# A fuzzy document retrieval system using the keyword connection matrix and a learning method

Yasushi Ogawa, Tetsuya Morita and Kiyohiko Kobayashi

*Research and Development Center, RICOH Co., Ltd., 16-1 Shinei-cho, Kohoku-ku, Yokohama, 223 Japan*

Received June 1989
Revised October 1989

*Abstract:* We have already proposed a fuzzy document retrieval system using a *keyword connection matrix* to represent similarities between keywords. By using the keyword connection matrix, documents are graded according to their relevance to a user query. We have also proposed a learning method to modify the relationship values so as to reduce the difference between relationship values initially assigned using statistical information, and the user's evaluation.

There were, however, two problems with our previous method: First, we restricted compound queries to logical OR; second, our method could not accept ambiguous user judgments in the learning process. We have extended our method to solve these two problems. In the new method, compound queries composed of keywords with AND, OR and/or NOT are processed, and the learning method has been modified to allow fuzzy judgements as well as compound queries.

The new method has been implemented on a Unix workstation. Measurement of the recall and precision ratios has shown the effectiveness of our method. The averages for these ratios in several retrievals were 56% and 40% respectively, compared to 41% and 43% using the conventional crisp method. The effect of learning was also measured. The average of the recall and presision ratios were increased to 75% and 50% after 30 learning iterations.

*Keywords:* Information retrieval; fuzzy relation; indexing; learning; performance evaluation.

## 1. Introduction

Due to the increase in the number of documents in recent years, storage and retrieval of documents has become an important function in modern offices. Even though information technology has been improving rapidly, documents are poorly formatted and data management methods based on conventional Boolean principles are not powerful enough to deal with them. Thus, several approaches for a generalized Boolean retrieval model, which introduces weighting or ranking of values, have been studied [13]. Since fuzzy set theory proposed by Zadeh [15] is appropriate for dealing with ill-defined information, several researchers have used it to weight or rank values. There have been, for instance, fuzzy indexing systems [2, 11, 12], fuzzy retrieval systems using fuzzy thesauri based on either crisp indices [7] or fuzzy indices [3, 5], a fuzzy retrieval system using citations [9], and a question-answering system using topological fuzzy sets [8].

0165-0114/91/$03.50 © 1991—Elsevier Science Publishers B.V. (North-Holland)

We have already proposed a fuzzy document retrieval system using a *keyword connection matrix*[1] which represents similarities between keywords much like a thsaurus does [10]. Our method assumed crisp (Boolean) indexing, because most available databases are based on crisp indexing, and because a fuzzy thesaurus is much easier to maintain than a fuzzy index since the number of keywords is usually smaller than the number of documents. By using the keyword connection matrix, fuzzy indexing is generated from crisp indexing, and documents are graded according to their relevance to the user's query. In addition, the keyword connection matrix helps the user formulate a query appropriate to the subject he or she wishes to retrieve. Compared to other systems, an important new function in our previous method was the inclusion of a learning method. Because relationship values are initially assigned based on statistical information, they do not always accord with the user's similarity measures. To lessen this problem, we proposed a learning method for relationship values using the gradient descent method. During the learning process, the values are changed so as to reduce the difference between relationship values based on statistical information, and the user's evaluation. It should be noted that learning is carried out by evaluating, not a set of documents as a whole, but some documents selected by the user.

There were, however, two problems in our previous method. First, because we restricted compound queries to disjunctions, the user could not retrieve documents with queries such as "I need documents with both CAD and LSI" or "Are there any documents without LSI". Second, in the learning process, our method could not accept ambiguous user judgements such as "This document is almost what I wanted" or "The document is only slightly relevant". We have extended our method to solve these two problems. Compound queries composed of keywords with conjunction, disjunction and/or negation are processed in the new method by applying fuzzy intersection and complementation. The learning method has been modified to allow fuzzy judgments in addition to complex queries.

In the next section, the keyword connection matrix is explained. In Section 3, fuzzy retrieval using the keyword connection matrix, including query refinement, is explained. Section 4 describes the learning method for the keyword connection matrix and gives the formulas necessary to compute the change in value. Section 5 describes an application system implementing the proposed method on a workstation and Section 6 evaluates the method based on measurement of recall and precision ratios. Finally, our conclusions are presented in Section 7.

## 2. Keyword connection matrix

A keyword connection matrix is composed of a number of keywords and their relationships [10], where relationship values represent the conceptual similarity between two keywords. In this sense, it is a kind of a thesaurus which describes relations between keywords. In fact, the initial values are assigned in the same

[1] Originally called *keyword connection*.

way as values for related terms in a fuzzy thesaurus in [4, 6]. The keyword connection matrix is represented by a $K \times K$ matrix $\boldsymbol{W}$ where $K$ is equal to the number of keywords. Figure 1 illustrates a keyword connection matrix.

Relationship values are restricted to the interval [0, 1], where 0 indicates no relationship between two keywords and 1 indicates the strongest possible relationship. Initial relationship values are assigned based on the assumption that the more documents in which two keywords co-occur, the more they relate to each other [1, 4, 7]. $W^*_{ij}$, the initial relationship value between the $i$-th and the $j$-th keywords, is given as

$$W^*_{ij} = \begin{cases} \dfrac{N_{ij}}{N_1 + N_j - N_{ij}}, & i \neq j, \\ 1, & i = j, \end{cases} \tag{1}$$

where $N_{ij}$ is the number of documents containing both the $i$-th and the $j$-th keywords, and $N_i$ and $N_j$ are the number of documents including the $i$-th and the $j$-th keyword, respectively. This formula determines the normalized co-occurrence of two keywords in documents. Relationship values are changed during the learning process as described in Section 4. Values for cases where $j = i$ are initially set to the value 1 and never change. It should be noted that the keyword connection matrix is symmetric.

