Information Processing & Management Vol. 26, No. 2, pp. 209-218, 1990
Printed in Great Britain.

0306-4573/90 $3.00 + .00

DOCUMENT RETRIEVAL: A STRUCTURAL APPROACH

XIN LU
School of Library and Information Science, University of Western Ontario,
London, Ontario, Canada

(*Received* 4 *January* 1989; *accepted in final form* 4 *April* 1989)

Abstract—This paper describes a structural document retrieval model which has been designed based on lexical-semantic relationships between index terms and an algorithm of measuring tree-to-tree distance. In this model, documents and query statements are structurally coded in order to take into account any hierarchy or ordering among the conceptual coordinates and are structurally matched by using the algorithm that cannot be expressed in a form of equation. The proposed model has been compared to the vector retrieval model using a small database and the results have been analyzed using a precision-recall graph and a statistical test. Both the graph and the testing result suggest that on this small database the proposed model tends to improve retrieval effectiveness. However, the structural retrieval model needs to be refined and more elaborate experiments are required in order to further confirm the findings.

1. INTRODUCTION

In an ideal environment of document retrieval, a document or a query statement is represented by a group of distinct index terms as well as the semantic relationships betwfen those terms so that retrieval could be directly conducted on a structure of semantic relationships. However, those relationships have been virtually neglected in various retrieval models because of the limited knowledge of natural language processing. For example, the inverted file design merely maintains and processes non-semantic Boolean relations; the vector model and probabilistic retrieval model, on the other hand, define no relationships between index terms and assume that the coordinate axes representing the distinct terms are orthogonal. Although these retrieval models can be further refined and developed, they could not lead to a breakthrough in document retrieval without looking at index term relationships [1].

The use of both index terms and their conceptual relationships in document retrieval is not a new proposal. For example, the SYNTOL group [2] in the early 60's used the four syntactic relations, i.e., coordinative, consecutive, associative and predicative, to connect the SYNTOL words or the main ideas expressed in a document. Pairs of SYNTOL words linked with one of the four relations are called syntagmas. Since the same SYNTOL words may appear in different syntagmas, a document or a query statement can be represented as one or more directed graphs. The matching procedure in the SYNTOL system requires that all syntagmas occurring in a given query should be present in the retrieved documents, unless they are part of OR or NOT Boolean connectives. But the experiments have shown a disappointing performance of the SYNTOL system [3]. The poor performance might be attributed to the automatic indexing mechanism which uses a limited syntactic analysis procedure and, more fundamentally, the four relations which are perhaps unnecessarily explicit and have very limited capacity to link index terms into meaningful structures.

Another example of using index terms and their conceptual relations to represent and to retrieve documents is the SMART project [4]. In the SMART system, syntactic relationships between index terms, i.e., noun phrase, subject-verb, verb-object and subject-object, are determined after performing a syntactic analysis on the sentences occurring in documents and query statements. The results of this process are then normalized by looking up a dictionary of criterion trees. Criterion trees are predetermined frames including concepts, syntactic indicators and the syntactic relationships between the concepts. Since different concepts are attached to one node of a tree, one criterion tree can correspond to many different syntactic structures in English and can be encoded and stored in a matrix. With such a matrix, a matching procedure can be developed to determine whether the query graph is a subgraph of the document graph. However, the comparative experiments have shown that syntactic information contributes little to retrieval effectiveness [5]. One possible explanation of the adynamic performance is that the tools used for syntactic analysis are not powerful enough. Specifically, phrase structure grammar used for parsing is a simple grammar with many functional limitations; the context-sensitive dictionary of criterion trees only has a limited subject coverage and thus some information might be simply not identified during indexing. Another possible explanation is that the syntactic structures or criterion trees are poor indicators of semantic contents of documents.

The third and the last example is the relational indexing system [6,7]. In this system, the nine conceptual relations such as concurrence, association and functional dependence derived from a psychological theory of thinking are used to connect pairs of concepts in a given document. When this indexing procedure is applied to every pair of concepts of the document, one conceptual graph is created as the document representation. The retrieval procedure can be computerized by converting the graphs into connection tables. But the overall retrieval performance of the relational indexing system is not very encouraging. The major problem might be the insufficiency of these nine relations in representing documents from all fields.

