Fuzzy Sets and Systems 38 (1990) 191–205
North-Holland

Information retrieval based on fuzzy associations

S. Miyamoto

Institute of Information Sciences and Electronics, University of Tsukuba, Ibaraki 305, Japan

Received May 1988
Revised December 1988

Abstract: The aim of the present paper is to propose a fuzzy set model for information retrieval and to develop methods and algorithms for fuzzy information retrieval based on the fuzzy set model. A process of information retrieval is represented as a diagram that consists of three components. Each component has its inherent fuzziness. As typical examples for describing the three components, we consider a fuzzy association as a generalization of a fuzzy thesaurus for the first component, a fuzzy inverted index for the second component, and a fuzzy filter for the third component. Efficient algorithms for fuzzy retrieval on large scale bibliographic databases are developed. The significance of the present method is that current techniques in researches of bibliographic databases without fuzzy sets are studied in the framework of fuzzy sets and their implications are made clear using the model herein.

Keywords: Information storage and retrieval; fuzzy associations; algorithms.

1. Introduction

Research in fuzzy information retrieval has been concentrated on theoretical aspects such as processing of fuzzy queries, e.g., [7, 11, 16, 17, 25], mathematical properties of a fuzzy thesaurus [18, 19], generalization of implication operators as fuzzy relations [10], and so on. While different types of fuzziness in information retrieval have been studied in detail, practical considerations are still rare. For example, information retrieval of bibliographic databases needs processing of a large number of articles. Nevertheless, there have been few studies that discuss efficient algorithms for a large set of documents and user interfaces in an environment of a fuzzy retrieval. Hardware for information retrieval was not well-developed for realizing fuzzy information retrieval several years ago. Now, however, computers and the peripheral devices become faster and faster, which will enable application of theories of fuzzy information retrieval for practical databases.

In the present paper, we consider two aspects of fuzzy information retrieval, that is, efficient algorithms for fuzzy retrieval and a block diagram representation of fuzzy retrieval process. These two aspects have not been studied in detail by other researchers. In particular, the block diagram representation provides a new way of describing different functions in fuzzy retrieval. The diagram has three

components: the first component represents fuzzy thesaurus; the second component is a fuzzy index; and the third component is called here a fuzzy filter for fuzzy retrieval. Functions of these components are discussed in detail. A fuzzy set model for fuzzy thesaurus is given and an algorithm for generating a fuzzy thesaurus of a large scale is developed. Two kinds of efficient algorithms using a fuzzy thesaurus and a fuzzy index are discussed. As an example of the third component, a linear diagonal filter that expresses a user's preference on an index is considered.

2. Three components in information retrieval

Let $D = \{d_1, d_2, \ldots, d_n\}$ be a finite set of documents for retrieval. Each document has several descriptors as indexes of the documents. Descriptors may be keywords, citation indexes, or other kinds of indexes. A set of descriptors is denoted by $W = \{w_1, w_2, \ldots, w_m\}$. For the most part we assume that W is a set of keywords, although W may stand for another kind of a set of descriptors, as will be explained below. Correspondence between a document and descriptors in W is given by a function $T : D \rightarrow [0, 1]^W$. For a given $d \in D$, $T(d)$ means a subset of descriptors (keywords) in W indexed to the document d. $T(d)$ may be crisp or fuzzy. Therefore we assume that $T(d)$ is fuzzy in general. The inverse of T is denoted as U ($U = T^{-1}$). It is clear that for a given $w \in W$, $U(w)$ means documents that have the keyword w. T and U are represented by fuzzy relations or matrices. We do not distinguish a fuzzy relation and its matrix representation, as the equivalence between a fuzzy relation defined on a pair of finite sets and a matrix representation associated to it is trivial. In the same way, a fuzzy set $q = \sum q_i/w_i$ of W is represented by a vector $q = (q_1, q_2, \ldots, q_m)^{\mathrm{T}}$. We do not distinguish between the fuzzy set q and the vector representation q by the same reason.

Suppose that a query is in general a fuzzy set q of W, a response is a fuzzy set r of D, and the relation between q and r is described by the fuzzy relation U defined on $D \times W$. We call U a fuzzy index here. Let $m_A(\cdot)$ be the membership of a fuzzy set in general. Then, the relation between the query and the response is given by (see Section 4 for the details)

$$r = \sum_j \max \min[U(d_i, w_j), m_q(w_j)]/d_i. \tag{1}$$

Note that when we put $m_q(w_j) = q_j$, $m_r(d_i) = r_i$, and $u_{ij} = U(d_i, w_j)$, we represent the two fuzzy sets as vectors $q = (q_1, q_2, \ldots, q_m)^{\mathrm{T}}$ and $r = (r_1, r_2, \ldots, r_n)^{\mathrm{T}}$, and the fuzzy relation as a matrix $U = (u_{ij})$. Then the equation (1) is written as

$$r = Uq \tag{2}$$

using fuzzy algebra, that is, an algebra where addition is maximum ($a + b = \max(a, b)$) and multiplication is minimum ($ab = \min(a, b)$). Since we do not distinguish a fuzzy relation and its matrix representation, the equation (2) is regarded as an abbreviation of (1). (Prade and Testemale [17] consider the same type of formula using a possibility degree.)