## 3. Fuzzy retrieval using the keyword connection matrix

Previous fuzzy document retrieval systems based on fuzzy set theory yielded, as a retrieval result, a fuzzy set representing how well each document matches the query. In this section, we describe a fuzzy document retrieval system using a keyword connection matrix.

### 3.1. *Overview of the fuzzy retrieval method*

The user queries the system for retrieval of a subject which he or she wants. The query contains some keywords and possibly logical operators such as AND

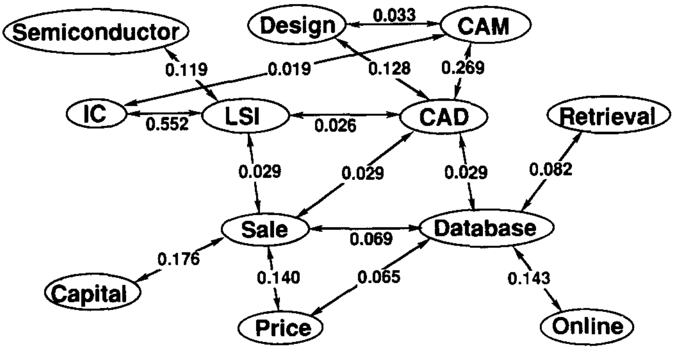

Fig. 1. Keyword connection matrix, initial values.

(conjunction), OR (disjunction) and NOT (negation). The query can be converted into conjunctive normal form, consisting of subqueries including only OR and NOT, by repeatedly applying the basic rules of Boolean algebra. A query in conjunctive normal form is written as

$$\text{Query} = \text{SubQuery}(1) \wedge \cdots \wedge \text{SubQuery}\ (N),$$

$$\text{SubQuery}(h) = K_1 \vee \cdots \vee K_{n_h} \vee \neg K_{n_h+1} \vee \cdots \vee \neg K_{n_h+m_h},$$

where $\wedge$, $\vee$ and $\neg$ represent AND, OR and NOT, $K_i$ represents the $i$-th keyword in the query. Moreover, $N \geqslant 1$ for the query, $n_h \geqslant 0$, $m_h \geqslant 0$ and $n_h + m_h \geqslant 1$ for the $h$-th subquery. The $h$-th subquery can be represented by the two sets $Q(h)^+$ and $Q(h)^-$, where $Q(h)^+$ denotes the set of keywords without NOT, and $Q(h)^-$ denotes the set of keywords with NOT. Thus, no keyword is included in both $Q(h)^+$ and $Q(h)^-$.

The retrieval result from existing ordinary methods is as follows. The retrieval result for the $h$-th subquery is given by

$$\text{SubResult}(h) = D(K_1) \cup \cdots \cup D(K_{n_h}) \cup \overline{D(K_{n_h+1})} \cup \cdots \cup \overline{D(K_{n_h+m_h})}$$

where $D(K)$ represents the set of documents indexed by keyword $K$, $D_1 \cup D_2$ is the union of two sets $D_1$ and $D_2$, and $\bar{D}$ is the complement of set $D$. The OR operator in the query requests the union of corresponding document sets, and the NOT operator requests the complement. The result can then be represented as

$$\text{Result} = \text{SubResult}(1) \cap \cdots \cap \text{SubResult}(N)$$

where $D_1 \cap D_2$ is the intersection of two sets $D_1$ and $D_2$. The AND operator in the query necessitates the intersection of corresponding sets. The result is a crisp set whose elements exactly match the query.

The methodology in a fuzzy retrieval system is just identical except that the retrieval result is a fuzzy set. $D(K)$ is usually a fuzzy set [11, 14], and the result is fuzzily generated using fuzzy union, fuzzy intersection and fuzzy complement [15]. In our system, however, $D(K)$ is originally a crisp set, and fuzzy indices are generated from the keyword connection matrix in much the same way as for a fuzzy thesaurus as in [3, 5, 7]. In this way, our system is intended to enhance the user interface of a conventional crisp retrieval system. The membership value for each document represents its *relevance as a retrieved document*. Hereafter, the membership value for the $i$-th document $d_i$ is written as $r_i$ for simplicity; i.e. $r_i = \mu_{\text{Result}}(d_i)$.

To make the system convenient for the user, documents with relevance values greater than 0 are sorted in descending order following computation of the relevance values for all the documents in the system. If the user wants a crisp result similar to that obtained from an ordinary crisp retrieval system, the system chooses the appropriate documents using either of two methods [11]: In the *thresholding by relevance value* method, an $\alpha$-cut, based on an appropriate threshold value, of the fuzzy retrieved result is presented to the user. In the *thresholding by the number of documents* method, a limit is set for the number of appropriate documents and the system selects those documents with the highest relevance values.

### 3.2. *Computation of relevance value using a keyword connection matrix*

The computation of the relevance value, in other words the membership value, is explained in this subsection. The relevance value is computed in the following three steps:

(a) Generation of fuzzy indices.

(b) Computation of relevance values for each subquery.

(c) Computation of overall relevance value.

In the following subsection, these steps will be explained in detail.

#### 3.2.1. *Generation of fuzzy indices*

Let $A_i$ denote the crisp set of keywords indexed to the $i$-th document. In our method, indexing is made fuzzy by the keyword connection matrix as follows. $R_{ij}$ representing the strength of the relationship between the $i$-th document and the $j$-th keyword is defined as

$$R_{ij} \equiv \bigoplus_{K_k \in A_i} W_{jk} \tag{2}$$

where $W_{jk}$ is the relationship value between the $j$-th and the $k$-th keywords in the keyword connection matrix. $\oplus$ denotes the algebraic sum defined by $\bigoplus_i X_i = 1 - \prod_i (1 - X_i)$. Equation (2) becomes

$$R_{ij} = 1 - \prod_{K_k \in A_i} (1 - W_{jk}).$$

$R_{ij}$ is referred to as the membership value of the $i$-th document with a fuzzy index to the $j$-th keyword.

