ISE-TR-88-68

INFORMATION RETRIEVAL BASED ON FUZZY ASSOCIATIONS

by

Sadaaki MIYAMOTO

June 3, 1988

INSTITUTE
OF
INFORMATION SCIENCES AND ELECTRONICS

UNIVERSITY OF TSUKUBA

Information Retrieval Based on Fuzzy Associations

S. Miyamoto

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

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 divided into three stages. Each stage has its inherent fuzziness. As typical examples for describing the three stages, we consider a fuzzy association as a generalization of a fuzzy thesaurus on the first stage. a fuzzy inverted index on the second stage, and a fuzzy filter on the third stage. Efficient algorithms for fuzzy retrieval on large scale bibliographic databases are developed. A 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

Researches in fuzzy information retrieval have been concentrated on theoretical aspects such as processing of fuzzy queries, e.g., [7,11,16,24], mathematical properties of a fuzzy thesaurus [17,18], 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. Hardwares for information retrieval were not well-developed for realizing fuzzy information retrievals 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.

We take in this paper a conservative standpoint which means the following. The subject of fuzzy information retrieval can be approached from two sides. The researches mentioned above face one side that shows various types of fuzziness possibly included in fuzzy information retrieval. The other side which will be the subject of this paper is linked to current studies in crisp information retrieval. Some topics of researches on this side are as follows. How a current system of an ordinary (crisp, nonfuzzy) information retrieval system can be extended for including fuzzy retrieval? What is the significance of fuzzy sets as a mathematical model for studying current problems of interest in

crisp information retrieval? What can we contribute practically by introducing the idea of fuzzy information retrieval? The latter side of researches is studied by a conservative standpoint which means that introduction of fuzziness is limited to a scope that links a fuzzy mathematical model to current studies in nonfuzzy information retrieval. Thus, we do not consider a fuzzy user queries in detail, since a user of information retrieval may not have the idea of fuzzy queries. Maybe he will issue crisp queries. He may not expect an output with membership values. We will show that fuzzy information retrieval is still useful even in such a case. That is, a crisp query is extended to a fuzzy query using a fuzzy association, and the final output is summarized into several classes of relevance. A user will see the output classes without referring directly to the memberships.

While other researches in fuzzy information retrieval pursuit fuzziness that are not considered in current information retrieval system of crisp type, the present study extracts fuzziness hidden behind current studies on crisp information retrieval and analysis of bibliographic information. By the latter approach, implication of current researches in the field of crisp retrieval are made clear and different topics of the current researches are interrelated using a fuzzy set model. Of course new results are derived and algorithms for large scale databases are developed. Thus, we show in this paper how current studies are put into a unified framework of fuzzy sets. Then possibilities of researches that are not dealt with in the current studies are made clear. The possibilities include fuzzy association

retrieval that is a generalization of a fuzzy thesaurus. Fuzzy retrieval needs a new user-interface, which in turn requires study of a new type of user profiles and options on output of results of retrieval. Algorithms that are useful for a large scale database are necessary for realizing fuzzy retrieval on a practical retrieval system. How a fuzzy information retrieval is implemented in practice should be also studied. This paper is concerned with all of these features. Thus, the main objective of this paper is to show how fuzzy sets provide an appropriate model for information retrieval and to link interest of researchers in crisp information retrieval to studies in fuzzy information retrieval.

2. Three stages 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 $q=(q_1, q_2, \ldots, q_m)$. We do not distinguish between the fuzzy set q and the vector representation q by the same reason.

A simple information retrieval system may be visualized as a block diagram as Fig. 1 whose input is a query and output is a response as a set of documents. Specifically, the query is in general a fuzzy set q of W, the response is a fuzzy set r of D, and the relation between q and r is described by the fuzzy relation U defined on DxW. Let $m_A(\cdot)$ be the membership of a fuzzy set in general. Then, relation between the query and the response is given by (See section 4 for the detail.)

$$r = \sum \max_j \min [U(d_i, w_j), m_q(w_j)] / d_i \qquad (1)$$

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 $q=(q_1, q_2, \ldots, q_m)$ and $r=(r_1, r_2, \ldots, r_n)$, and the fuzzy relation as a matrix $U=(u_{ij})$. Then the equation (1) is written as

$$r = U q \qquad (2)$$

using fuzzy algebra, that is, an algebra where the addition is maximum ($a + b = \max(a,b)$) and the 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). Note also that the equation (2) agrees well with the block diagram representation.