The three reviewed studies provide one suggestion for the structural approach to document retrieval: the use of pure syntactic relations and those conceptual relations derived from the psychological theory is probably not sufficient for representing the contents of documents. In the present study, the approach to document retrieval is therefore to use lexical-semantic relations to represent documents. "Lexical relations encapsulate the necessary semantic information in a compact and convenient fashion" [8]. Their applications in various disciplines indicate that lexical-semantic relations are valuable in question-answering [9], in representing and using real-world knowledge [10], and in text generation [11]. In the field of document retrieval, a number of experimental studies have shown their potential for retrieval applications.

In investigating the possibility of enhancement of document description records by adding keywords from end-users' queries, Tague [12] found that 29% of the title-user keyword pairs which appeared to exhibit a non-transient lexical-semantic relationship did not occur as title-title keyword pairs. Thus, it is valuable to expand document description records with those selected keywords from the queries for which the documents have proved relevant. The process of selection can be based on lexical-semantic relationships between keywords.

Instead of trying to enhance document description records, Fox [13] conducted an experimental study to see whether queries could be enhanced by adding extra terms which have lexical-semantic relationships with those terms in the queries and the expanded queries would lead to better retrieval effectiveness. Unlike Tague's study, Fox included a comprehensive list of lexical-semantic relations and categorized them into eleven main groups. He also proposed the effect of each relation category in terms of improving recall or precision. Following are some of his interesting findings:

1. using all lexical relation categories yields important improvements (up to 16.5% in precision) over the original non-enhanced queries;
2. adding all categories except for antonyms gives even better performance (up to 20% improvement);
3. use of the antonym category has a uniform negative effect on performance;
4. the behavior of predicates and paradigmatic relations are not clear.

Four years later, a more elaborately controlled experimental study [8] further confirmed Fox's findings. However, all of these studies merely attempted to use lexical-semantic relations to help select significant terms to enhance either doccument or query descriptions and ultimately to improve retrieval performance. They did not directly use lexical-semantic relations in representing and retrieving documents. They aimed at no more than refining the existing retrieval models.

The following sections describe the development and the test of a structural model of document retrieval which, to a certain extent, does take account of both index terms and their semantic relationships. The model applies the current automatic indexing technique to extract index terms from documents or their surrogates, manually connects those terms with their lexical-semantic relationships, and finally uses tree-to-tree distance to measure structural closeness between a document and a query statement instead of the exact-matching methods found in the earlier research.

2. LEXICAL-SEMANTIC RELATIONS

All lexical-semantic relations used in this study are grouped into the five categories [14] in Table 1. This list is not as comprehensive as that used by Fox [13] and Wang [8]. The author considers that too many relations would blur their differences, making them difficult to use accurately. The antonym category has been excluded because of its negative effect on retrieval effectiveness. Most of the collocation relations and paradigmatic relations have also been excluded since their usefulness in improving retrieval effectiveness is not clear.

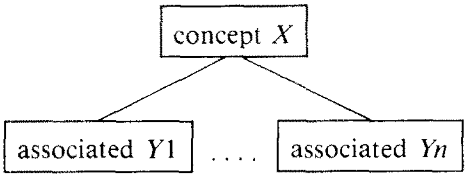
The lexical configuration behind the lexical-semantic relations of group 2, 3, 4 is hierarchy. Two structural types of hierarchy may be distinguished: those which branch, and those which are not capable of branching. In the following descriptions and discussions, these two types of hierarchy are simply referred to as "tree" and "chain," respectively. A special case of hierarchical structure is an individual point (one node tree). This type of