#### 3.2.2. *Computation of relevance values for each subquery*

In fuzzy set theory, the union of two sets $A$ and $B$ is sometimes specified as

$$\mu_{A \cup B}(x) = \mu_A(x) \oplus \mu_B(x) = 1 - \mu_A(x) \cdot \mu_B(x)$$

where $\mu_A(x)$ and $\mu_B(x)$ represent the membership values of element $x$ in the fuzzy sets, $A$ and $B$. Beside MAX, which is usually applied for the union, the algebraic sum is used to compute the derivative needed for the learning method as described in the next section.

The complement of the fuzzy set $A$ is defined as

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

Relevance values for subqueries are computed using

$$r_i(h) = \Big( \bigoplus_{K_j \in Q(h)^+} R_{ij} \Big) \oplus \Big\{ \bigoplus_{K_j \in Q(h)^-} (1 - R_{ij}) \Big\}$$

$$= 1 - \Big( \prod_{K_j \in Q(h)^+} S_{ij} \Big) \Big( \prod_{K_j \in Q(h)^-} R_{ij} \Big) \tag{3}$$

where

$$S_{ij} \equiv 1 - R_{ij} = \prod_{K_k \in A_i} (1 - W_{jk}).$$

This result is identical to the result using the ordinary, crisp method when **$W$** is the unit matrix. If $Q(h)^+$ or $Q(h)^-$ is $\emptyset$ (the empty set), the relevance value is given by

$$\begin{aligned} r_i(h) &= 1 - \prod_{K_j \in Q(h)^+} S_{ij}: \quad Q(h)^- = \emptyset, \\ r_i(h) &= 1 - \prod_{K_j \in Q(h)^-} R_{ij}: \quad Q(h)^+ = \emptyset. \end{aligned} \tag{4}$$

Equation (4) is equivalent to the old formula for the query without complementation [10].

### 3.2.3. *Computation of overall relevance value*

After relevance values for all the subqueries are determined, the overall relevance value can be computed. In fuzzy set theory, the intersection of two sets is sometimes defined as

$$\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x).$$

The algebraic product is used for the intersection instead of MIN just as the algebraic sum was used for the union. Thus, the relevance value for the $i$-th document is computed by

$$r_i = \prod_{h=1}^{N} r_i(h).$$

This fuzzy result is identical to the crisp result derived by the ordinary, crisp method when **$W$** is the unit matrix.

## 3.3. *Refinement of the query*

The retrieval process is interactive and nondeterministic [13], and, thus, it is very important to help the user to make better queries. Since the keyword connection matrix represents the similarity between two keywords, it can be used to refine the query. The strength of the relationship between a keyword and the query can be computed using the keyword connection matrix.

In the process of making a query, the user can request the system to list keywords in order of the strength of their relationship to the query. The strength of the relationship, referred to as the *relevance as a keyword,* is calculated in the same way as the relevance value for a document. $T_i$, the relevance value as a keyword of the $i$-th keyword, is computed as follows:

$$T_i(h) = 1 - \left( \prod_{K_j \in Q(h)^+} W_{ij} \right) \left\{ \prod_{K_j \in Q(h)^+} (1 - W_{ij}) \right\},$$

$$T_i = \sum_{h=1}^{N} T_i(h).$$

If the user requests related keywords, relevance values for all keywords are computed and keywords are listed in descending order to their values.

## 4. Learning method for the keyword connection matrix

Initial relationship values in the keyword connection matrix do not always concur with the user's evaluation of the degree of similarity between two keywords. Therefore, relationship values can be modified using the following method so that they come closer to the user evaluation.

### 4.1. *Learning based on the gradient descent method*

Learning is carried out by evaluating, not a set of documents as a whole, but some individual documents selected by the user. That is, when the user sees a document in the result, he or she can evaluate its appropriateness for the query. An error function evaluates the difference between the initially computed relevance value and the user's judgment. In an earlier version of our method, the user's judgment was limited to either wholly relevant or wholly irrelevant [10]. However, it is sometimes difficult to judge whether a document is appropriate for the query or not. In those cases, the user may select a judgment such as "This document is almost what I wanted" or "The document is only slightly relevant". The error function $E(x)$ of the relevance value is defined as

$$E(x)=\tfrac{1}{2}(t-x)^2$$

where $t$ is between $[0, 1]$ and represents the user's judgment of the suitability of the document. The value is 0 when the user judges the document completely irrelevant, and 1 when it is judged completely relevant. The old error function described in [10] is included in the new one if $t$ is restricted to either 0 or 1.

All relationship values in the keyword connection matrix are then modified so as to reduce the value of the error function:

$$W_{mn}^{\text{new}} \leftarrow g(W_{mn}^{\text{old}}+\lambda \Delta W_{mn})$$

where $\lambda$ is a learning coefficient and $g(\,)$ is a normalizing function to ensure that the relationship value remains within $[0, 1]$:

$$g(x)=\begin{cases}1, & 1<x,\\ x, & 0\leqslant x\leqslant 1,\\ 0, & x<0.\end{cases}$$

The value of $\Delta W_{mn}$ is computed as follows using the gradient descent method:

$$\Delta W_{mn}=-\frac{\partial E_i}{\partial W_{mn}}$$

where $E_i$ is the value of the error function for the $i$-th document. Because

$$\frac{\partial E_i}{\partial W_{mn}}=-(t-r_i)\frac{\partial r_i}{\partial W_{mn}},$$

$\Delta W_{mn}$ is given by

$$\Delta W_{mn}=(t-r_i)\frac{\partial r_i}{\partial W_{mn}}.$$

Thus in order for the keyword connection matrix to learn the user's judgment, the partial derivative of the relevance value with respect to the relationship value must be computed. The formulas for computing the partial derivative are derived in the rest of the section.