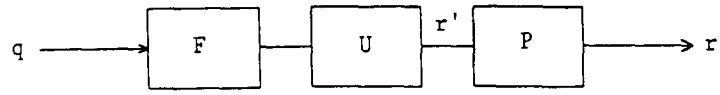

Fig. 1. Representation of an information retrieval process by a block diagram with three components.

Let us consider a block diagram shown as Figure 1. This diagram has three components shown by the symbols F, U, and P. First, a given query may not be adequate for the indexes of the database, and therefore the input query is expanded to include synonyms and related keywords that are more appropriate. Thus, a typical example of the function F of the first component is a fuzzy thesaurus. The second component U shows fuzzy index already described above. The last component is called here a fuzzy filter for information retrieval. Although various forms of the fuzzy filters may be considered, we consider only a simple linear filter here, which will be described in Section 5. Meanwhile, we note that a linear filter here expresses a user's preference in the next examples:

(a) Find *recent* documents that have keyword w.

(b) Find documents that have keyword w and are *relevant to my field of interest.*

The italicized parts are realized by linear filters.

Now, consider an equation that corresponds to the diagram in Figure 1:

$$r = PUFq. \tag{3}$$

This equation gives a mathematical description of fuzzy information retrieval. Here q and r are fuzzy sets represented by vectors; P, U, and F are fuzzy relations represented by matrices. Note that fuzzy algebra (maximum for addition and minimum for multiplication) is used. If we do not use P and F, then the relation (3) is reduced to the equation (2). The equation (3) is justified in Sections 4 and 5.

The three components F, U, and P may be studied in the ordinary framework of crisp retrieval. However, as we will see later, the framework of fuzzy sets is natural and adequate for considering problems in information retrieval. From the next section we consider how fuzziness is introduced and studied as each component of the diagram in Figure 1.

3. Fuzziness in a thesaurus: first component

A thesaurus in information retrieval is a special type of a dictionary in which for a title keyword, associated keywords are given in terms of a few categories of the association. Here we deal with three categories: RT (related terms), NT (narrower terms), and BT (broader terms). (See, e.g., [12] for details.) These categories are represented as binary relations between a pair of keywords $v, w \in W$.

We assume that w is a title word and v is an associated word to w. If v is in the category NT, then the meaning of v is narrower than that of w. If v is in BT, then the meaning of v is broader than that of w. If v is in RT, then the meaning of v is somehow related to that of w. These relationships are represented by three binary

relations N, B, and R: $N(v, w) = 1$ if v is in the category NT for the title word w, and $N(v, w) = 0$ otherwise; $B(v, w) = 1$ if v is in BT, and $B(v, w) = 0$ otherwise; $R(v, w) = 1$ if v is in RT, and $R(v, w) = 0$ otherwise. Moreover, it is natural to assume that $B(v, w) = N(w, v)$ and $R(v, w) = R(w, v)$. That is, we assume that B is the inverse relation of N and the relation R is symmetric. Therefore we consider only the two relations R and N from now on.

It appears to be easy to consider conceptually a fuzzy thesaurus as a generalization of R and N to fuzzy relations. It is necessary, however, to show how a fuzzy thesaurus is constructed and used in fuzzy information retrieval.

Methods of automatic generation of thesauri have been studied by a number of researchers (e.g., [20, 23]). A well-known technique for this is based on counting frequencies of simultaneous occurrences of pairs of keywords in a set of documents. This technique is closely related to a mathematical model based on fuzzy sets. Moreover, we have a better interpretation of this technique of automatic generation of thesauri using fuzzy sets. For showing this, let us introduce a fuzzy set model.

Let $C = \{c_1, c_2, \ldots, c_p\}$ be a finite set of concepts where each c_i, $i = 1, \ldots, p$, represents a unit of concept. Let $h : W \rightarrow [0, 1]^C$ be a fuzzy set valued function which maps each keyword to its corresponding concepts as a fuzzy set in C. That is, $h(w)$, $w \in W$, is concepts of the word w. The set C and the function h are introduced in an abstract manner: we do not find C and h in a real world. For an application of this model, we replace the set C and the function h by some substitutes, as we will see below.

We define two fuzzy relations $R(v, w)$ and $N(v, w)$ using the set C and the function h as follows:

$$R(v, w) = \frac{|h(v) \cap h(w)|}{|h(v) \cup h(w)|}, \tag{4}$$

$$N(v, w) = \frac{|h(v) \cap h(w)|}{|h(v)|}, \tag{5}$$

where $|A|$ for a fuzzy set A means the cardinality of A. (Sometimes $|A|$ is written as $\sum$ Count(A). See [8, 28]. Note also that the relations (4) and (5) are introduced in a different context of similarity measure and inclusion measure of a pair of fuzzy sets in [2].)