Table 1. Lexical-semantic relations

1. Synonymy		
Cognitive Synonymy	*c-s*	fiddle : violin
Near Synonymy	*n-s*	citation : reference
Morphology Variation	*m-v*	man : men
2. Taxonomy		
Taxonymy	*t-x*	horse : stallion
Co-Taxonymy	*t-x*	ewe : ram
3. Part-Whole		
Meronymy	*p-w*	arm : hand
Co-Meronymy	*p-w*	palm : figure
Group-member	*p-w*	senate : senator
Class-member	*p-w*	proletariat : worker
Collection-member	*p-w*	forest : tree
Entity and its features	*p-w*	cloth : size
Made of	*p-w*	tire : rubber
Comes from	*p-w*	milk : cow
Piece	*p-w*	lump : sugar
4. No*n*-*b*ranching Hierarchy		
Chain	*n-b*	shoulder : upper arm : elbow : forearm
Helices	*n-b*	spring : summer : autumn : winter
Rank-terms	*n-b*	single : double : triple
Grade-terms	*n-b*	small : big : huge
Degree-terms	*n-b*	fail : pass : credit : distinction
5. Other Semantic Relations		
Entity and associated process(es)	*e-pc*	library : circulation
Entity and associated entity(ies)	*e-e*	car : garage
Entity and asociated person(s)	*e-ps*	hospital : patient
Process and associated entity(ies)	*pc-e*	automation: computer
Process and associated process(es)	*pc-pc*	indexing : decision-making
Process and associated person(s)	*pc-ps*	indexing : indexer
Person and associated process(es)	*ps-pc*	user : searching
Person and associated entity(ies)	*ps-e*	librarian : book

structure is specifically named as "point" to emphasize that there are some elements which have no relationships with other elements within a set.

The lexical relation of synonymy is not directly used in representing documents but is used in building a synonym dictionary required for the matching process. The semantic relations in group 5 have no natural configuration. In order to have a unified document representation, they are converted into the following artificial structure:



X should be one of those central topics discussed in a document which is to be represented. X is a process (pc), an entity (e), or a person (ps). $Y_1, \ldots, Y_n$ are associated processes, entities, or persons used for the discussion of X.

3. DOCUMENT REPRESENTATION

Definition 1. A document is a finite set of elements called concepts which are in the form of a word or term; the document is denoted by **D**, the concepts by $c_1, c_2,, \ldots c_n$ $(n > 0)$.

Definition 2. A concept point S^p is a single concept which has no lexical-semantic connections with any other concepts within **D**.

Definition 3. A concept chain S^c on **D** is a vertically ordered sequence of elements of **D**. The link in S^c is an indicator of one non-branching hierarchical lexical relationship between two neighbour concepts.

Definition 4. A concept tree S^t on **D**

1. contains a unique concept c_r that can be distinguished from all others, called the root of the tree;
2. all other concepts, if any, can be grouped into disjoint sets $T_1, T_2, \ldots, T_n$ which are themselves trees and for each of which the distinguished concept is linked to c_r by a single arc. The arc is in fact an indicator of a lexical-semantic relation.

With these definitions, one document (or query statement) can be represented as

$$D = \{\{S_1^t, \ldots, S_x^t\}; \{S_1^c, \ldots, S_y^c\}; \{S_1^p, \ldots, S_z^p\}\}$$
$$0 \leq x, y, z \leq n; x + y + z > 0$$

where S_i^t is one of S^{t-x}, S^{p-w}, S^{e-pc}, S^{e-e}, S^{e-ps}, S^{pc-e}, S^{pc-pc}, S^{pc-ps}, S^{ps-pc} and S^{ps-e}. In the following discussion, all structures, i.e., S^t, S^c and S^p, are regarded as tree structures.

4. THE MATCHING FUNCTION

Given two tree structures of the same type, T and T', their structural similarity can be measured by calculating the *editing* cost of transforming T into T', a technique developed in the field of pattern recognition to measure the distance between two trees. The actual algorithm to be used has been developed by Selkow [15]. Three editing operations have been defined, namely change of label, insertion of subtree, and deletion of subtree. A non-negative cost is associated with each operation. Thus the transformation of T into T' is made in a sequence of operations, each of which incurs a cost. The distance $\delta(T, T')$ between T and T' is the *minimum total cost of transforming T into T'*.