### 4.2. *Computation of the partial derivative for a subquery*

First of all, we show how to compute the partial derivative of relevance values for subqueries. From (3),

$$\frac{\partial r_i(h)}{\partial W_{mn}} = -\left(\prod_{K_j \in Q(h)^-} R_{ij}\right)\frac{\partial}{\partial W_{mn}}\left[\prod_{K_j \in Q(h)^+} S_{ij}\right] - \left(\prod_{K_j \in Q(h)^+} S_{ij}\right)\frac{\partial}{\partial W_{mn}}\left[\prod_{K_j \in Q(h)^-} R_{ij}\right].$$

From the definition of $S_{ij}$,

$$\frac{\partial}{\partial W_{mn}}\left[\prod_{K_j \in Q(h)^+} S_{ij}\right] = \begin{cases} -\displaystyle\prod_{\substack{K_j \in Q(h)^+ \\ K_k \in A_i \\ \text{s.t.}\,(j,k)\neq(m,n)}} (1 - W_{jk}), & K_n \in A_i \text{ and } K_m \in Q(h)^+, \\ 0, & \text{otherwise,} \end{cases}$$

and from the definition of $R_{ij}$,

$$\frac{\partial}{\partial W_{mn}}\left[\prod_{K_j \in Q(h)^-} R_{ij}\right] = \begin{cases} \left\{\displaystyle\prod_{\substack{k_k \in A_i \\ \text{s.t.}\,k\neq n}} (1 - W_{jk})\right\}\left(\displaystyle\prod_{\substack{k_j \in Q(h)^+ \\ \text{s.t.}\,j\neq m}} R_{ij}\right), & K_n \in A_i \text{ and } K_m \in Q(h)^-, \\ 0, & \text{otherwise.} \end{cases}$$

When $W_{mn} \neq 1$, we get the following relationships:

$$\prod_{\substack{K_j \in Q(h)^+ \\ K_k \in A_i \\ \text{s.t.}\,(j,k)\neq(m,n)}} (1 - W_{jk}) = \frac{\prod_{K_j \in Q(h)^+} S_{ij}}{1 - W_{mn}}, \tag{5}$$

$$\prod_{\substack{K_k \in A_i \\ \text{s.t.}\,k\neq n}} (1 - W_{mk}) = \frac{1 - R_{im}}{1 - W_{mn}}. \tag{6}$$

When $R_{im} \neq 0$, we get another relationship:

$$\prod_{\substack{K_j \in Q(h)^- \\ \text{s.t.}\,j\neq m}} R_{ij} = \frac{\prod_{K_j \in Q(h)^-} R_{ij}}{R_{im}}. \tag{7}$$

When $R_{im} = 0$ (i.e. $S_{im} = 1$), from the definition of $R_{ij}$,

$$W_{mk} = 0, \text{ i.e. } W_{mk} \neq 1, \quad \text{for all } K_k \in A_i. \tag{8}$$

On the other hand, when $W_{mn} = 1$ and $K_n \in A_i$,

$$R_{im} = 1 - \prod_{K_k \in A_i} (1 - W_{mk}) = 1. \tag{9}$$

We have to consider the following three cases given the assumption that there is no keyword in both $Q(h)^+$ and $Q(h)^-$:

(a) $K_m \in Q(h)^+$,
(b) $K_m \in Q(h)^-$,
(c) $K_m \notin Q(h)^+$ and $K_m \notin Q(h)^+$.

### 4.2.1. *When $K_m \in Q(h)^+$*

Because $K_m \notin Q(h)^-$ in this case,

$$\frac{\partial}{\partial W_{mn}}\left[\prod_{K_j \in Q(h)^+} S_{ij}\right] = \begin{cases} -\displaystyle\prod_{\substack{k_j \in Q(h)^+ \\ K_k \in A_i \\ \text{s.t.}\,(j,k)\neq(m,n)}} (1 - W_{jk}), & K_n \in A_i, \\ 0, & \text{otherwise}, \end{cases}$$

$$\frac{\partial}{\partial W_{mn}}\left[\prod_{K_j \in Q(h)^-} R_{ij}\right] = 0.$$

Thus the partial derivative can be computed as follows:

$$\frac{\partial r_i}{\partial W_{mn}} = \begin{cases} \left(\displaystyle\prod_{K_j \in Q(h)^-} R_{ij}\right)\left\{\displaystyle\prod_{\substack{K_j \in Q(h)^+ \\ K_k \in A_i \\ \text{s.t.}\,(j,k)\neq(m,n)}} (1 - W_{jk})\right\}, & K_n \in A_i, \\ 0, & \text{otherwise}. \end{cases} \tag{10}$$

When $W_{mn} \neq 1$, however, the formula is simplified. First, from (3),

$$\left(\prod_{K_j \in Q(h)^+} S_{ij}\right)\left(\prod_{K_j \in Q(h)^-} R_{ij}\right) = 1 - r_i(h). \tag{11}$$

Using (5) and (11),

$$\left(\prod_{K_j \in Q(h)^-} R_{ij}\right)\left\{\prod_{\substack{K_j \in Q(h)^+ \\ K_k \in A_i \\ \text{s.t.}\,(j,k)\neq(m,n)}} (1 - W_{jk})\right\} = \left(\prod_{K_j \in Q(h)^-} R_{ij}\right)\frac{\displaystyle\prod_{K_j \in Q(h)^+} S_{ij}}{1 - W_{mn}} = \frac{1 - r_i(h)}{1 - W_{mn}}.$$