The meaning of R and N is clearly explained by Figure 2. Namely, $R(v, w)$ is

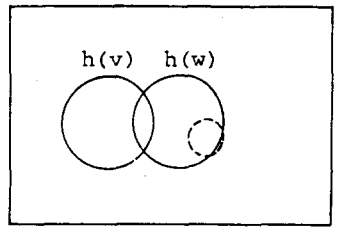

Fig. 2. Concepts of the words v and w in the set C.

the 'area' of intersection of $h(v)$ and $h(w)$ over the area of the union of $h(v)$ and $h(w)$; $N(v, w)$ is the area of the intersection of the two fuzzy sets over the area of $h(v)$. If $h(v) \subseteq h(w)$, that is, the concepts of v are included in the concepts of w, then $N(v, w) = 1$. This means that the relation N expresses narrower terms. On the other hand, $R(v, w) = 1$ if and only if $h(v) = h(w)$. It is also easy to see that $R(v, w) = R(w, v)$, whereas $N(v, w) \neq N(w, v)$ in general. The difference between R and N is illustrated by the area surrounded by the dashed curve in Figure 2: if this area means $h(v)$, then $N(v, w) = 1$ but $R(v, w) \ll 1$. From the above property it is natural to call the above fuzzy relations a fuzzy thesaurus. The relation R defined by (4) and N defined by (5) can be considered as a generalization of RT and NT in the usual sense, respectively. Note that if we apply alpha-cuts on R and N, we will have a pair of binary relations that imply RT and NT in the usual form.

Application of the above model for automatic generation of a thesaurus needs specification of the set C. To specify C precisely is of course impossible. Therefore we replace the set C by another set that is available for practical use. This replacement implies that we allow the latter set as a substitute for the set C.

Current studies of automatic generation of fuzzy thesauri use a set D of documents for counting simultaneous occurrences. Therefore, we use the set $D = \{d_1, d_2, \ldots, d_n\}$ as a substitute for C. The function $h : W \to [0, 1]^D$ is naturally defined in terms of frequencies of occurrence of $w \in W$ in the document $d \in D$. Let h_{ik} be the frequency of occurrence of w_i in the document d_k. If we take

$$h(w_i) = h_{i1}/d_1 + h_{i2}/d_2 + \cdots + h_{in}/d_n$$

then the values of membership may be outside of the unit interval. A simple way to avoid this is to introduce a large positive number M such that $0 \leq h_{ik}/M \leq 1$ for all $i = 1, \ldots, m$, $k = 1, \ldots, n$, and let

$$h(w_i) = (h_{i1}/M)/d_1 + (h_{i2}/M)/d_2 + \cdots + (h_{in}/M)/d_n.$$

Then, using (4) and (5), we have

$$R(w_i, w_j) = \frac{\sum_k \min(h_{ik}, h_{jk})}{\sum_k \max(h_{ik}, h_{jk})}, \tag{6}$$

$$N(w_i, w_j) = \frac{\sum_k \min(h_{ik}, h_{jk})}{\sum_k h_{ik}}. \tag{7}$$

The number M disappears in calculating R and N as above. Therefore we need not determine an actual value of M.

Remark. In a foregoing paper [13] we called the relations R and N given by (6) and (7) a pseudothesaurus to emphasize the fact that the set C is replaced by the set D. Here, however, we call them a fuzzy thesaurus for simplicity, since the difference between a fuzzy thesaurus and a fuzzy pseudothesaurus is not important in this paper.

There are other ways for defining a fuzzy RT and a fuzzy NT. The relations defined above are typical, however. The above two measures have different

backgrounds. The relation R is closely related to the Jaccard coefficient in cluster analysis [1]. The relation N defined by (7) is identical with a measure proposed by Salton [20]. Salton proposed a measure which is identical with $N(v, w)$ using a heuristic argument without a mathematical model such as the one defined above. He used a threshold K and defined two relations:

$$v \text{ and } w \text{ are synonymous} \quad \text{iff} \quad N(v, w) \geq K \text{ and } N(w, v) \geq K,$$

$$w \text{ is a parent of } v \quad \text{iff} \quad N(v, w) \geq K \text{ and } N(w, v) < K.$$

We may interpret the two relations *synonymous* and *parent* as RT and BT, respectively. Apart from the difference in terminology, we note that the foregoing research heuristically introduced measures of associations, whereas we develop here a fuzzy set model for thesauri and define the measures based on this model. Another difference is the following. In foregoing research, thresholds are applied to measures for generating binary relations of the crisp type of a thesaurus. On the other hand, we use fuzzy relations themselves as a fuzzy thesaurus for fuzzy information retrieval. A form of fuzzy information retrieval through a fuzzy thesaurus is formulated in the next section. Meanwhile, we turn to other aspects of the above formulation. This is, an algorithm for generating fuzzy thesauri and a generalization of the above model.

Even by present computers, it is difficult to calculate values of the fuzzy relations (6) and (7) using arrays in a straightforward way, since the numbers of elements in W and in D are very large. Therefore the size of the matrices may amount to several thousands times several hundred thousands. Although techniques to handle sparse matrices may be applied, there is another method for generating R and N based on manipulation of sequential files. The principal tool for this is sorting.