The complexity of this algorithm can be calculated by defining the *signature* of a tree as the vector $(t_1, \ldots, t_i, \ldots, t_d)$ where t_i is the number of nodes of the tree T at level i and d is the deepest level. For trees T and T' with signatures $(1, t_2, \ldots, t_d)$ and $(1, t_2', \ldots, t_{d'}')$ respectively, the number of calculations necessary is $O(\Sigma_{i=1}^{min(d,d')} t_i t_i')$.

If EDIT(T, T') denotes such an editing function, $\delta(T, T')$ is equal to the minimum cost associated with EDIT(T, T'). EDIT(T, T'), the coded Selkow's algorithm, consists of three subfunctions CLAB(a, b), DEL(t) and INS(t), which correspond to changing labels, deleting a subtree and inserting a subtree. DEL(t) and INS(t) can also be called individually. In this research the cost associated with CLAB(a, b) is *two* and the cost associated with DEL(t) and INS(t) is equal to the *size* of t, i.e., the size of a subtree. For instance, if C_L, C_I and C_D represent the cost of a single operation of changing the label of a node, inserting a node and deleting a node, respectively, then $C_L = 2$, $C_D = C_I = 1$.

However EDIT(T, T') cannot be used directly to calculate the similarity between a document (D) and a query statement (Q) since they are represented by a group of tree structures of different types. Given that X_i and Y_i are the numbers of tree structures of type i $(1 \le i \le 12)$ in a document representation and a query representation and that D and Q contain ΣX_i and ΣY_i tree structures, $\delta(D, Q)$, the minimum cost of transforming D into Q, can be calculated by using the following procedure:

```
Procedure   Calculate $\delta(D, Q)$;

Begin
    for $i := 1$ to 12 do
    1. if $(X_i = 0)$ and $(Y_i > 0)$ then
       call the subfunction INS$(t)$ $Y_i$ times to insert $Y_i$ trees in $Q$ into $D$ and add the
       cost to $\delta(D, Q)$;
    2. if $(X_i > 0)$ and $(Y_i = 0)$ then
       call the subfunction DEL$(t)$ $X_i$ times to delete $X_i$ trees from $D$ and add the cost
       to $\delta(D, Q)$;
    3. if $(X_i, Y_i > 0)$ and $(X_i = Y_i)$ then
       for $j := 1$ to $Y_i$ do
           call EDIT$(t, t')$ to transform every tree in $D$ into the $j$th tree in $Q$, add the
           lowest cost among transformations to $\delta(D, Q)$ and logically remove the tree
           which is associated with the transformation of the lowest cost from $D$;
    4. if $(X_i, Y_i > 0)$ and $(X_i < Y_i)$ then
       repeat 3 for $j := 1$ to $X_i$;
       call INST$(t)$ $Y_i$-$X_i$ times to insert the trees left in $Q$ into $D$ and add the cost to
       $\delta(D, Q)$;
    5. if $(X_i, Y_i > 0)$ and $(X_i > Y_i)$ then
       repeat 3 for $j := 1$ to $Y_i$;
       call DEL$(t)$ $X_i$-$Y_i$ times to delete the trees left in $D$ and add the cost to
       $\delta(D, Q)$;
End.
```

When D is entirely different from Q (i.e., given any type of tree, no node in D is identical to any node in Q), $\delta(D, Q)$ becomes maximum and this $\delta(D, Q)_{\max}$ is equal to $\Sigma_{i,j}^{12, Xi}$DEL(t) + $\Sigma_{i,j}^{12, Yi}$INS(t). $\delta(D, Q)_{\max}$ can be used to convert a distance between D and Q into a normalized similarity value. That is,

$$S = 1 - \delta(D, Q)/\delta(D, Q)_{\max} \; (0 \le S \le 1).$$

S is equal to one when D is identical to Q and zero when D is entirely different from Q.

