ALGORITHMS FOR A FUZZY ASSOCIATION RETRIEVAL

S. Miyamoto

February 1987
WP-87-20

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

FOREWORD

This paper deals with the creation of a thesaurus for information retrieval using fuzzy set theory. The author names the generalization as a fuzzy association. It is shown that the fuzzy association incorporates some current methods of indexing for bibliographic databases. An algorithm to develop the fuzzy association is given. A method of information retrieval through the fuzzy association is developed and two algorithms for this are discussed.

Alexander B. Kurzhanski
Chairman
System and Decision Sciences Program

Algorithms for a Fuzzy Association Retrieval

S. Miyamoto

1. Introduction

Since information retrieval inherently contains fuzzy aspects, fuzzy information retrieval has been studied by many researchers (e.g., Negoita, 1973; Tahani, 1976; Radecki 1979.) Nevertheless, these researchers have concentrated on theoretical aspects and few studies have been devoted to efficient algorithms and practical considerations for fuzzy retrieval. Nowadays various devices and software for databases and information retrieval have been developed. Therefore the theory of fuzzy retrieval should be put into practice by developing efficient algorithms for it.

A typical example of fuzziness in information retrieval is a fuzzy thesaurus. Conventional thesauri of an ordinary type (e.g., ERIC, 1980) can be considered as an α-cut of underlying fuzzy thesauri. Therefore, several researches have been devoted to this (Reisinger, 1974; Radecki, 1976). Moreover, the author has proposed an efficient algorithm for generating a pseudothesaurus for information retrieval that can include over 10,000 different keywords as indices of documents in large-scale databases (Miyamoto et al., 1983).

In this paper a concept of fuzzy association retrieval is proposed. An association for information retrieval incorporates the fuzzy thesaurus and other advanced indices. Namely, in view of recent researches on bibliographic databases, citation indexing, and the clustering of scientific articles (Garfield, 1979) are considered. We show how associations for information retrieval are constructed. Effective algorithms for the construction of the associations are considered.

Moreover, information retrieval through the association is formulated. Two algorithms for fuzzy retrieval are considered. The first is based on hashing; the second on sorting, which is less efficient than the former, but easier to implement. Retrieved articles in the fuzzy retrieval should be ordered according to their relevance when displayed to a user. That is, a more relevant document should be printed prior to less important ones. Here we have a problem of ordering. Of course, this can be solved using a standard sorting tool. Moreover, we consider a method of classification, since exact ordering is not necessary for fuzzy retrieval in general.

One significant feature of the approach herein is that fuzzy retrieval can be built as an extended feature of current softwares for practical information retrievals. Its implementation is easy, since no modification on the underlying retrieval softwares of the ordinary type are necessary.

2. Associations for information retrieval

Let $D = \{d_1, d_2, \ldots, d_m\}$ be a set of articles in a database. $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_p\}$ are two sets of indices

for articles in D. The sets X and Y may well represent the same set ($X = Y$) or not ($X \neq Y$) but they represent the same type of object (e.g., X and Y may represent two different sets of kewwords). In other cases they may represent two different kinds of objects (e.g., X may represent a set of keywords and Y may represent a set of citations). A function $f(x,y)$: $X \times Y \rightarrow [0,1]$ is called an association when the function shows grades of relationship between x and y, $x \in X$, $y \in Y$.

Before introducing the method of fuzzy retrieval, the construction of $f(x,y)$ must be discussed. For this we need to consider specific situations. Therefore, we show in several examples how the associations are used in information retrieval.

2.1 Fuzzy thesaurus

A thesaurus in information retrieval is a sort of dictionary in which a title word is given with its associated words. Associated words are grouped into several categories. A typical example of a part of a thesaurus is given in Figure 1, in which the category BT means broader terms. For example, "Logic" has a broader meaning than the title "MATHEMATICAL LOGIC". The category NT means narrower term: "Alghorithms" has a narrower meaning than the title. The category RT means related terms: "Computational Linguistics" is related to the title word.

A thesaurus can be formalized by introducing two binary relations. Let X be a set of keywords and f_R and f_N be binary relations: f_R, f_N: $X \times X \rightarrow \{0,1\}$. For $x,y \in X$, $f_R(x,y) = 1$ iff x and y are related terms or, in other words, x is a related term

of y and y is also a related term of x; $f_N(x,y) = 1$ iff x is a narrower term of y. Moreover, we assume that BT is the inverse relation of NT: $f_N(x,y) = 1$ iff y is a broader term of x. Note that $f_R(x,y) = f_R(y,x)$.