In the following description of an algorithm for generating a fuzzy thesaurus which is called here GFT, the symbol (a, b, c) means a record in which fields are a, b, and c. $\{(a, b, c)\}$ means a set of records such as (a, b, c). The set $\{(a, b, c)\}$ is stored as a sequential file in the memory of a computer. Input to the algorithm GFT is a set D of documents. Each document $d \in D$ has a number of keywords in W. A keyword $w \in W$ may occur twice or more in a document. The frequency of occurrence of w_i in d_k is denoted by h_{ik}. Output from GFT is a set of records $\{(w_i, w_j, R(w_i, w_j))\}$ for all pairs (w_i, w_j) such that $R(w_i, w_j) \neq 0$. For simplicity, we do not describe generation of $N(w_i, w_j)$, since it is easy to modify GFT for generating $N(w_i, w_j)$. Note that GFT uses two work files WORK1 and WORK2 which are sequential. Note also that the algorithm uses *a* for–repeat loop [6], where 'for all' means that all elements in a file are examined sequentially.

Algorithm GFT (Generation of a fuzzy thesaurus).
// Find pairs of keywords in every document. //
for all $d_k \in D$ do
  find all keywords $w_i \in W$ and calculate h_{ik}
  for all (w_i, w_j), $w_i < w_j$, that are found in d_k do
    make record $(w_i, w_j, \min(h_{ik}, h_{jk}))$
    output $(w_i, w_j, \min(h_{ik}, h_{jk}))$ to WORK1

```
  repeat
  for all w_i that are found in d_k do
    make record (w_i, h_ik)
    output (w_i, h_ik) to WORK2
  repeat
repeat
// Sort WORK1 and WORK2. //
sort WORK1 into increasing order of the key (w_i, w_j)
sort WORK2 into increasing order of the key w_i
// Calculate R. Scan WORK1 and WORK2. //
for all (w_i, w_j) in WORK1 do
  find all records for (w_i, w_j) in WORK1
    and all records for w_i and w_j in WORK2
  R(w_i, w_j) ← Σ_k min(h_ik, h_jk)/(Σ_k h_jk + Σ_k h_jk − Σ_k min(h_ik, h_jk))
  output (w_i, w_j, R(w_i, w_j)) to an output file
repeat
end-of-algorithm GFT.
```

Note that in the first large for–repeat loop, we do not calculate a record $(w_i, w_j, \min(h_{ik}, h_{jk}), \max(h_{ik}, h_{jk}))$. If we calculate the latter form of records with $\max(h_{ik}, h_{jk})$, many records in WORK1 will have $\min(h_{ik}, h_{jk}) = 0$, and the number of records in WORK1 will be far greater than that in GFT.

In a foregoing paper [13] an experimental calculation on three thousand documents and thirty thousand keywords was carried out using a former version of the algorithm GFT based on sorting and the result shows a reasonable amount of 800 sec of CPU time.

Another possible application of the above model is a generalization of the concept of fuzzy thesaurus defined above to fuzzy associations of different types. We have fuzzy associations by replacing the set of keywords by other sets. We assumed before that W is a set of keywords. For the moment, however, we consider that W is a set of descriptors, which means that other kinds of indexes such as citation indexes [4] are taken as W. The above model is directly applied and we have two relations $R(w_i, w_j)$ and $N(w_i, w_j)$. We call the relations defined by (4) and (5) a fuzzy association on W based on the set C. When the set C is replaced by D, equations (6) and (7) define a fuzzy association on W based on the set of documents.

Suppose that W is a set of bibliographic citations. There is a large scale Science Citation Index database, and therefore, it is not exceptional to use a query in terms of citations for searching documents indexed by citations. Thus, a fuzzy association on citations expands the query and documents are found that have the given citation or the associated citations. It is obvious that the algorithm GFT is useful in generating various kinds of fuzzy associations.

Another significance of fuzzy associations is that current studies in analysis of bibliographic information are discussed in terms of a model of fuzzy associations. A typical example of bibliographic analysis is clustering of documents. For example, two methods of clustering using citations have been proposed: one

called bibliographic coupling, by Kessler [9], and the other co-citation, proposed by Small [22]. Bibliographic coupling is a method by which documents are clustered using frequencies of common citations; the co-citation method clusters cited documents using frequencies of source documents that refer to a pair of cited documents simultaneously.

To see what the present model of fuzzy association contributes to this subject, note that cluster analysis can be divided into three stages: (1) determination of a set of objects to be grouped and of a set of attributes on which a similarity measure is defined, (2) definition of a similarity measure, and (3) generation of clusters by choosing an appropriate algorithm.

The above two methods of citation clustering concern stage 1. Other studies (e.g., [4]) proposed heuristic algorithms at stage 3. In this way, in these studies either stage 1 or stage 3 have been considered but stage 2 has not been discussed in detail. The present model can deal with these stages of clustering of documents in a unified framework as follows.

