TERM-WEIGHTING APPROACHES IN AUTOMATIC TEXT RETRIEVAL

GERARD SALTON and CHRISTOPHER BUCKLEY
Department of Computer Science, Cornell University, Ithaca, NY 14853, USA

(*Received* 19 *November* 1987; *accepted in final form* 26 *January* 1988)

Abstract—The experimental evidence accumulated over the past 20 years indicates that text indexing systems based on the assignment of appropriately weighted single terms produce retrieval results that are superior to those obtainable with other more elaborate text representations. These results depend crucially on the choice of effective term-weighting systems, This article summarizes the insights gained in automatic term weighting, and provides baseline single-term-indexing models with which other more elaborate content analysis procedures can be compared.

1. AUTOMATIC TEXT ANALYSIS

In the late 1950s, Luhn [1] first suggested that automatic text retrieval systems could be designed based on a comparison of content identifiers attached both to the stored texts and to the users' information queries. Typically, certain words extracted from the texts of documents and queries would be used for content identification; alternatively, the content representations could be chosen manually by trained indexers familiar with the subject areas under consideration and with the contents of the document collections. In either case, the documents would be represented by *term vectors* of the form

$$D = (t_i, t_j, \ldots, t_p) \tag{1}$$

where each t_k identifies a content term assigned to some sample document D. Analogously, the information requests, or queries, would be represented either in vector form, or in the form of Boolean statements. Thus, a typical query Q might be formulated as

$$Q = (q_a, q_b, \ldots, q_r) \tag{2}$$

or

$$Q = (q_a \text{ and } q_b) \text{ or } (q_c \text{ and } q_d \text{ and } \ldots) \text{ or } \ldots \tag{3}$$

where q_k once again represents a term assigned to query Q.

A more formal representation of the term vectors of eqns (1) and (2) is obtained by including in each vector all possible content terms allowed in the system and adding term weight assignments to provide distinctions among the terms. Thus, if w_{dk} (or w_{qk}) represents the weight of term t_k in document D (or query Q), and t terms in all are available for content representation, the term vectors for document D and query Q can be written as

$$D = (t_0, w_{d_0}; t_1, w_{d_1}; \ldots; t_t, w_{dt})$$

and

$$Q = (q_0, w_{q_0}; q_1, w_{q_1}; \ldots; q_t, w_{qt}). \tag{4}$$

This study was supported in part by the National Science Foundation under grants IST 83-16166 and IRI 87-02735.

In the foregoing formulation, the assumption is that w_{dk} (or w_{qk}) is equal to 0 when term k is not assigned to document D (or query Q), and that w_{dk} (or w_{qk}) equals 1 for the assigned terms.

Given the vector representations of eqn (4), a *query-document similarity* value may be obtained by comparing the corresponding vectors, using for example the conventional vector product formula

$$\text{similarity}(Q,D) = \sum_{k=1}^{t} w_{qk} \cdot w_{dk}. \tag{5}$$

When the term weights are restricted to 0 and 1 as previously suggested, the vector product of eqn (5) measures the number of terms that are jointly assigned to query Q and document D.

In practice, it has proven useful to provide a greater degree of discrimination among terms assigned for content representation than is possible with weights of 0 and 1 alone. In particular, term weights in decreasing term importance order could be assigned, in which case the weights w_{dk} (or w_{qk}) could be allowed to vary continuously between 0 and 1, the higher weight assignments near 1 being used for the most important terms, whereas lower weights near 0 would characterize the less important terms. In some circumstances, it may also be useful to use normalized weight assignments, where the individual term weights depend to some extent on the weights of other terms in the same vector. A typical term weight using a vector length normalization factor is $\dfrac{w_{dk}}{\sqrt{\sum_{\text{vector}} (w_{di})^2}}$ for documents $\left(\text{or } \dfrac{w_{qk}}{\sqrt{\sum_{\text{vector}} (w_{qi})^2}} \text{ for queries}\right)$. When a length normalized term-weighting system is used with the vector similarity function of eqn (5), one obtains the well-known cosine vector similarity formula that has been used extensively with the experimental Smart retrieval system [2,3]:

$$\text{similarity}(Q,D) = \frac{\sum_{k=1}^{t} w_{qk} \cdot w_{dk}}{\sqrt{\sum_{k=1}^{t} (w_{qk})^2 \cdot \sum_{k=1}^{t} (w_{dk})^2}}. \tag{6}$$