MATHEMATICAL LOGIC

UF Symbolic Logic
NT Algorithms
Mathematical Formulas
Set Theory
BT Logic
Mathematics
RT Computational Linguistics
Game Theory
Matrices
Statistics

MATHEMATICAL MODELS

BT Models
RT Diagrams
Information Theory
Mathematical Applications
Mathematical Concepts
Mathematical Formulas

Figure 1

Conceptually, it is easy to consider a fuzzy thesaurus as an extension of the above relations. Therefore we consider f_R and

f_N as fuzzy relations: f_R, f_N: X x X -> [0,1] hereafter. Fuzzy thesauri as fuzzy relations have been considered in several works (Reisinger, 1974; Radecki, 1976). A fuzzy thesaurus may be defined directly by some manual procedure. On the other hand, frequently a thesaurus can be automatically generated or determined by a semiautomatic method (Salton, 1971) through a mathematical model that justifies the generation procedure.

For example, in an earlier paper the author proposed a model of concept space to generate a fuzzy thesaurus or a fuzzy pseudothesaurus (Miyamoto et al., 1983). We briefly describe the model here, since it is applicable to any of the later examples in this section.

Let $Z = \{z_1, z_2, \ldots, z_q\}$ denote a set of various concepts. A function $h : X \to [0,1]^Z$ is assumed to be given; A measure M is assumed to be defined on subsets in Z. In the sequel we assume that M is a measure of counting the number of elements on a fuzzy subsets: if $h(x_i) = h_{ij}$ on z_j, then $M(h(x_i)) = \sum_j h_{ij}$. There are different ways of defining f_R and f_N in this model. Here they are defined as (Miyamoto et al., 1983):

$$f_R(x_i, x_j) = \frac{M(h(x_i) \cap h(x_j))}{M(h(x_i) \cup h(x_j))}$$

$$f_N(x_i, x_j) = \frac{M(h(x_i) \cap h(x_j))}{M(h(x_i))} \qquad (1)$$

In practice the set Z should be replaced by other sets. In the semiautomatic method of Salton (Salton, 1971) concepts of

keywords are expressed by a set of characteristics that a keyword may or may not have. Specialists are asked to what extent these characteristics are relevant to a set of keywords. The results of the questionnaire are accumulated to define the function h. In short, in the semiautomatic method a substitute for Z of the concept set is considered. On the other hand, in the fully automatic method of Salton (Salton, 1971), and also in the method proposed by the author (Miyamoto et al., 1983), the set Z is replaced by the set D of articles. In the latter case the function h is defined by frequency of occurrence of a keyword in the articles. Specifically, if the frequency of occurrence of the keyword x_i in the article d_j is represented as h_{ij}, then we can define $h(x_i) = \{h_{i1}/N, h_{i2}/N, \ldots, h_{im}/N\}$, where N is a sufficiently large number such that $0 \leq h_{ij}/N \leq 1$ for all i,j. Note that it is not necessary to choose actual value for N, since it disappears in the definition of f_R and f_N when the set operations are defined by the minimum and maximum. In the latter case the generated relation is called a pseudothesaurus (Miyamoto et al., 1983).

Actual calculation of the values for f_R and f_N seems to require a large amount of computation when the numbers of keywords and articles are large. However, we have an efficient algorithm which will be described later.

2.2 Bibliographic citations

Recently, one of the main interests in the field of bibliographic databases has been citations. Bibliographic

citations have been studied as indicators of scientific activities (Garfield, 1979). Moreover, clustering of articles in the Science Citation Index (SCI) has been considered and advanced indices based on bibliographic citations have been developed. In the clustering of articles based on citations, first a similarity measure on an arbitrary pair of cited articles should be defined in a database. The similarity measure is based on co-citations, which means that two cited articles have the same citing articles. Therefore, if the number of co-citations between a pair of cited articles is large, then the pair will have a large similarity value. A cluster of generated literature based on the co-citation measure has been considered as being a group of literature of special interest that cannot be clarified by other types of ordinary indices. Therefore, the groups obtained by clustering can serve as a kind of advanced indice (Garfield, 1979).

The above considerations can be discussed in the framework herein. Moreover, the method herein suggests the possibility of association between a pair of clusters.