5. EXPERIMENT

A simple experiment was designed to compare this structural model to the vector retrieval model. A database of 79 records was created. Each record consists of an article's title and abstract. The articles were selected from JASIS between 1975 to 1978. They are about the topics of bibliometrics and document retrieval. Similarly, another set of 31 abstracts dealing with the same topics was selected from JASIS and SCIENTOMETRICS of 1980 to 1987 as artificial query statements for testing. Because two of these query statements have no relevant relationships to those 79 records, 29 query statements were actually used in the experiment. Table 2 describes this test database. Judgments of topic relevance were conducted by two Ph.D. students specializing in the two fields. The two indexing mechanisms, conventional automatic indexing and structural indexing, were applied to the database, generating two test databases: the vector database and the structural database. Tables 3–5 describe the characteristics of the two test databases.

The details of creating the vector database and conducting searches are:

1. The individual words that make up a document excerpt or query statement were first recognized.
2. A stop list containing 306 function words was used to eliminate such words.
3. Phrases in each word list were identified manually and frequency information was

Table 2. Collection characteristics

	Document	Query
Number of records	79	29
Total number of different non-stop words	1739	810
Maximum number of non-stop words in a record	90	85
Minimum number of non-stop words in a record	12	24
Average number of non-stop words in a record	49	47
Total number of different stems	1125	576

Table 3. Characteristics of the vector database

Stem Frequency	No. of Stems		Document Frequency	No. of Stems	
	Document	Query		Document	Query
1	513	286	1	597	346
2	170	93	2	162	84
3	90	55	3	90	54
4	64	36	4	73	33
5	47	21	5	50	21
6	33	14	6	27	18
7	36	18	7	23	6
8	20	6	8	15	5
9	17	6	9	12	2
10	22	9	10	12	2
11	11	7	11	10	2
12	7	2	12	6	1
13	9	2	13	4	1
14	10	6	14	7	0
15	4	3	15	6	0
16	6	1	16	4	0
17	6	2	17	2	0
18	4	0	18	2	1
19	2	1	19	4	0
20	5	0	20	3	0
21	5	0	21	2	0
22	5	0	22	4	0
23	5	0	23	2	0
24	1	0	24	1	0
25	2	1	25	3	0
26	3	0	26	0	0
27	3	1	27	0	0
28	3	1	28	0	0
29	4	1	29	0	0
30+	18	4	30+	4	0

Table 4. Characteristics of the structural database

		t-x		*p-w*		*n-b*		Other		
	Size	Doc	Qry	Doc	Qry	Doc	Qry	Doc	Qry	Total
	2	37	13	103	42	0	0	135	71	401
	3	27	9	70	22	1	0	42	30	201
	4	8	0	23	7	3	1	13	11	66
	5	4	0	11	7	1	0	4	2	29
	6	0	0	12	1	0	0	1	0	14
	7+	0	0	9	3	1	0	0	0	13
Total		76	22	228	82	6	1	195	114	724

Table 5. Characteristics of the structural database

	Document	Query
Average number of non-point structures per document	6.4	7.6
Average number of point structures per document	28	25
Average size of non-point structures	2.8	2.7
Average number of synonym groups per document	2.0	2.4

adjusted accordingly. This step is necessary in that the vector prototype in this study is not capable of identifying phrases using co-occurrence information and such a prototype may be biased against the vector model when being compared to the structural model which does maintain phrase information.

4. The scope of the remaining word occurrences was broadened by reducing each word to word stem form; this was done by using a relatively simple suffix removal algorithm [16] together with the special rules to look after exceptions.
5. Following suffix removal and phrase identification, multiple occurrences of a given word stem or phrase were combined into a single term for incorporation into the document or query vectors. Synonyms recognized by the author were processed in the same way.
6. A term weight was assigned to each term in the vector reflecting the usefulness of the term in the collection environment of the experiment; the weighting function [17] is

$$\mathrm{Weight}_{ik} = \frac{Freq_{ik}}{DocFreq_k}$$

7. A cosine matching function was used to calculate similarities between query stem vectors and document stem vectors. The search results were ranked according to similarity value.

The tasks of structural indexing and matching were completed manually by going through the following steps:

1. For each term list created from the first three steps of the preceding indexing process, take a pair of terms each time to see if there is a lexical-semantic relationship between them. If it is the case, the pair of terms is either added into an existing tree structure of same type or organized as a new tree structure.
2. Calculate similarity values manually for each document query pair using the matching function developed and then rank all 79 documents for each query statement in the descending order of similarity value.