(1′) Stage 1 concerns choice of the set W and the set C in the present model. Bibliographic coupling means that W is a set of documents and C is a set of citations. Co-citation means that W is a set of citations and C is the set of documents.

(2′) For stage 2, a symmetric measure of the fuzzy association such as $R(w_i, w_j)$ is useful. As already mentioned, $R(w_i, w_j)$ defined by (6) is a generalization of a well-known similarity measure for clustering which is called the Jaccard coefficient [1]. The algorithm GFT is useful for a large set of objects.

(3′) For stage 3, graph-theoretical algorithms are useful for generating clusters in case of a large set of documents. A typical graph-theoretical algorithm is the nearest neighbor method which is shown to be equivalent to calculation of the transitive closure $\hat{R}(w_i, w_j)$, [2, 27], provided that we use $R(w_i, w_j)$ as the similarity measure for clustering. The algorithm GFT followed by the minimal spanning tree (MST) algorithm generates $\hat{R}(w_i, w_j)$, i.e., clusters by the nearest neighbor method [1]. Note that both GFT and MST (by Kruskal's algorithm, cf. [6]) are based on sorting of sequential files of the same type of records.

Thus, the present model provides a unified framework for considering all the three stages of document clustering.

4. Fuzziness in retrieval: second component

We show in this section that output of the second component is expressed as $r' = UFq$. (See Figure 1.) We assume that F is a fuzzy relation that represents a fuzzy association defined above. The reader may consider that F shows fuzzy related terms: $F(v, w) = R(v, w)$. Note also that U is a fuzzy relation $U(d, w)$ on $D \times W$ or a fuzzy set valued function $U : W \to [0, 1]^D$. We use these notations interchangeably without confusion.

First let us consider that F and U are binary, i.e., thesaurus and indexing are crisp. In this case $U(w)$ means a crisp subset of documents that have the keyword w as an index. Note that U is implemented as an inverted index of a retrieval

system. For the crisp case a retrieval through a thesaurus given a keyword w is as follows. (a) Examine the thesaurus F and find all associated terms $v_{l_1}, v_{l_2}, \ldots, v_{l_p}$. (b) Find subsets $U(v_{l_1}), U(v_{l_2}), \ldots, U(v_{l_p})$. (c) Establish the retrieved set of documents as the union of $U(v_{l_1}), \ldots, U(v_{l_p})$: $\bigcup_{1 \leq i \leq p} U(v_{l_i})$.

Now, let $Uf(d, w)$ be a fuzzy relation that shows the degree of membership of the document d in the retrieved set r' by giving the keyword w to the system in Figure 1. When U and F are crisp, we have

$$Uf(d, w) = \begin{cases} 1 & \text{iff } d \in U(v_l) \text{ for some } v_l \text{ such that } F(v_l, w) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

When the thesaurus F is fuzzy and U is crisp, noting that the union is defined by max, we have [14]

$$Uf(d, w) = \max_{\substack{d \in U(v) \\ \text{for all } v \in W}} F(v, w). \tag{8}$$

Where the function U is represented as a binary relation, we have

$$Uf(d, w) = \max_{v \in W} \min[U(d, v), F(v, w)]. \tag{9}$$

The last equation is valid also for a fuzzy relation $U(d, v)$. So we obtain the relation between a keyword and a document described by (9), when F and U are both fuzzy.

Thus, if a query q is a simple keyword w, then the response r' is the fuzzy set $r' = UF(\cdot, w)$. For a fuzzy query $q = \sum q_i/w_i$, the response is

$$r' = \sum_i \max \min[Uf(d_j, w_i), q_i]/d_j = (UF)q,$$

using (9). Since the associative law holds for fuzzy algebra, we have $r' = UFq$.

Remark. The equation (9) is represented in terms of Sugeno's integral [24] as

$$Uf(d, w) = \int_W U(d, \cdot) \circ \bar{F}(\cdot, w) = \int_W F(\cdot, w) \circ \bar{U}(d, \cdot)$$

where the fuzzy measure $\bar{F}(\cdot, w)$ and $\bar{U}(d, \cdot)$ are defined by

$$\bar{F}(K, w) = \max_{v \in K} F(v, w) \quad \text{and} \quad \bar{U}(d, K) = \max_{v \in K} U(d, v),$$

respectively.

As we considered in the previous section, an algorithm for calculating Uf or r' is necessary, since manipulation of U and F as arrays is cumbersome. Here we show two types of algorithms. The first algorithm which is called here FR1 is based on sorting on sequential files.

For simplicity, we assume that input to FR1 is a keyword $w \in W$. The fuzzy thesaurus is assumed to be stored as a file FT (Fuzzy Thesaurus). This algorithm uses two more files: WORK (WORK file) and OUT (OUTput file). The

inverted index $U(v)$ for a given v consists of records $\{(d, U(d, v))\}$: a record $(d, U(d, v))$ consists of the document identifier d and the value of membership $U(d, v)(= T(v, d))$ for a fuzzily indexed keyword v. Output from FR1 is a set of records $\{(d, Uf(d, w))\}$ for all $d \in D$ such that $Uf(d, w) \neq 0$. Note that in the following algorithms conditional statements are described by if–then–endif [6].