Let us consider a more complex diagram shown as Fig. 2. This diagram has three components. First, a given query may not be adequate for the indexes of the database, therefore the input query is expanded to include synonyms and related keywords that are more appropriate. Thus, a typical example of the function F on the first stage is a fuzzy thesaurus. The second component is already described above. The last component is called here a fuzzy filter in information retrieval. A fuzzy filter may decrease membership values of some part of retrieved documents r' after the second stage, or it may increase membership of another part of the retrieved documents. As we will see later, fuzzy thesauri and fuzzy associations as a generalization of the former are represented as a "linear" operation in the sense of fuzzy algebra. On the other hand, a fuzzy filter may be nonlinear in general, when it amplifies some part of r'. Thus, we represent r = P(U F q) . Note that since P is nonlinear, we do not write as r = PUFq, although in section 5 we deal solely with a linear filter. From now we assume that the function F is a fuzzy thesaurus or a fuzzy association, and that U is a fuzzy inverted index as explained above.

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 are introduced and studied as each component of the three stages in Fig. 2.

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 detail.) 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., [19,22]). A well-known technique for this is based on counting frequencies of simulta-

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

$$R(v,w) = \frac{| h(v) \cap h(w) |}{| h(v) \cup h(w) |} \tag{3}$$

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

where $|A|$ for a fuzzy set A means the cardinality of A. (See [2]. Sometimes $|A|$ is written as $\sum \mathrm{Count}(A)$. See [8,27].)

The meaning of R and N is clearly explained by Fig. 3. Namely, $R(v,w)$ is 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 is 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 a dashed curve in Fig.

3. if this area means h(v), then N(v,w)=1 but R(v,w)<<1. From the above property it is natural to call the above fuzzy relations as a fuzzy thesaurus. The relation R defined by (3) and N defined by (4) 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 $h: W \rightarrow [0,1]^D$ is naturally defined in terms of frequencies of occurrences of $w \in W$ in the document $d \in D$. Let h_{ik} be the frequency of occurrences of w_i in the document d_k. If we take

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

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

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

Then, using (3) and (4), we have

$$R(w_i, w_j) = \frac{\sum_k \min (h_{ik}, h_{jk})}{\sum_k \max (h_{ik}, h_{jk})} \qquad (5)$$

$$N(w_i, w_j) = \frac{\sum_k \min (h_{ik}, h_{jk})}{\sum_k h_{ik}} \qquad (6)$$

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 (5) and (6) as a pseudothesaurus to emphasize the fact that the set C is replaced by the set D. Here, however, we call them as a fuzzy thesaurus for simplicity, since 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 (6) is identical with a measure proposed by Salton [19]. 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 and w are synonymous iff $N(v,w) \geq K$ and $N(w,v) \geq K$

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

We may interprete the two relations synonymous and parent as RT and BT, respectively. Apart from difference of the terminologies, 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 the model. Another difference is the following. In foregoing

researches 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 fuzzy information retrieval through a fuzzy thesaurus is formulated in the next section. Meanwhile, we turn to other aspects of the above formulation. That is, an algorithm for generating fuzzy thesauri and a generalization of the above model.

Even by the present computer, it is difficult to calculate values of the fuzzy relations (5) and (6) using arrays in a straightforward way, since numbers of elements in W and in D are very large. Therefore the size of the matrices may be 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 a storage 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. Frequency of occurrences 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 description of a loop by for-repeat [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 WORK 2. //

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

$R(w_i, w_j) \leftarrow \sum_k \min(h_{ik}, h_{jk}) / (\sum_k h_{ik} + \sum_k h_{jk} - \sum_k \min(h_{ik}, h_{jk}))$

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 loop of for-repeat, 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 an 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 the CPU time.

Another possible application of the above model is a generalization of the concept of the 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 (3) and (4) as a fuzzy association on W based on the set C. When the set C is replaced by D, equations (5) and (6) 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 database Science Citation Index, 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 the 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 is called bibliographic coupling by Kessler [9] and the other is co-citation proposed by Small [21]. Bibliographic coupling is a method by which documents are clustered using frequencies of common citations; 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, 3. generation of clusters by choosing an appropriate algorithm.
The above two methods of citation clustering concern the stage 1. Other studies (e.g., [4]) proposed heuristic algorithms on the stage 3. In this way, in these studies either the stage 1 or the stage 3 have been considered but the 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'. The 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 the stage 2, a symmetric measure of the fuzzy associaiton such as $R(w_i, w_j)$ is useful. As is already mentioned, $R(w_i, w_j)$ defined by (5) 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 the 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 $\hat{R}(w_i, w_j)$ [2,26], 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