Therefore, (10) becomes

$$\frac{\partial r_i(h)}{\partial W_{mn}} = \begin{cases} \dfrac{1 - r_i(h)}{1 - W_{mn}}, & K_n \in A_i \text{ and } W_{mn} \neq 1, \\ \left\{\displaystyle\prod_{\substack{K_j \in Q(h)^+ \\ K_k \in A_i \\ \text{s.t.}\,(j,k)\neq(m,n)}} (1 - W_{jk})\right\}\left(\displaystyle\prod_{K_j \in Q(h)^-} R_{ij}\right), & K_n \in A_i \text{ and } W_{mn} = 1, \\ 0, & \text{otherwise}. \end{cases} \tag{12}$$

4.2.2. *When $K_m \in Q(h)^-$*

The partial derivative is computed as follows in the same way as when $K_m \in Q(h)^+$. Since $K_m \notin Q(h)^+$,

$$\frac{\partial r_i(h)}{\partial W_{mn}} = \begin{cases} -\left(\prod_{K_j \in Q(h)^+} S_{ij}\right)\left\{\prod_{\substack{K_k \in A_i \\ \text{s.t.}\, k \neq n}} (1-W_{mk})\right\}\left(\prod_{\substack{K_j \in Q(h)^- \\ \text{s.t.}\, j \neq m}} R_{ij}\right) & K_n \in A_i, \\ 0, & \text{otherwise.} \end{cases}$$

If $K_n \in A_i$ and $W_{mn} \neq 1$ and $R_{im} \neq 0$, the next relation is derived using relations (6), (7), and (11):

$$\left(\prod_{K_j \in Q(h)^+} S_{ij}\right)\left\{\prod_{\substack{K_k \in A_i \\ \text{s.t.}\, k \neq n}} (1-W_{mk})\right\}\left(\prod_{\substack{K_j \in Q(h)^- \\ \text{s.t.}\, j \neq m}} R_{ij}\right) = \left(\prod_{K_j \in Q(h)^+} S_{ij}\right)\frac{1-R_{im}}{1-W_{mn}}\frac{\prod_{K_j \in Q(h)^-} R_{ij}}{R_{im}}$$
$$= \frac{1-R_{im}}{R_{im}}\frac{1-r_i(h)}{1-W_{mn}}.$$

If $K_n \in A_i$ and $W_{mn} = 1$, then $R_{im} \neq 0$ from equation (9). Equation (7) can be used to yield

$$\left(\prod_{K_j \in Q(h)^+} S_{ij}\right)\left\{\prod_{\substack{K_k \in A_i \\ \text{s.t.}\, k \neq n}} (1-W_{mk})\right\}\left(\prod_{\substack{K_j \in Q(h)^- \\ \text{s.t.}\, j \neq m}} R_{ij}\right) = (1-r_i(h))\left\{\prod_{\substack{K_k \in A_i \\ \text{s.t.}\, k \neq n}} (1-W_{mk})\right\}.$$

Here the relation $R_{im} = 1$ is used. Furthermore, if $K_n \in A_i$ and $R_{im} = 0$, the following equation results from the definition of $R_{ij}$ because $W_{mn} = 0$:

$$\left(\prod_{K_j \in Q(h)^+} S_{ij}\right)\left\{\prod_{\substack{K_k \in A_i \\ \text{s.t.}\, k \neq n}} (1-W_{mk})\right\}\left(\prod_{\substack{K_j \in Q(h)^- \\ \text{s.t.}\, j \neq m}} R_{ij}\right) = \left(\prod_{K_j \in Q(h)^+} S_{ij}\right)\left(\prod_{\substack{K_j \in Q(h)^- \\ \text{s.t.}\, j \neq m}} R_{ij}\right).$$

The resulting formula is

$$\frac{\partial r_i(h)}{\partial W_{mn}} = \begin{cases} -\frac{1-R_{im}}{R_{im}}\frac{1-r_i(h)}{1-W_{mn}}, & K_n \in A_i \text{ and } W_{mn} \neq 1 \text{ and } R_{im} \neq 0, \\ -(1-r_i(h))\left\{\prod_{\substack{K_k \in A_i \\ \text{s.t.}\, k \neq n}} (1-W_{mk})\right\}, & K_n \in A_i \text{ and } W_{mn} = 1, \\ -\left(\prod_{K_j \in Q(h)^+} S_{ij}\right)\left(\prod_{\substack{K_j \in Q(h)^- \\ \text{s.t.}\, j \neq m}} R_{ij}\right), & K_n \in A_i \text{ and } R_{im} = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

4.2.3. *When $K_m \notin Q(h)^+$ and $K_m \notin Q(h)^-$*

In this case,

$$\frac{\partial}{\partial W_{mn}}\left[\prod_{K_j \in Q(h)^-} R_{ij}\right] = \frac{\partial}{\partial W_{mn}}\left[\prod_{K_j \in Q(h)^+} S_{ij}\right] = 0$$

and the result is always very simple:

$$\frac{\partial r_i(h)}{\partial W_{mn}} = 0. \tag{14}$$

### 4.3. *Computation of the partial derivative for a query*

The partial derivative of the relevance value for a query is computed using

$$\frac{\partial r_i}{\partial W_{mn}} = \sum_{h=1}^{N} \left\{ \frac{\partial r_i(h)}{\partial W_{mn}} \prod_{\substack{k=1 \\ \text{s.t. } k \neq h}}^{N} r_i(k) \right\}.$$

However the computation of the value becomes much easier as follows.