Algorithm FR1 (Fuzzy retrieval).
// First step: Find all records. //
for all v such that $F(v, w) \neq 0$ in FT do
  for all $d \in U(v)$ do
    $p(d, v) \leftarrow \min[U(d, v)), F(v, w)]$
    output record $(d, p(d, v))$ to a work file WORK
  repeat
repeat
// Second step: Find values of Uf. //
sort WORK into increasing order of the first key d
  and into decreasing order of the second key p
// The above sorting means that in the resulting sequence, a record (d_i, p_i) //
// before another record (d_j, p_j) satisfies either $d_i < d_j$ or $d_i = d_j$, $p_i > p_j$. //
take the first record (d_1, p_1) in WORK
$(D, P) \leftarrow (d_1, p_1)$
for all d_j in WORK do
  // the d_j's are sequentially examined. //
  if $D \neq d_j$ then
    output (D, P) to an output file OUT
    $(D, P) \leftarrow (d_j, p_j)$
  endif
repeat
output (D, P) to OUT
// OUT contains exactly those records that represent $p = Uf(d, w)$ defined by (9). //
// Third step: If necessary, sort again. //
sort OUT into the decreasing order of the key p
and print OUT
end–of–FR1.

At the end of the second step in FR1, all the necessary records are obtained as a retrieved fuzzy set. The third step arranges the retrieved set for printing from the most relevant documents to less relevant ones. When a retrieved set is not printed, e.g., fuzzy set operations are performed on two or more retrieved sets, then the third step is unnecessary.

Another algorithm which is called here FR2 needs stronger assumptions but requires less processing time. The algorithm FR2 does not use sorting. Input and output are the same as those in FR1. Here, however, we need three assumptions. First, $U(v)$ is crisp. That is, the thesaurus is fuzzy but directly indexed keywords do not have any membership specification. Second, the fuzzy thesaurus has the

following form: for each $w \in W$, there is a sequential file

$$F(w) = \{(v_{l_1}, f_{l_1}), (v_{l_2}, f_{l_2}), \ldots, (v_{l_s}, f_{l_s})\}, \quad f_{l_k} = F(v_{l_k}, w), \ k = 1, \ldots, s,$$

which satisfies $f_{l_1} \geq f_{l_2} \geq \cdots \geq f_{l_s}$. That is, the sequential file $F(w)$ is arranged according to decreasing order of $F(v, w)$. Third, we use a binary valued function $B: D \rightarrow \{0, 1\}$. The function $B(d)$ means that $B(d) = 1$ if d is already retrieved, otherwise $B(d) = 0$.

Algorithm FR2 (Fuzzy retrieval).
// Initialize B. //
for all $d \in D$ do
  $B(d) \leftarrow 0$
repeat
// Keyword w is given. //
for all $v_{l_k} \in F(w)$, $k = 1, \ldots, s$, do
  for all $d \in U(v_{l_k})$ do
    if $B(d) = 0$ then
      $B(d) \leftarrow 1$
      output record (d, f_{l_k}) to OUT
    endif
  repeat
repeat
end-of-FR2.

Note that from the second assumption it is clear that the resulting OUT is arranged according to decreasing order of the key f of the records $\{(d, f)\}$. This algorithm is not useful when $U(v_{l_k})$ is fuzzy.

As described above, the method of automatic generation of thesauri naturally leads to a fuzzy thesaurus. On the other hand, large scale bibliographic databases do not have fuzzy indexes: Keyword indexes and other descriptors are specified in a crisp way. Therefore a study of the second component in Figure 1, the fuzzy inverted index, should include how crisp indexes are modified to fuzzy ones. For this purpose, the following considerations are useful.

(a) (Weighting on crisp descriptors) To give a weight as the membership on each descriptor, frequency of occurrence of a descriptor in a document may be transformed into a weight in the unit interval.

(b) (Automatic indexing) In many cases original data for bibliographic databases do not have adequate indexes. Therefore, a large amount of human effort is necessary for specifying descriptors. Moreover, methods of automatic indexing have been studied which use some technique of pattern matching. As is usual in pattern matching techniques, some descriptors are judged to be quite adequate, some others are more or less relevant, and so on. It is also usual that some degree of relevance is obtained for each candidate descriptor. In such a case, fuzzy indexes are useful, since the degree of relevance is immediately interpreted as the membership of a fuzzy descriptor.

Remark. A simple way to realize fuzzy retrieval is to implement the system as an extended feature of a crisp retrieval system. In this way, fuzzy thesauri and fuzzy associations are easier to implement than the fuzzy inverted index, since the latter needs modification on a greater scale of the underlying system of crisp information retrieval.