A vector matching system performing global comparisons between query and document vectors provides ranked retrieval output in decreasing order of the computed similarities between Q and D. Such a ranked output is useful because controls are now available over the size of the retrieved document set, and iterative retrieval strategies based on successive query reformulations are simplified. A system that first retrieves those items thought to be of main interest to the users will necessarily prove helpful in interactive information retrieval.

In designing automatic text retrieval systems, two main questions must be faced. First, what appropriate content units are to be included in the document and query representations? Second, is the determination of the term weights capable of distinguishing the important terms from those less crucial for content identification?

Concerning first the choice of content terms, various possibilities must be considered. In most of the early experiments, *single terms* alone were used for content representation, often consisting of words extracted from the texts of documents and from natural language query formulations. [3–7] In many cases, quite effective retrieval output has been obtained using single-term content representations. Ultimately, however, sets of single terms cannot provide complete identifications of document content. For this reason, many enhancements in content analysis and text indexing procedures have been proposed over the years

in an effort to generate complex text representations. The following possibilities have been considered in this connection:

1. The generation of sets of *related terms* based on the statistical cooccurrence characteristics of the words in certain contexts within the document collection. The assumption normally made is that words that cooccur with sufficient frequency in the documents of a collection are in fact related to each other [8–11].
2. The formation of *term phrases* consisting of one or more governing terms (the phrase heads) together with corresponding dependent terms (the phrase components). Phrases are often chosen by using word frequency counts and other statistical methods, possibly supplemented by syntactic procedures designed to detect syntactic relationships between governing and dependent phrase components [12–17].
3. The use of word grouping methods of the kind provided by *thesauruses*, where classes of related words are grouped under common headings; these class headings can then be assigned for content identification instead of the individual terms contained in the classes [18–20]. Alternatively, term relationships useful for content identification may also be obtainable by using existing machine-readable dictionaries and lexicons [21–24].
4. The construction of *knowledge bases* and related artificial intelligence structures designed to represent the content of the subject area under consideration; entries from the knowledge base are then used to represent the content of documents and queries [25–30].

From the beginning, it was evident that the construction and identification of complex text representations was inordinately difficult. In particular, it became clear that most automatically derived term dependencies were valid only locally in the documents from which the dependent term groups were originally extracted; this implies that dependent term groups could not be counted upon to produce useful content identifiers in new document contexts different from those originally used [11]. The experiences gained with the use of automatically generated term phrases proved similarly discouraging: for some collections, improvements in retrieval effectiveness of up to 20% (in search recall and precision) were obtainable by using phrase identifiers instead of single terms; but for other collections these same phrase procedures did not furnish any improvements at all. Moreover, even sophisticated syntactic analysis programs could not be relied upon to produce useful complex content identifiers [16].

As for the use of preconstructed vocabulary schedules and term classifications, the problem is that viable procedures for the construction of effective vocabulary tools covering subject areas of reasonable scope appear to be completely lacking. The same goes for the construction of knowledge bases designed to reflect the structure of disclosure areas. Until more becomes known about the desired form and content of dictionaries and thesauruses, little gain should be expected from these tools in text analysis and document indexing.