To begin with, we show that output of the second component is expressed as $r'=UFq$. (See Fig. 2.) We assume that F is a fuzzy relation that represents a fuzzy association defined above. A 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 DxW or a fuzzy set valued function $U:W\rightarrow[0,1]^D$. 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. 1. Examine the thesaurus F and find all associated terms $v_{\ell_1}, v_{\ell_2}, \ldots, v_{\ell_p}$. 2. Find subsets $U(v_{\ell_1})$, $U(v_{\ell_2}), \ldots, U(v_{\ell_p})$. 3. Establish the retrieved set of documents as the union of $U(v_{\ell_1}), \ldots, U(v_{\ell_p})$: $\bigcup_{1 \le i \le p} U(v_{\ell_i})$.

Now, let Uf(d,w) be a fuzzy relation that shows degree of membership of the document d in the retrieved set r' by giving the keyword w to the system in Fig. 2. When U and F are crisp, we have

Uf(d,w) = 1 iff $d \in U(v_\ell)$ for some v_ℓ such that $F(v_\ell, w)=1$

Uf(d,w) = 0 otherwise.

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

$$Uf(d,w) = \max_{\substack{d \in U(v) \\ \text{for all } v \in W}} F(v,w) \qquad (7)$$

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

$$Uf(d,w) = \max_{v \in W} \min [\, U(d,v),\ F(v,w) \,] \qquad (8)$$

The last equation is valid also for a fuzzy relation U(d,v). Thus we obtain relation between a keyword and a document described by (8), when F and U are both fuzzy.

Remark The last equation is represented in terms of the Sugeno's integral [23]

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

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

$$\bar{F}(K,w) = \max_{v\in K} F(v,w)$$

and

$$\bar{U}(d,K) = \max_{v\in K} U(d,v)$$

respectively. []

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

$$r' = \max_i \min [Uf(d,w_i),q_i]$$

which is equivalent to r'=UFq.

As we consider in the previous section, algorithm for calculating Uf or r' is necessary, since manipulation of U and F as arrays is cumbersome. Here we show two types of the algorithms. 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). The inverted index U(v) for a given v consists of records {(d,U(d,v))}: a record (d,U(d,v)) cocnsists 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)≠0 in FT   do
   for all d∈U(v) do
      p(d,v) ← 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 the increasing order of the first key d and into
the 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) ← (d_1 ,p_1 )
for all d_j in WORK   do   // d_j's are sequentially examined. //
  if D ≠ d_j then
     output (D,P) to an output file OUT
     (D,P) ← (d_j ,p_j )
   endif
repeat
output (D,P) to OUT
// OUT contains exactly those records that represent p=Uf(d,w)//
// defined by (8). //
// 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 as a retrieved fuzzy set are obtained. 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 a 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 $F(w)=\{(v_{\ell_1}, f_{\ell_1}), (v_{\ell_2}, f_{\ell_2}), \ldots, (v_{\ell_s}, f_{\ell_s})\}$. $f_{\ell_k}=F(v_{\ell_k}, w)$, $k=1,\ldots,s$, which satisfies $f_{\ell_1} \geq f_{\ell_2} \geq \ldots \geq f_{\ell_s}$. That is, the sequential file F(w) is arranged according to the 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 iff 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} ∈ F(w), k=1,..,s do
   for all d∈U(v_{ℓ_k}) do
      if B(d)=0 then
         B(d) ← 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 the decreasing order of the key f of the records {(d,f)}. This algorithm is not useful when $U(v_\ell)$ is fuzzy.

As is described above, a 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 the crisp way. Therefore a study of the second component in Fig. 2, the fuzzy inverted index, should include how crisp indexes are modified to fuzzy ones. For this purpose, the following considerations are useful. 1. (weighting on crisp descriptors) To give a weight as the membership on each descriptor, frequency of occurrences of a descriptor in a document may be transformed into a weight in a unit interval. 2. (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 auto-

matic 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 for the descriptors. In such a case, fuzzy indexes are useful, since the degree of relevance is immediately interpreted as a 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. By this way, fuzzy thesauri and fuzzy associations are easier to implement than the implementation of the fuzzy inverted index, since the latter needs modification in a greater scale of the underlying system of crisp information retrieval. []

5. Fuzziness on output: third component

Although the third component in Fig. 2 is nonlinear in general, here we consider solely a simple type of a "linear" filter. A linear filter in fuzzy retrieval is defined as follows. Let $g=(g_1,g_2,\ldots,g_n)\in[0,1]^D$ be n-vector which is identified with a fuzzy set $\sum g_i/d_i$ of D. For any output $r'=(r_1',r_2',\ldots,r_n')$ from the second component, a filtered output $r=(r_1,r_2,\ldots,r_n)$ is represented as

$$r_i = \min [\, g_i, r_i' \,], \qquad i=1,2,\ldots,n$$

or, in terms of fuzzy sets. $r = g \cap r'$.