5. Fuzziness on output: third component

To introduce a fuzzy filter, let us begin with the examples in Section 2: (a) Find *recent* documents that have keyword *w*. (b) Find documents that have keyword *w* and are *relevant to one's field of interest.* We consider example (a) first. The second example is dealt with in the same way. The italicized part *recent* can be represented as a seperate query: "Find recent documents". Assume that another set $Z = \{z_1, z_2, \ldots, z_s\}$ of an index and another fuzzy index $V: Z \rightarrow [0, 1]^D$ are given. We suppose that Z is a set of years of publication of documents. Then the fuzzy query *recent* is naturally represented as a fuzzy set $y = (y_1, y_2, \ldots, y_s)^T$ in vector form. The response of this query is given by

$$g = Vy.$$

On the other hand, assume that query q in the set W represents keyword w and that $r' = UFq$. Now, the query of example (a) is interpreted as: "Find documents that have keyword w and that are recent". Accordingly, the response r of the last query is represented as the intersection of r' and g. Namely, the response of the query of example (a) is

$$r = r' \cap g. \tag{10}$$

In other words, $r_i = \min[r'_i, g_i]$, $i = 1, 2, \ldots, n$.

Consider example (b). We use the same symbols but we assume now that Z is a set of scientific journals where the documents are published papers. In many cases the relevance of documents to a particular field of research is expressed by a preference for journals. Suppose that a user's preference is expressed also by a fuzzy set y of the same symbol. We obtain the response r given by (10) of the same expression as in case of example (a), although the meanings are different.

Actually, a user does not input the second query y. Instead, the query y should be implemented as a kind of user profile that shows the user's general preference. The profile is applied to response $r' = UFq$ by equation (10). Therefore, we represent the response $g = Vy$ as the third component P. The relation P in the matrix form is given by:

$$P = \mathrm{Diag}(g) = \begin{bmatrix} g_1 & & 0 \\ & \ddots & \\ 0 & & g_n \end{bmatrix}.$$

We call the above relation P a linear diagonal filter or simply a linear filter. Using fuzzy algebra, we have

$$r = \mathrm{Diag}(g)UFq = PUFq \tag{11}$$

It is easy to see that the above equation (11) is equivalent to (10). Thus, the three components of the relation (3) are all justified. We note also that the response $g = Vy$ is calculated by

$$g_i = \max_j \min[V(d_i, z_j), y_j], \quad i = 1, 2, \ldots, n,$$

according to fuzzy algebra that is applied to all the components of fuzzy retrieval.

Actual processing of a linear filter is based on next equation. For each retrieved document d in the response r', we have a record (d, p'), where p' is the membership value of d. Then the final membership p is calculated by

$$p = \max_j \min[y_j, V(d, z_j), p']. \tag{12}$$

In general, the number of elements in Z is assumed to be far smaller than that in D. Therefore, direct calculation of p using g and (10) needs a larger amount of calculation than (12).

It should be noted that the first component of a fuzzy thesaurus expands an input query, whereas the linear filter reduces membership. Thus, a query is expanded, a database is searched, and then the retrieved set is reduced by the linear filter.

Here we have considered only a linear diagonal filter. Other types of linear and nonlinear filters for information retrieval are an interesting subject for further research.

6. Classification of output

The output of a fuzzy retrieval should be sorted according to decreasing order of membership, since a user wishes to examine more relevant documents prior to less relevant ones. On the other hand, most retrieval systems for bibliographic databases do not print out retrieved documents immediately after they are retrieved. The reason for this is that frequently a retrieved subset includes a large number of documents so that it is expensive and cumbersome to print out all the documents in a retrieved set. Therefore, a retrieved set is first established and then another request for printing is needed, frequently with options to select some portion or fields of the retrieved documents.

In case of fuzzy information retrieval, when a retrieved subset is established, the subset should be divided into several layers according to the values of membership so that a user can select some layers from a retrieved set. Thus, when $K-1$ thresholds $\alpha_1, \alpha_2, \ldots, \alpha_{k-1}$ such that $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_{k-1} < 1$ are given, a retrieved fuzzy subset $\mathrm{FS} \subset D$ may be divided into K layers $\mathrm{FS}_1, \mathrm{FS}_2, \ldots, \mathrm{FS}_k : d \in \mathrm{FS}_i$ iff the membership of d is in $(\alpha_{i-1}, \alpha_i]$. (Assume that $\alpha_0 = 0$, $\alpha_k = 1$.) In other words, if we denote the alpha-cut of FS by $C(\alpha)\mathrm{FS}$, then $\mathrm{FS}_i = C(\alpha_{i-1})\mathrm{FS} - C(\alpha_i)\,\mathrm{FS}$.

In general it is difficult to fix parameters $\alpha_1, \ldots, \alpha_{k-1}$ beforehand, since a retrieved set may have a large number of low membership values and the number

of documents in FS_1 may be large and $FS_2, \ldots, FS_k$ may have few documents. Therefore for efficient use of the layers, parameters $\alpha_1, \ldots, \alpha_{k-1}$ should be determined dynamically after a retrieved set is obtained. A simple policy is to determine $\alpha_1, \ldots, \alpha_{k-1}$ so that the numbers of documents in all the layers are the same. That is, if we denote the number of documents in FS_i by $|FS_i|$, then this policy requests $|FS_1| = |FS_2| = \cdots = |FS_k|$. In a foregoing paper [15], we showed this policy optimizes two different criteria. The above policy is based on the assumption that an equal amount of attention is paid to all the layers. Actually, however, layers of higher relevance FS_k, $FS_{k-1}, \ldots$ will have more attention than layers of lower relevance. Therefore some other criteria should be considered. (See [15].)