Let X be a set of citations (cited articles) and D be the set of citing articles. The function $h : X \to [0,1]^D$ represents the grade of relevance of each citation to the citing articles. Note that the function h can be derived from the frequencies of occurrence of citations in the set of citing articles, as in previous work. Then it is easy to obtain a similarity measure, $s : X \times X \to R^+$, based on the function h. For example, we can consider the measure f_R in the previous example:

$$s(x_i, x_j) = \frac{M(h(x_i) \cap h(x_j))}{M(h(x_i) \cup h(x_j))} \qquad (2)$$

There are many clustering procedures applicable to the set of citations and we omit the detail (cf. Garfield, 1979). We assume here that clusters of citations have already been generated. Let $\bar{x}_1 = \{x_{11}, \ldots, x_{1v}\}, \ldots, \bar{x}_K = \{x_{K1}, \ldots, x_{Kw}\}$ be the groups generated by the clustering. We do not assume that $\bar{x}_i \cap \bar{x}_j = \emptyset$ nor $\cup \bar{x}_i = X$. These subgroups of citations can serve as a fuzzy index to the source articles in the database as follows. If we assume that $h(x_i) = \{h_{i1}, \ldots, h_{im}\}$, then it is natural to define a fuzzy indexing function $g : \{\bar{x}_i\} \rightarrow [0,1]^D$ as

$$g(\bar{x}_i) = \{ (1/N) \sum_{\text{all } x_k \in \bar{x}_i} h_{k1}, \ldots, (1/N) \sum_{\text{all } x_k \in \bar{x}_i} h_{km} \}, \quad i=1,..,K. \qquad (3)$$

where N is the normalizing constant such that $0 \leq \Sigma h_{kt} \leq 1$ for all t, i. Again we need not determine the value of N, as shown below. That is, an association f between a pair of clusters in $\{\bar{x}_i\}$ can be defined using the above g:

$$f(\bar{x}_i, \bar{x}_j) = \frac{M(g(\bar{x}_i) \cap g(\bar{x}_j))}{M(g(\bar{x}_i) \cup g(\bar{x}_j))} \qquad (4)$$

$$(f : \{\bar{x}_i\} \times \{\bar{x}_j\} \rightarrow [0,1])$$

2.3 Association between keywords and citations

The above two examples deal with associations between a pair of elements taken out of the same set, whereas the third deals

with an association defined on two sets of different type. We assume that X is a set of keywords and Y is a set of citations. Let k be a fuzzy keyword index function ($k : X \to [0,1]^D$) and h be a fuzzy citation index ($h : Y \to [0,1]^D$). Then we define $f : X \times Y \to [0,1]$:

$$f(x,y) = \frac{M(k(x) \cap h(y))}{M(k(x) \cup h(y))} \qquad (5)$$

Remark. When k and h are determined by the frequency of occurrence, the normalizing constants should be considered for both functions. If a common constant is chosen for h and k, then the constant disappears in the definition of f. On the other hand, if two different constants (e.g., N_k for k and N_h for h) are chosen, then we should determine appropriate values for them.

The above three example show how recent considerations of advanced indices for bibliographic databases can be discussed in the present framework of fuzzy association.

2.4 Algorithm for generating the associations

Now, an algorithm to generate the above associations should be considered. For simplicity, the algorithm for computing f_R and f_N (equation (1)) is given here, but the algorithm is applicable to other examples with slight modifications.

In the description of the algorithms below, the symbol (a,b,c) means a record with fields a, b,and c. The symbol

{(a,b,c)} means a set of records. A function Ad(x) means the address determined by a hashing function (Knuth, 1973) for a given x. Two kinds of hashing functions are used. At each address $Ad((x_i,x_j))$, $x_i,x_j \varepsilon X$, a record of two integers (N_{ij},X_{ij}) is stored; at each address $Ad(x_i)$, $x_i \varepsilon X$, a record (F_i) is stored.

Algorithm for generating a fuzzy association:

Input: a set of articles including the keywords in X.

Output: The set of values for the fuzzy association $f(x_i,x_j)$ for all pairs $x_i,x_j \varepsilon X$.

A1.(Initialize) for all x_i and x_j, compute $Ad((x_i,x_j))$ and $Ad((x_i))$;

at $Ad((x_i,x_j))$ $N_{ij} \leftarrow 0$; $X_{ij} \leftarrow 0$;

at $Ad((x_i))$ $F_i \leftarrow 0$;

A2. for all articles represented by k

for all x_i and x_j that occur in article k make records

$(x_i,x_j,f_i,\min(f_i,f_j),\max(f_i,f_j))$;

[Note: $x_i < x_j$ in lexicographic order, f_i means frequency of occurrence of x_i in k]

at $Ad((x_i,x_j))$ $N_{ij} \leftarrow N_{ij} + \min(f_i,f_j)$;

$X_{ij} \leftarrow X_{ij} + \max(f_i,f_j)$;

at $Ad(x_i)$ $F_i \leftarrow F_i + f_i$;

A3. for all x_i and x_j

$f_R(x_i,x_j) \leftarrow N_{ij}/X_{ij}$;

$f_N(x_i,x_j) \leftarrow N_{ij}/F_i$;

End of algorithm.