The linear filter defined above implies that a user has a prior preference that acts as a threshold on retrieved sets represented by a fuzzy set g of D. If the membership of a

document d_i exceeds the threshold g_i, then the membership is reduced to g_i. Let us define a matrix

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

Suppose that the relations F and U are matrices, and that q, r', and r are vectors. Noting that the addition is minimum and the multiplication is maximum, we have a linear expression

$$r = \mathrm{Diag}(g)\ U\ F\ q \qquad (9)$$

It appears to be difficult to assume that one has a threhold of a prior preference on every document. However, it is usual that we have preference on some index set of descriptors. For example, every bibliographic database has an index of scientific journals where the documents were published. A specialist has a strong preference on scientific journals. Thus, it is not difficult at all to give preference on a set of scientific journals. Moreover, statistics on journal rankings have been published which enables another way of weighting on journals. Let $Z=\{z_1, z_2, \ldots, z_p\}$ be a set of descriptors (possibly, scientific journals) and $y=(y_1, y_2, \ldots, y_p) \in [0,1]^Z$ represents the preference given on the set Z. Let $V: D \rightarrow [0,1]^Z$ be a function which maps each document to its descriptors (V(d) shows descriptors of d). V(d) is represented as a fuzzy relation V(z,d) using the same symbol. Then, the preference y is transformed to the linear filter g:

$$g_i = \max_j \min [\, y_j \,,\, V(z_j, d_i) \,] \qquad (10)$$

In other words, $g = V^{-1}(y)$.

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

$$p = \max_j \min [y_j , V(z_j,d), p']. \qquad (11)$$

In general number of elements in Z is assumed to be far smaller than that in D. Therefore direct calculation of p using g_i and (9) needs a larger amount of calculation than (11).

It should be noted that the first stage 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.

6. Classification of output

The output of a fuzzy retrieval should be sorted according to the decreasing order of the 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 issued, frequently with options to select some portion or fields of the retrieved documents.

In case of a 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 out of a retrieved set. Thus, when K-1 thresholds $\alpha_1, \alpha_2, \ldots, \alpha_{K-1}$ such that $0<\alpha_1<\alpha_2<\ldots<\alpha_{K-1}<1$ are given, a retrieved fuzzy subset $FS \subset D$ may be divided into K layers $FS_1, FS_2, \ldots, FS_K$: $d \in FS_i$ iff membership of d is in $(\alpha_{i-1}, \alpha_i]$. (Assume that $\alpha_0=0$, $\alpha_K=1$.) In other words, if we denote alpha-cut of FS by $C(\alpha)FS$, then $FS = C(\alpha_{i-1})FS - C(\alpha_i)FS$.

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 $|FS_1|=|FS_2|=\ldots=|FS_K|$. In an foregoing paper [15], we showed this policy optimizes two different criteria. The above policy is based on an 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].)

6. 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 di-

vided a process of information retrieval into three stages. The last stage 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 a current model of weighted retrieval without fuzzy sets.

There are many problems to be solved theoretically and practically as future 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. [4,5,20,25]. 2. Studies of efficient algorithms for large scale databases. In particular, development of hardwares 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., Syst., Man, and Cybern., Vol.4, 310-313, 1974.

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, Maryland, 1978).

7. J. Kacprzyk and A. Ziolkowski, Database queries with fuzzy linguistic quantifiers, IEEE Trans., Syst., Man, and Cybern., vol. SMC-16, No.3, 474-479.

8. A. Kandel, Fuzzy Mathematical Techniques with Applications, (Addison-Wesley, Reading, Massachusetts, 1986).

9. M. M. Kessler, Bibliographic coupling between scientific papers, American Documentation, 14, 1, 10-25, 1963.

10. L. J. Kohout, E. Keravnou, 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, Int. J. Man-Machine Studies, 19, 45-56, 1983.