#### 4.3.1. *When* $r_i(h) \neq 0$

In this case, because $r_i(h) \neq 0$ for all $h$ $(1 \leqslant h \leqslant N)$,

$$\prod_{\substack{k=1 \\ \text{s.t. } k \neq h}}^{N} r_i(k) = \frac{r_i}{r_i(h)}.$$

Therefore,

$$\frac{\partial r_i}{\partial W_{mn}} = \sum_{h=1}^{N} \left\{ \frac{\partial r_i(h)}{\partial W_{mn}} \sum_{\substack{k=1 \\ \text{s.t. } k \neq h}}^{N} r_i(k) \right\} = \sum_{h=1}^{N} \left\{ \frac{\partial r_i(h)}{\partial W_{mn}} \frac{r_i}{r_i(h)} \right\}$$

$$= r_i \sum_{h=1}^{N} \left\{ \frac{\partial r_i(h)}{\partial W_{mn}} \frac{1}{r_i(h)} \right\}. \tag{15}$$

#### 4.3.2. *When* $r_i = 0$

In this case, the formulas differ in the number of $h$ which make $r_i(h) = 0$. When the number of $h$ is 1,

$$\frac{\partial r_i}{\partial W_{mn}} = \sum_{h=1}^{N} \left\{ \frac{\partial r_i(h)}{\partial W_{mn}} \prod_{\substack{k=1 \\ \text{s.t. } k \neq h}}^{N} r_i(k) \right\} = \frac{\partial r_i(h^*)}{\partial W_{mn}} \left\{ \prod_{\substack{k=1 \\ \text{s.t. } k \neq h^*}}^{N} r_i(k) \right\} \tag{16}$$

where $h^*$ makes $r_i(h^*) = 0$. When the number of such $h$ is more than 1,

$$\frac{\partial r_i}{\partial W_{mn}} = 0. \tag{17}$$

The partial derivative of the relevance value with respect to the relationship value can be computed using (12)–(17). From these formulas, $\Delta W_{mn} = 0$ when $K_m \notin Q$ or $K_n \notin A_i$ where

$$Q \equiv \bigcup_{h=1}^{N} \{Q(h)^{+} \cup Q(h)^{-}\}.$$

Thus, only the partial derivative for the pairs $(m, n)$ of $K_m \in Q$ and $K_n \in A_i$ need be computed.

## 5. Application system

We have developed an application system based on the proposed method on a Unix workstation. As illustrated in Figure 2, the system is composed of three parts; an OCR (Optical Character Recognition) block, a data storage block and a Unix workstation. The OCR block, used for data input, consists of a 400 dpi image scanner and character recognition system for Japanese and/or English text. The optical disk unit, used for data storage, can store 0.8GB data in a 5.25" optical disk. These two blocks are connected to the workstation by SCSI.

The fuzzy document retrieval is carried out on the workstation. There are three main functions – *fuzzy document retrieval, related keyword retrieval* and *keyword connections learning* – described in the previous sections. Fuzzy document retrieval retrieves an appropriate number of documents of adequate quality. Related keyword retrieval offers a related term to the user to aid in refining queries. Keyword connections learning improves retrieval performance to come closer to the user's evaluation. As a result of iterative learning, the keyword connection matrix becomes more sensitized to the subject of the query which was presented by the user.

In addition to these functions, the *automatic indexing block* is implemented. Manual keyword indexing in a small database was relatively easy; however, some kind of indexing assistance is necessary to deal with large-scale text databases in realistic applications. Automatic indexing is carried out using morpheme analysis, particle analysis, frequency counting and stop-word elimination.

## 6. Performance evaluation

### 6.1. *Evaluation model*

Performance of information retrieval systems is normally measured with the following two scores, the *recall ratio R* and the *precision ratio P*:

$$R = \frac{\#(\text{Relevant documents in the result})}{\#(\text{All relevant documents})},$$

$$P = \frac{\#(\text{Relevant documents in the result})}{\#(\text{Documents in the result})},$$

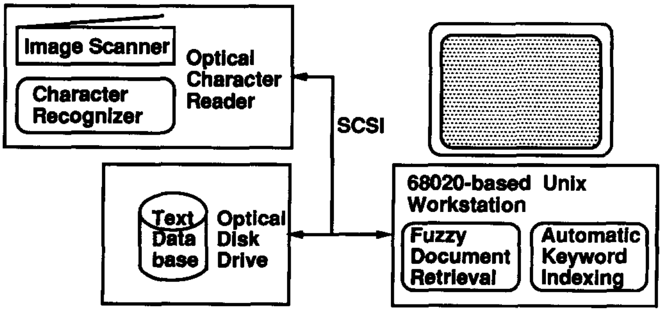

Fig. 2. Application system configuration.

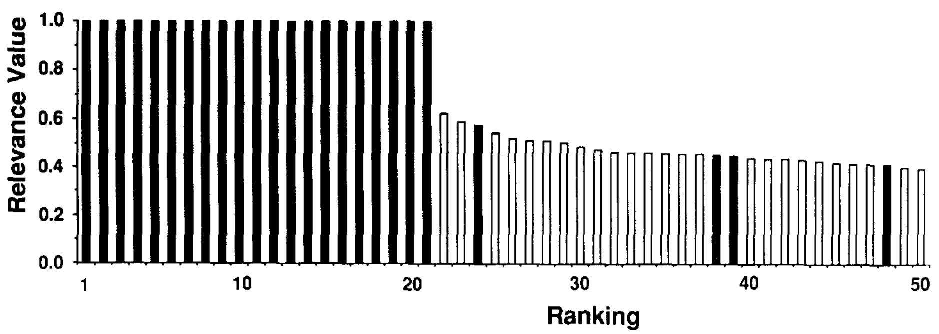

Fig. 3. Retrieval result, ranking vs. relevance value.

where #($X$) is the number of $X$. The recall ratio represents how many of the relevant documents are retrieved, while the precision ratio represents how many of the retrieved documents are relevant.