If we denote the average number of keywords per article by q, it is easy to see that the amount of computation is Order($mq^2/2$), where m is the number of articles. Since all the pairs of occurrences of the keywords in the set of articles is equal to $mq^2/2$, it is obvious that the above algorithm is optimal.

Remark. The problem of collision in a hashing is not considered here for simplicity (Knuth, 1973). []

3. Information retrieval through associations

Let us remember that $D = \{d_1, d_2, \ldots, d_m\}$ is a set of articles, $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_p\}$ are two sets of indices, and $f(x,y)$, $x \in X$, $y \in Y$ is a fuzzy association.

A function of indexing, $T : D \to 2^X$ is assumed to be given. That is, $T(d)$, $d \in D$ represents indices in X for a given article. A function $U : X \to 2^D$ is an inverse of T. Namely, $d \in U(x)$ iff $x \in T(d)$. The function U is realized as an inverted file in an actual implementation.

Remark. In the previous section a set D of articles was used to generate associations. The set D there and here represent, in general, different sets, although the same symbol D is used. []

Now, a fuzzy extension $T_f : Y \times D \to [0,1]$ of the indexing function is considered:

$$T_f(y,d) = \sup_{x \in T(d)} f(x,y) \qquad (6)$$

The fuzzy set $\{T_f(y,d)\}$, $y \in Y$, represents the extended indices for the article d through the association f.

Then consider an inverse relation U_f:

$$U_f(d,y) = T_f(y,d) \tag{7}$$

The function U_f is called here a fuzzy retrieval function. Note that the following equation holds:

$$U_f(d,y) = \sup_{\substack{d \in U(x) \\ \text{all } x \in X}} f(x,y) \tag{8}$$

To prove the above equation, note that

$$\sup_{x \in T(d)} f(x,y) \geq f(x',y) \text{ for any } x' \in X \text{ such that } d \in U(x')$$

and

$$\sup_{\substack{d \in U(x') \\ \text{all } x' \in X}} f(x',y) \geq f(x'',y) \text{ for any } x'' \in T(d)$$

since $d \in U(x')$ iff $x' \in T(d)$. Taking the suprema of the right-hand sides of the above inequalities, we have

$$\begin{aligned} U_f(d,y) = T_f(y,d) = \sup_{x \in T(d)} f(x,y) &\geq \sup_{\substack{d \in U(x') \\ \text{all } x' \in X}} f(x',y) \\ &\geq \sup_{x'' \in T(d)} f(x'',y) = T_f(y,d) = U_f(d,y) \end{aligned} \tag{9}$$

The equation (8) shows that fuzzy retrieval can be implemented through an inverted file and a file of the fuzzy associations. This means that fuzzy retrieval is easily implemented on an existing crisp type of information retrieval system by the simple addition of a file of fuzzy association.

Remark. We have determined a fuzzy index by defining the supremum (6). An alternative definition of the fuzzy index is possible, for example, by considering the average values of f(x,y) for all $x \varepsilon T(d)$. In the alternative definition we can derive an equation like (8) and the fuzzy retrieval can also implemented through the inverted file U and the file of the fuzzy association. []

Now, an algorithm for fuzzy retrieval is described. In the following algorithm, retrieved articles are displayed according to the order of relevance determined by the fuzzy association: an article with a larger value of f should be printed prior to that with a smaller value of f.

Algorithm for fuzzy retrieval (outline):
Input: a keyword y ($y \varepsilon Y$).
Output: the set of articles with the values of association $\{(d,U_f(d,y))\}$ where $U_f(d,y) \neq 0$. The retrieved articles are ordered according to decreasing order of U_f.

R1. Determine values of $U_f(d,y)$ for all documents d with $U_f(d,y) \neq 0$ and make a set of records $\{(d,U_f(d,y))\}$.
R2. Sort the set of records $\{(d,U_f(d,y))\}$ by the key $U_f(d,y)$.
End of algorithm.