7. Conclusion

A fuzzy set model provides a clear view on current crisp methods in information retrieval and their implications; it suggests what should be studied furthermore. In Section 2 we divided the process of information retrieval into three components. The last component, of a fuzzy filter, has not been studied in the crisp framework. The fuzzy set model enables the study of the third component. There have been studies of weighted retrieval (see, e.g., Heaps [5]), which suggest the use of weighting on outputs. Readers will find how the fuzzy set model provides a clearer view than the current model of weighted retrieval without fuzzy sets.

There are many problems to be solved theoretically and practically as further studies of fuzzy information retrieval. Some problems are as follows. (1) Discussion of crisp techniques of advanced indexing and retrieval using a fuzzy set model, cf. [4, 5, 21, 26]. (2) Studies of efficient algorithms for large scale databases. In particular, development of hardware for information retrieval should be taken into account. (3) Application of methods in fuzzy information retrieval to related areas, for example, structure of texts and bibliography and its application to education.

References

[1] M.R. Anderberg, *Cluster Analysis for Applications* (Academic Press, New York, 1973).
[2] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications* (Academic Press, New York, 1982).
[3] J.C. Dunn, A graph theoretic approach of pattern classification via Tamura's fuzzy relation, *IEEE Trans. Systems Man Cybernet.* **4** (1974) 310–313.
[4] E. Garfield, *Citation Indexing-Theory and Application in Science, Technology, and Humanities* (Wiley, New York, 1979).
[5] H.S. Heaps, *Information Retrieval: Computational and Theoretical Aspects* (Academic Press, New York, 1978).
[6] E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms* (Computer Science Press, Rockville, MD, 1978).

[7] J. Kacprzyk and A. Ziolkowski, Database queries with fuzzy linguistic quantifiers, *IEEE Trans. Systems Man Cybernet.* **16**(3) (1986) 474–479.

[8] A. Kandel, *Fuzzy Mathematical Techniques with Applications* (Addison-Wesley, Reading, MA, 1986).

[9] M.M. Kessler, Bibliographic coupling between scientific papers, *Amer. Documentation* **14**(1) (1963) 10–25.

[10] L.J. Kohout, E. Keravnou and W. Bandler, Information retrieval system using fuzzy relational products for thesaurus construction, *Proc. of the IFAC Symposium on Fuzzy Information, Knowledge Representation, and Decision Analysis*, Marseille, France (Pergamon Press, Oxford, 1983).

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

[12] F. W. Lancaster, *Vocabulary Control for Information retrieval* (Information Resources, Washington, DC, 1972).

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

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

[15] S. Miyamoto, Fuzzy relations as general knowledge for information retrieval and classes of relevance, *Preprint of Second IFSA Congress*, Tokyo (July 1987) 719–722.

[16] C.V. Negoita and P. Flondor, On fuzziness in information retrieval, *Internat. J. Man–Machine Stud.* **8** (1976) 711–716.

[17] H. Prade and C. Testemale, Application of possibility and necessity measures to document information retrieval, in: B. Bouchon and R.R. Yager, Eds., *Uncertainty in Knowledge-Based Systems*, Lecture Notes in Computer Science No. 286 (Springer-Verlag, Berlin, 1987) 265–274.

[18] T. Radecki, Mathematical model of information retrieval system based on the concept of fuzzy thesaurus, *Inform. Process. and Management* **12** (1976) 313–318.

[19] T. Radecki, Fuzzy set theoretical approach to document retrieval, *Inform. Process. and Management* **15** (1979) 247–259.

[20] G. Salton, Ed., *The SMART Retrieval System, Experiments in Automatic Document Processing* (Prentice-Hall, Englewood Cliffs, NJ. 1971).

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

[22] H. Small, Co-citation in the scientific literature: a new measure of the relationship between two documents, *J. Amer. Soc. Inform. Sci.* **24**(4) (1973) 265–269.

[23] K. Spark Jones, *Automatic Keyword Classification for Information Retrieval* (Butterworth, London, 1971).

[24] M. Sugeno, Theory of fuzzy integrals and its applications, Thesis, Tokyo Institute of Technology (1974).

[25] V. Tahani, A conceptual framework for fuzzy query processing: a step toward very intelligent database systems, *Inform. Process. and Management* **13** (1977) 289–303.

[26] C.J. van Rijsbergen, *Information Retrieval*, Second Edition (Butterworth, London, 1979).

[27] L.A. Zadeh, Similarity relations and fuzzy orderings, *Inform. Sci.* **3** (1971) 177–200.

[28] L.A. Zadeh, The role of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets and Systems* **11** (1983) 199–227.