Assignment of term pairs to the various categories of the lexical-semantic relations and assignment of the various tree structures to documents and query statements are, of course, subjective on the part of the author. In carrying out these tasks, the author interpreted the meaning of terms with respect to the given subject contexts.

The results from this experiment are presented and compared by means of a recall-precision graph and a statistical test. The graph in Fig. 1 reveals that the performance of the structural retrieval model is superior to the vector model. The signed pair test was used to analyze the results of this experiment in order to see whether the apparent superiority is statistically significant. The null hypothesis for the experiment is that the proposed structural model of document retrieval is not more effective than the vector model. Table 6 gives

Table 6. Statistical test results

Standard recall	Sign test
0.1	0.032
0.2	0.022
0.3	0.067
0.4	0.032
0.5	0.093
0.6	0.013
0.7	0.045
0.8	0.172
0.9	0.163
1.0	0.124

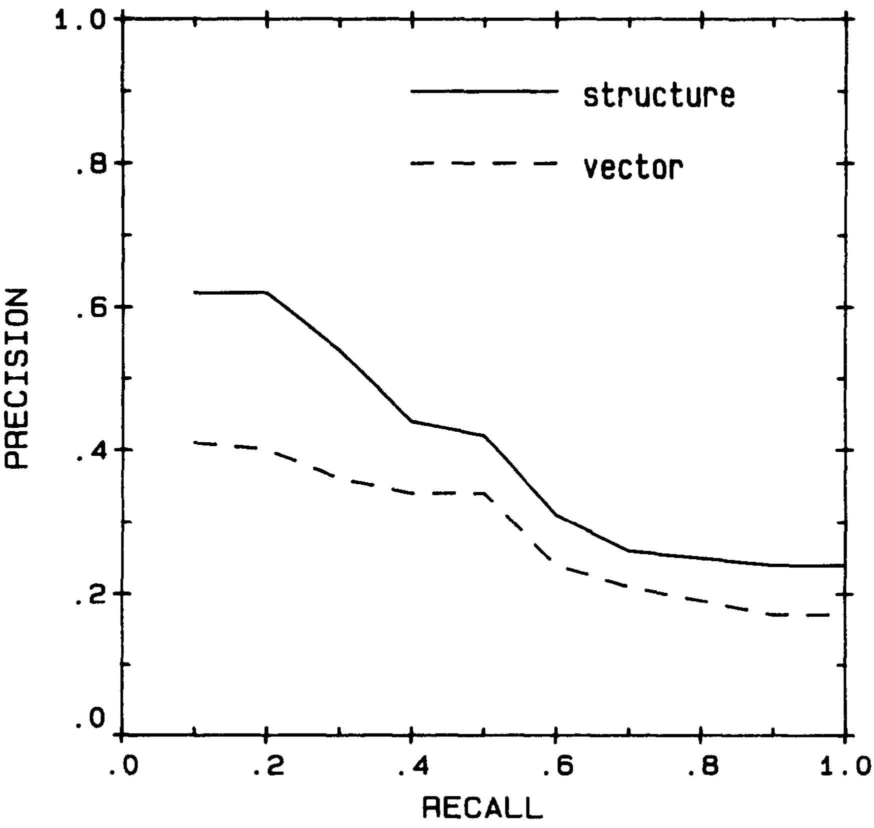


Fig. 1. Recall-precision graph.

the statistical test results, i.e., the probabilities that the null hypothesis is acceptable. Thus, the smaller number reflects that the proposed retrieval model is more effective than the vector model.

6. DISCUSSION

The results of this typical document retrieval experiment demonstrate the potential of using index terms as well as their lexical-semantic relationships to represent and retrieve documents. However, some design defects of this experiment might affect the validity of the results. The size of the test database is small and the number of query statements is small as well. These query statements are artificial ones instead of real ones. The indexing should be conducted by other people rather than the author himself. The proposed retrieval model needs to be further elaborated and a comprehensive experiment with tighter controls is required to confirm the primitive results.