The two parts of the above outline of the algorithm should be described in detail. In the following we adopt a natural assumption that the number m of the articles may be large (e.g., 10^5) but the numbers of elements in X and Y are not very large (e.g., 10^3), and that the average number of x, which has positive

values of f ($f(x,y)\neq 0$) for an arbitrary y (denoted by t) is not large (e.g., Order(10)). Moreover we assume that the average number of articles in U(x) for an arbitrary x (denoted by u) is not large: e.g., u = Order(10).

An optimal algorithm for the first part, R1, uses hashing. We use the same symbol Ad(.) for the hashing function as that in the algorithm for generating the fuzzy association.

R1.0 (Assumption) Let us assume that at each record Ad(d), $d \varepsilon D$, there is a record of the form $(d,U_f(d,y))$.

R1.1 (Initialize) at each Ad(d), $d \varepsilon D$, let $U_f(d,y) \leftarrow 0$;

R1.2 find all x with $f(x,y)\neq 0$ using a fuzzy association file;

R1.3 for all x found in R1.2

    find all $d \varepsilon U(x)$;

    at Ad(d), $U_f(d,y) \leftarrow \max(U_f(d,y),f(x,y))$;

R2.1 accumulate all the records $\{(d,U_f(d,y))\}$ with $U_f(d,y)\neq 0$;

R2.1 sort the records $\{(d,U_f(d,y))\}$ into decreasing order with the key U_f;

R2.3 display the records according to the sorted order;

End of algorithm.

In the procedure R1.2 - R1.3 the amount of computation is order(tu). The procedure is optimal, since we should examine all x with $f(x,y)\neq 0$ and all articles $d \varepsilon U(x)$. A problem in the above algorithm lies in R1.1 and R2.1, which require examination of every location of all d. In general, these parts do not require much computation time when techniques are used that depends on

particular computer hardwares. When procedures R1.1 and R2.1 are considered to be time consuming, however, a sorting algorithm can be applied (Miyamoto and Nakayama, 1986).

R1.1' for all x with $f(x,y) \neq 0$
    for all $d \in U(x)$
        make record $(d, f(x,y))$;

R1.2' sort $\{(d, f(x,y))\}$ in dewcreasing order by the first key d and the second key $f(x,y)$;

R1.3' scan the sorted sequence $\{(d, f(x,y))\}$, for any subsequence of records that represents the same instance of d, take the first record and delete the rest of the records in that subsequence. [Note: the first record in the above subsequence represents the value of $U_f(d,y)$.]

R2.1' sort the sequence into decreasing order with the key f.

R2.2' display the records according to the sorted order.

End of algorithm.

The procedure R1.1' - R1.3' requires more computation than R1.2 - R1.3, but the latter algorithm does not have any subprocedure such as R1.1 and R2.1.

In the second part, R2, of the algorithm, frequently exact ordering is not necessary. Instead, it is necessary to classify the retrieved articles into several classes of grade of relevance, e.g., a class of high relevance, one of medium relevance, and so on. Specifically, the interval [0,1] should be subdivided into subintervals $[0, \alpha_1]$, $(\alpha_1, \alpha_2], \ldots, (\alpha_{k-1}, 1]$.

Retrieved articles with $\alpha_{i-1} < U_f(d,y) \leq \alpha_i$, $i=1,..,k$, forms the class C_i. Users of fuzzy retrieval may output the last class, C_k, or the last two C_k, C_{k-1}, and so on. If the number of retrieved articles is equal to β, it is obvious that the above classification needs a computational time of Order(β), whereas an exact ordering requires that of Order($\beta \log\beta$).

A problem in the above classification is to determine parameters α_i, $i=1,..,k-1$. To consider this problem let us assume that articles in $C_\ell,...,C_k$ are displayed. Then the following two cases should be considered.

(1) Articles in C_i are ordered by sorting in class C_i ($\ell \leq i \leq k$)

(2) Articles in C_i are randomly chosen one by one and printed, without sorting.

Let us tentatively adopt the unrealistic assumption that we can control the number, β_i, of articles in each class C_i arbitrarily by an appropriate choice of the parameters $\alpha_1,..,\alpha_{k-1}$. Then we can show for both (1) and (2) that the choice of equal numbers of articles in all the classes $\beta_1 = \beta_2 =...= \beta_k$ is optimal in some sense.

