuzzy set," *Information Processing and Management*, vol. 13, pp. 109–116, 1977.

[15] ——, "Fuzzy set theoretical approach to document retrieval," *Information Processing and Management*, vol. 15, pp. 247–259, 1979.

[16] ——, "Generalized Boolean methods of information retrieval," *Int. J. Man-Machine Studies*, vol. 18, no. 5, pp. 409–439, 1983.

[17] R. Rousseau, "On relative indexing in fuzzy retrieval systems," *Information Processing and Management*, vol. 21, no. 5, pp. 415–417, 1985.

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

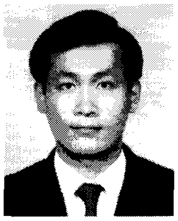
[19] V. Tahani, "A fuzzy model of document retrieval system," *Information Processing and Management*, vol. 12, pp. 177–187, 1976.

[20] J. Y. Wang and S. M. Chen, "A knowledge-based method for fuzzy information retrieval," in *Proc. First Asian Fuzzy Systems Symp.*, Singapore, Nov. 1993.

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

[22] M. Zemankova, "FIIS: A fuzzy intelligent information system," *Data Engineering*, vol. 12, no. 2, 1989.

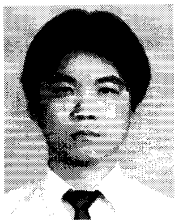
[23] R. Zwick, E. Carlstein, and D. V. Budescu, "Measures of similarity among fuzzy concepts: A comparative analysis," *Int. J. Approximate Reasoning*, vol. 1, pp. 221–242, 1987.



Shyi-Ming Chen (M'88) was born on January 16, 1960, in Taipei, Taiwan, Republic of China. He received the B.S. degree in electronic engineering from the National Taiwan Institute of Technology, Taipei, Taiwan, in 1982, and the M.S. and Ph.D. degrees in electrical engineering from the National Taiwan University, Taipei, Taiwan, in June 1986 and June 1991, respectively.

From August 1987 to July 1989 and from August 1990 to July 1991, he was with the Department of Electronic Engineering, Fu-Jen University, Taipei, Taiwan. Since August 1991, he has been an Associate Professor in the Department of Computer and Information Science at National Chiao Tung University, Hsinchu, Taiwan. His current research interests include fuzzy systems, database systems, expert systems, and artificial intelligence. He has published more than 40 papers in international journals and conferences.

Dr. Chen is a member of the IEEE Computer Society, the IEEE Systems, Man, and Cybernetics Society, the International Fuzzy Systems Association (IFSA), and the Phi Tau Phi Scholastic Honor Society.



Jeng-Yih Wang was born in Taiwan, Republic of China, on September 29, 1963. He received the Ed.B. degree in industry education from the National Taiwan College of Education, Changhwa, Taiwan, in June 1987, and the M.S. degree in Computer and Information Science from the National Chiao Tung University, Hsinchu, Taiwan, in June 1993. His current research interests include fuzzy systems, database systems, and artificial intelligence.