To calculate these ratios, the relevant documents for each of several queries are selected by hand by four experimenters. Subsequently, the documents judged relevant to the same queries are retrieved by the system. The recall and precision ratios are then calculated. However, because the recall and precision ratios are originally defined for crisp retrieval, we have to convert fuzzy retrieval results into crisp ones. In our performance evaluation, we adopt the thresholding by relevance value method explained in Section 3.

A text database including 10 000 documents has been created to allow evaluation. Using the automatic indexing function, 1950 keywords were extracted from the documents, and 71 963 keyword connections were calculated using formula (1).

### 6.2. *Performance results: Comparison with crisp method*

Figure 3 shows the retrieval result for a query with a single keyword, *CAD*. Shaded bars correspond to relevant documents and plain bars correspond to irrelevant documents. It should be noted that only the top 50 documents are shown in this figure. Figure 4 shows the relationship between the threshold value

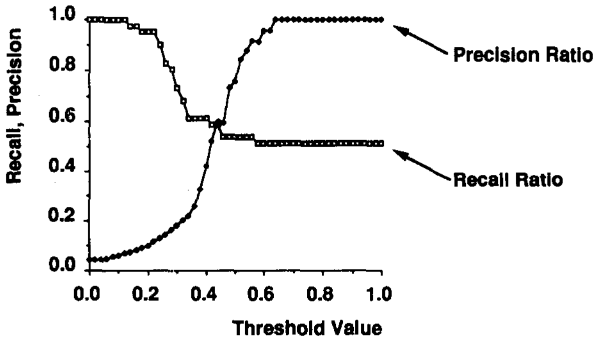

Fig. 4. Relationship between threshold value and recall/precision ratios (Query = *CAD*).

and the two ratios for the same query. The recall and precision ratios are inversely related as can be expected from their definitions.

Next, we compare our new method with the conventional crisp retrieval method. Normally, the optimal threshold value, which raises both the recall and precision ratios, varies according to the query as can be seen from Figures 4 and 5. Here, Figure 5 are results for a query *CAD* ∧ ¬*Database.*[2] Thus, a fixed threshold is inappropriate and must be determined dynamically. In the following, the threshold value $\alpha$ is computed using

$$\alpha = \mu \times \frac{\text{Total sum of relevance values}}{\#(\text{Documents with non-zero values of relevance})} \tag{18}$$

where $\mu$ is the threshold coefficient set to 1.6. We examine four single keyword queries: *CAD, LSI, Database* and *Sales,* as well as several queries generated by combining two keywords with AND, OR or both AND and NOT. The average of the recall and precision ratios for such queries are referred to as $R_{\text{avg}}$ and $P_{\text{avg}}$, respectively. $R_{\text{avg}} = 0.56$ and $P_{\text{avg}} = 0.40$ using the fuzzy retrieval method. Compared to the crisp method where $R_{\text{avg}} = 0.41$ and $P_{\text{avg}} = 0.43$, the recall ratio was greatly improved although the precision ratio was nearly the same.

### 6.3. *Performance results*: *The effectiveness of learning*

Next, we evaluate the effectiveness of the learning method. A single experiment includes a number of learning cycles. In each cycle, only the documents with a relevance value higher than the threshold value, determined using formula (18), are retrieved. To simplify the experiments, a document is evaluated to be either wholly relevant or wholly irrelevant. The learning coefficient $\lambda$ is fixed at 0.02 and the thresholding coefficient $\mu$ at 1.6, throughout the experiments.

Figure 6 shows the results after 30 learning cycles when the query is the single keyword *CAD.* Shaded bars again correspond to relevant documents. Compared

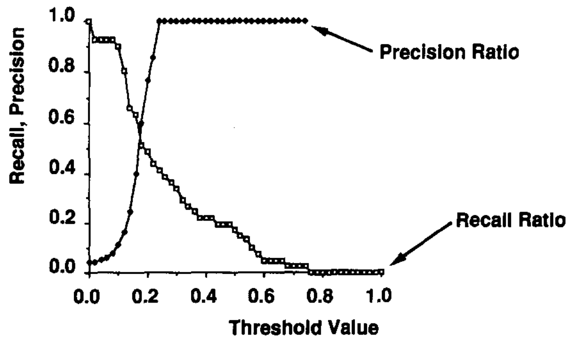

Fig. 5. Relationship between threshold value and recall/precision ratios (Query = *CAD*&¬*Database*).

[2] In Figure 5, the precision ratio cannot be computed for threshold values over 0.76, because the highest relevance value is 0.756 and thus there is no documents in the results for such a high threshold.

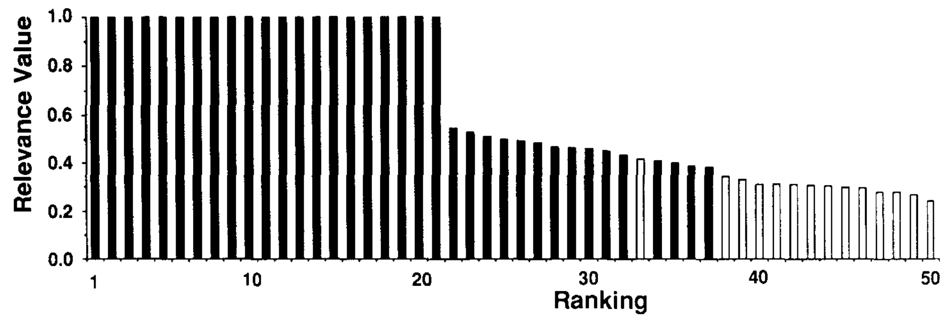

Fig. 6. Retrieval result after 30 learning cycles, ranking vs. relevance value.