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 cooccurrences and fuzzy set operations, IEEE Trans., Syst., Man, and Cybern., vol.SMC-13, No.1, 62-70, 1983.

14. S. Miyamoto and K. Nakayama, Fuzzy information retrieval based on a fuzzy pseudothesaurus, IEEE Trans., Syst., Man, and Cybern.. vol.16, No.2, 278-282, 1986.

15. S. Miyamoto, Fuzzy relations as general knowledge for information retrieval and classes of relevance, Preprint of Second IFSA Congress, Tokyo, July, 719-722 (1983).

16. C. V. Negoita and P. Flonder, On fuzziness in information retrieval, Int. J. Man-Machine Studies, 8, 711-716, 1976.

17. T. Radecki, Mathematical model of information retrieval system based on the concept of fuzzy thesaurus, Information Processing and Management, 12, 313-318, 1976.

18. T. Radecki, Fuzzy set theoretical approach to document retrieval, Information Processing and Management, 15, 247-259, 1979.

19. G. Salton, ed., The SMART Retrieval System, Experiments in Automatic Document Processing, (Prentice-Hall, Englewood Cliffs, NJ, 1971).

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

21. H. Small, Co-citation in the scientific literature: a new measure of the relationship between two documents, Journal of the American Society for Information Science, 24, 4, 265-269, 1973.

22. K. Spark Jones, Automatic Keyword Classification for Information Retrieval, (Butterworth, London, 1971).

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

24. V. Tahani, A conceptual framework for fuzzy query processing: a step toward very intelligent database systems, Information Processing and Management, Vol13, 289-303, 1977.

25. C. J. van Rijsbergen, Information Retrieval, Second Edition, (Butterworth, London, 1979).

26. L. A. Zadeh, Similarity relations and fuzzy orderings, Information Sciences, 3, 177-200. 1971.

27. L. A. Zadeh, The role of fuzzy logic in the management of uncertainty in expert systems, Fuzzy Sets and Systems, 11, 199-227, 1983.

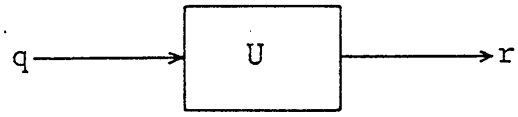

Fig. 1 A block diagram of a simple retrieval process

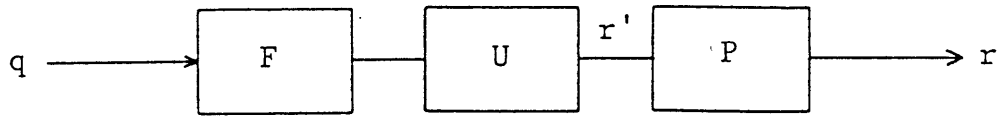

Fig.2 Representation of an information retrieval process with three components.

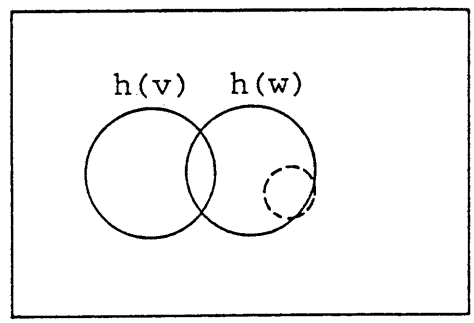

Fig. 3 Concepts of the words v and w in the set C.

INSTITUTE OF INFORMATION SCIENCES AND ELECTRONICS
UNIVERSITY OF TSUKUBA
SAKURA-MURA, NIIHARI-GUN, IBARAKI 305 JAPAN

REPORT DOCUMENTATION PAGE	REPORT NUMBER ISE-TR-88-68

TITLE

Information retrieval based on fuzzy associations

AUTHOR(S)

Sadaaki Miyamoto

REPORT DATE	NUMBER OF PAGES
June 3, 1988	27 + figures
MAIN CATEGORY: Information Storage and Retrieval	CR CATEGORIES: H.3.1, H.3.3

KEY WORDS

Information retrieval, fuzzy associations, algorithms

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 divided into three stages. Each stage has its inherent fuzziness. As typical examples for describing the three stages, we consider a fuzzy association as a generalization of a fuzzy thesaurus on the first stage, a fuzzy inverted fndex on the second stage, and a fuzzy filter on the third stage. Efficient algorithms for fuzzy retrieval on large scale bibliographic databases re developed. A 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 present model.

SUPPLEMENTARY NOTES