In terms of model development, it might be more meaningful to distinguish the pure lexical relations (e.g., *t-x*, *p-w*) from the pure semantic relations (e.g., *e-pc*, *pr-pc*) and to represent the two groups of relations with different structures such as tree and network. The indexing process in the proposed model perhaps could be accomplished by automatic means to avoid the problem of subjectivity associated with any manual indexing system. Without a function of automatic indexing, the proposed model will have little chance of wide acceptance. It is intriguing to watch the research progress of several scholars who are studying definitions in a machine readable form of dictionaries in the hope of finding methods for extracting lexical-semantical relations automatically [18]. Finally, the retrieval function could be made more efficient based on new algorithms either found or developed.

Although positive results are always important, emphasis at this time is the proposed approach to document representation and retrieval. Document retrieval can be viewed as a process of pattern recognition. One of the most important factors in constructing a retrieval model of this type is the choice of the representation space in which to code document patterns. In particular, the concept of *distance* can be generalized for these spaces and calculated not simply by a formula such as that which defines the Euclidean distance, for example, but by an algorithmic procedure that cannot be expressed in the form of equation and that takes into account any hierarchy or ordering among the coordinates. The model proposed in this study, like those earlier proposed models, represents such structural approach to document retrieval. The experimental results should be seen as an invitation to future development.

Acknowledgments—The author is indebted to the several Ph.D students at the University of Western Ontario for their relevance judgment work and their constructive criticism. My thanks go to Isola Ajiferuke, Dietmar Wolfram and Patricia Burt. My thanks also go to the referees for their helpful comments on an earlier draft of the article.

REFERENCES

1. Van Rijsbergen, C.J. A new theoretical framework for information retrieval. Sigir Forum, 21: 23-29; 1987.
2. Gardin, J.C. Syntol. New Brunswick, New Jersey: The Rutgers University Press; 1965.
3. Sparck Jones, K. and Kay, M. Linguistics and information science. New York: Academic Press; 1973.
4. Salton, G. Automatic phrase matching. In: Hays, D.G., editor. Readings in automatic language processing. New York: American Elsevier Publishing Company Inc.; 1966: 169-188.
5. Salton, G. Automatic information organization and retrieval. New York: McGraw-Hill Book Company; 1968.
6. Farradane, J. Relational indexing, Part I. Journal of Information Science, 1: 267-276; 1980.
7. Farradane, J. Relational indexing, Part II. Journal of Information Science, 1: 313-324; 1980.
8. Wang, Yih-Chen, *et al.* Relational thesauri in information retrieval. Journal of the American Society for Information Science, 36: 15-27; 1985.
9. Evens, M. and Smith, R. A lexicon for a computer question-answering system. American Journal of Computational Linguistics, Microfiches 81: 16-24; 1978.
10. Fahlmann, S. NETL: A system for representing and using real-world knowledge. Cambridge, MA: MIT Press; 1979.
11. Sowa, J. Conceptual structure: Information processing in mind and machine. Reading, MA: Addison-Wesley; 1984.

12. Tague, J.M. User-responsive subject control in bibliographic retrieval systems. Information Processing & Management, 17: 149–159; 1981.
13. Fox, E.A. Lexical relations: Enhancing effectiveness of information retrieval systems. Sigir Forum, 14: 6–35; 1980.
14. Cruse, D.A. Lexical semantics. Cambridge: Cambridge University Press; 1986.
15. Selkow, S.M. The tree-to-tree editing problem. Information Processing Letters, 6: 184–186; 1977.
16. Porter, M.F. An algorithm for suffix stripping. Program, 14: 130–137, 1980.
17. Salton, G. and McGill, M.J. Introduction to modern information retrieval. New York: McGraw-Hill Book Company; 1983.
18. Markowitz, J., Pin-Ngern, S., Evens, M., Anderson, J. and Li, S.M. Generating lexical database entries for phrases. Information in Text, Proceedings of the Fourth Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary; 1988 October 26–28; Waterloo, Canada.