In case (1), the articles C_ℓ - C_k should be sorted. Since the value of cannot be determined beforehand, we should take ℓ=1. An exact ordering without the above classification needs Order($\beta \log\beta$) computation and sorting in class C_i needs Order($\beta_i \log\beta_i$). We can show the inequality

$$\sum_{i=1}^{k} \beta_i \log\beta_i \leq \beta \log\beta \qquad (10)$$

without difficulty. Moreover, the solution of

$$\min \sum_{i=1}^{k} \beta_i \log \beta_i \qquad (11)$$

$$\text{subject to} \quad \beta_1 + \beta_2 + \ldots + \beta_k = \beta,$$

$$\beta_i \geq 0, \quad i=1,2,..,k$$

is given by $\beta_1 = \beta_2 = \ldots = \beta_k$. The proof is a simple application of the Lagrange multiplier and is omitted here.

Next, let us consider (2). In what sense are the subintervals with $\beta_1 = \beta_2 = \ldots = \beta_k$ optimal? Here the articles in C_i are randomly chosen one by one and sequentially printed. An article may be printed at the end of the sequence in which members are randomly chosen in C_i. If articles in C_i are sorted, such an article may be printed at the beginning of the sequence. Thus, when we compare the order of articles in the sorted sequence and the order generated by random choice, the maximum difference for an article between the two orders is equal to $\beta_i - 1$. Therefore, the maximum difference in the whole sequence given by $C_1,\ldots,C_k$, is $\max_{1\leq i\leq k} (\beta_i - 1)$. The it is clear that the solution of

$$\min_{\beta_i} \max_{1\leq i\leq k} (\beta_i - 1)$$

$$\text{subject to} \quad \beta_1 + \beta_2 + \ldots + \beta_k = \beta,$$

$$\beta_i \geq 0, \quad i=1,2,..,k$$

is given by $\beta_1 = \beta_2 = \ldots = \beta_k$.

Remark. In general, a large number of articles is found in one

trial with fuzzy retrieval through fuzzy associations. Therefore a user should be informed of the number of articles in each class and of the values of the parameters $\alpha_1, \ldots, \alpha_{k-1}$ before display of the retrieved articles. Moreover, since we cannot know the exact distribution of U_f for a set of retrieved articles beforehand, some method of estimating the distribution to give classifications with equal numbers of articles in each class, in an approximate sense, is necessary. []

4. Conclusion

The term association retrieval has been used in the field of library science (Lancaster, 1972). Here it is shown that various kinds of associations can be considered within a framework of fuzzy sets. Moreover, efficient algorithms for generating the associations and for retrieval through the associations are proposed. As noted earlier, the method of fuzzy retrieval can be implemented on practical information retrieval systems. Its implementation is easy with no drawback for the underlying crisp system. Furthermore, classification of the retrieved articles into several classes of grades of relevance is considered. Even in conventional thesauri of the crisp type, a retrieval through the thesaurus may lead to many articles that should be printed. Therefore, the above consideration suggests the introduction of a measure of relevance even for a conventional thesaurus.

References

ERIC (1980) Thesaurus of ERIC Descriptors. (Oryx, Phoenix, AZ).

Garfield, E. (1979), Citation Indexing - Its Theory and Application in Science, Technology, and Humanities. (Wiley, New York).

Knuth, D. E. (1973), The Art of Computer Programming, Vol. 3, Sorting and Searching. (Addison-Wesley, Reading, MA).

Lancaster, F. W. (1972), Vocabulary Control for Information Retrieval. (Information Resources, Washington DC).

Miyamoto, S., Miyake, T., and Nakayama, K. (1983), Generation of a pseudothesaurus for information retrieval based on co-occurrences and fuzzy set operations, IEEE Trans., Syst., Man, and Cybern., 13 (1) 62-70.

Miyamoto, S. and Nakayama, K. (1986), Fuzzy information retrieval based on fuzzy pseudothesaurus, IEEE Trans., Syst., Man, and Cybern., 16 (2) 278-282.

Negoita, C. V. (1973), On the application of the fuzzy set separation theorem for automatic classification in information retrieval systems, Information Sciences, 5, 279-286.

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

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

Reisinger, L. (1974), On fuzzy thesauri, in G. Bruckman et al., (eds.) COMPSTAT 1974, Proc., Symp., Comput., Stat., (Physica-Verlag, Vienna), 119-127.

Salton G. (Ed.) (1971), The SMART Retrieval System, Experiments

in Automatic Document Processing, (Prentice-Hall, Englewood Cliffs, NJ).

Tahani, V. (1977), A conceptual framework for fuzzy query processing - a step toward very intelligent database systems, Information Processing and Management, 13, 289-303.