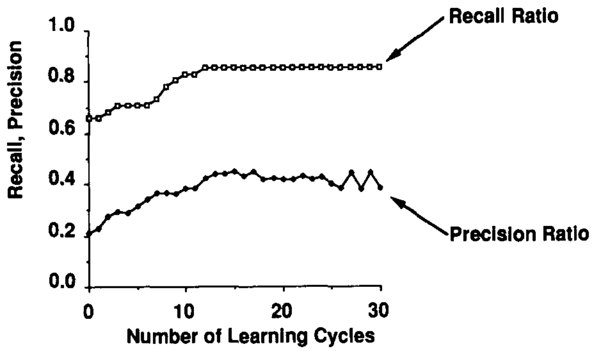

Fig. 7. Effectiveness of learning, number of learning cycles vs. recall/precision ratios.

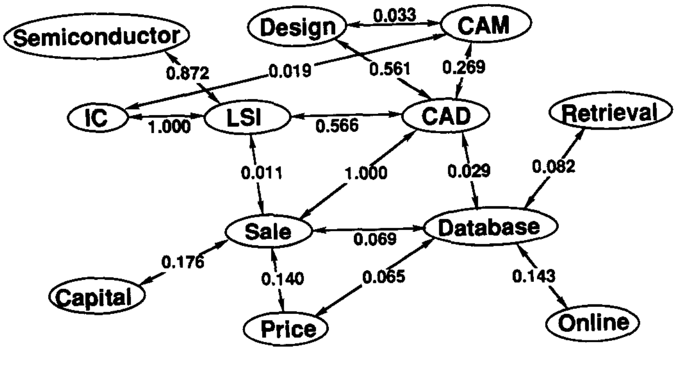

Fig. 8. Keyword connection matrix, after 30 learning cycles.

to the retrieval result before learning shown in Figure 3, the relevant documents receive a higher ranking after learning. Figure 7 shows the increase in the recall and precision ratios as a result of learning. The keyword connection matrix after learning is illustrated in Figure 8, where one can find some changes with regard to the initial relationship values in Figure 1.

We also measure the average of the ratios, $R_{avg} = 0.75$ and $P_{avg} = 0.50$. The recall ratio was much better than prior to learning and thus far superior to that of the crisp method. The precision ratio was also improved with learning and exceeded that of the crisp method. These results confirm the effectiveness of the learning method.

## 7. Conclusion

In this paper, we extended a fuzzy document retrieval method using the keyword connection matrix. Any queries composed of keywords and AND, OR and/or NOT are processed in the new method, and the learning method is modified for fuzzy judgments as well as for complex queries. The new method has been implemented on a Unix workstation in an application system, consisting of an OCR block, a data storage block, an automatic keyword indexing block and a fuzzy retrieval block.

The measurement of the recall ratio and precision ratio verified the effectiveness of the proposed method. Table 1 summarizes the performance, compared to conventional crisp method. Even without learning, the recall ratio increased although the precision ratio decreased slightly. Both ratios were greatly improved after incorporation of the learning method.

Table 1. Comparison of the proposed method using keyword connection matrix and the crisp method

| | Recall ratio | Precision ratio |
|---|---|---|
| Crisp method | 41% | 43% |
| Proposed method (before learning) | 56% | 40% |
| Proposed method (after learning) | 75% | 50% |

**Acknowledgements**

We gratefully acknowledge the help of K. Fujita, S. Imago, M. Mochinushi in the development of the automatic indexing function and the support of the R&D center OCR group with the character recognition system.

## References

[1] L.B. Doyle, Indexing and abstracting by association, *Amer. Documentation* **13** (1962) 378–390.

[2] D.H. Kraft and D.A. Buell, Fuzzy sets and generalized Boolean retrieval systems, *Internat. J. Man-Machine Stud.* **19** (1983) 45–56.

[3] S. Miyamoto, Information retrieval based on fuzzy associations, *Fuzzy Sets and Systems* **38** (1990) 191–205.

[4] S. Miyamoto, T. Miyake, and K. Nakayama, Generation of a pseudothesaurus for information retrieval based on co-occurences and fuzzy set operations, *IEEE Trans. Systems Man Cybernet.* **13**(1) (1983) 62–70.

[5] S. Miyamoto and K. Nakayama, Fuzzy information retrieval based on a fuzzy pseudothesaurus, *IEEE Trans. Systems Man Cybernet.* **16**(2) (1986) 278–282.

[6] T. Murai, M. Miyakoshi and M. Shimbo, A modeling of search oriented thesaurus use based on multivalued logical inference, *Inform. Sci.* **43** (1988) 185–212.

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

[8] K. Nakamura and S. Iwai, Topological fuzzy sets as a quantitative description of analogical inference and its application to question-answering system for information retrieval, *IEEE Trans. Systems Man Cybernet.* **12**(2) (1982) 193–204.

[9] K. Nomoto, S. Wakayama, T. Kirimoto and M. Kondo, Fuzzy retrieval system based on citations, in: *Proc. of the 2nd IFSA Congress* (1987) 723–726.

[10] Y. Ogawa, T. Morita and K. Kobayashi, Fuzzy document retrieval system and its learning method based on the keyword connection, in: *Proc. Int. Workshop on Fuzzy System Applications* (1988) 143–144.

[11] T. Radecki, Outline of a fuzzy logic approach to information retrieval, *Internat. J. Man-Machine Stud.* **14** (1981) 169–178.

[12] T. Radecki, Generalized Boolean methods of information retrieval, *Internat. J. Man-Machine Stud.* **18** (1983) 407–439.

[13] G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval* (McGraw-Hill, New York, 1983).

[14] R.R. Yager, A logical on-line bibliographic searcher: an application of fuzzy sets, *IEEE Trans. Systems Man Cybernet.* **10**(1) (1980) 51–53.

[15] L.A. Zadeh, Fuzzy sets, *Inform. and Control* **8** (1965) 338–353.