In reviewing the extensive literature accumulated during the past 25 years in the area of retrieval system evaluation, the overwhelming evidence is that the judicious use of single-term identifiers is preferable to the incorporation of more complex entities extracted from the texts themselves or obtained from available vocabulary schedules [31–37]. Two main problems appear in producing complex text identifiers:

1. When stringent conditions are used for the construction of complex identifiers, typified by the use of restrictive frequency criteria and limited cooccurrence contexts for the recognition of term phrases, then few new identifiers are likely to become available, and the performance of the retrieval system with complex identifiers will differ only marginally from the results obtainable with single term indexing.
2. On the other hand, when the construction criteria for the complex entities are relaxed, then some good identifiers are obtained, but also many marginal ones that do not prove useful. Overall, the single-term indexing will generally be preferred.

When single terms are used for content identification, distinctions must be introduced between individual terms, based on their presumed value as document descriptors. This leads to the use of term weights attached to the item identifiers. The considerations controlling the generation of effective weighting factors are outlined briefly in the next section.

2. TERM-WEIGHT SPECIFICATION

The main function of a term-weighting system is the enhancement of retrieval effectiveness. Effective retrieval depends on two main factors: one, items likely to be relevant to the user's needs must be retrieved; two, items likely to be extraneous must be rejected. Two measures are normally used to assess the ability of a system to retrieve the relevant and reject the nonrelevant items of a collection, known as *recall* and *precision*, respectively. Recall is the proportion of relevant items retrieved, measured by the ratio of the number of relevant retrieved items to the total number of relevant items in the collection; precision, on the other hand, is the proportion of retrieved items that are relevant, measured by the ratio of the number of relevant retrieved items to the total number of retrieved items.

In principle, a system is preferred that produces both high recall by retrieving everything that is relevant, and also high precision by rejecting all items that are extraneous. The recall function of retrieval appears to be best served by using broad, high-frequency terms that occur in many documents of the collection. Such terms may be expected to pull out many documents, including many of the relevant documents. The precision factor, however, may be best served by using narrow, highly specific terms that are capable of isolating the few relevant items from the mass of nonrelevant ones. In practice, compromises are normally made by using terms that are broad enough to achieve a reasonable recall level without at the same time producing unreasonably low precision.

The differing recall and precision requirements favor the use of composite term weighting factors that contain both recall- and precision-enhancing components. Three main considerations appear important in this connection. First, terms that are frequently mentioned in individual documents, or document excerpts, appear to be useful as recall-enhancing devices. This suggests that a *term frequency* (tf) factor be used as part of the term-weighting system measuring the frequency of occurrence of the terms in the document or query texts. Term-frequency weights have been used for many years in automatic indexing environments [1–4].

Second, term frequency factors alone cannot ensure acceptable retrieval performance. Specifically, when the high frequency terms are not concentrated in a few particular documents, but instead are prevalent in the whole collection, all documents tend to be retrieved, and this affects the search precision. Hence a new collection-dependent factor must be introduced that favors terms concentrated in a few documents of a collection. The well-known *inverse document frequency* (idf) (or inverse collection frequency) factor performs this function. The idf factor varies inversely with the number of documents n to which a term is assigned in a collection of N documents. A typical idf factor may be computed as $\log N/n$ [38].

Term discrimination considerations suggest that the best terms for document content identification are those able to distinguish certain individual documents from the remainder of the collection. This implies that the best terms should have high term frequencies but low overall collection frequencies. A reasonable measure of term importance may then be obtained by using the product of the term frequency and the inverse document frequency (tf $\times$ idf) [39–41].

The term discrimination model has been criticized because it does not exhibit well substantiated theoretical properties. This is in contrast with the probabilistic model of information retrieval where the relevance properties of the documents are taken into account, and a theoretically valid *term relevance* weight is derived [42–44]. The term relevance

weight—defined as the proportion of relevant documents in which a term occurs divided by the proportion of nonrelevant items in which the term occurs—is, however, not immediately computable without knowledge of the occurrence properties of the terms in the relevant and nonrelevant parts of the document collection. A number of methods have been proposed for estimating the term relevance factor in the absence of complete relevance information, and these have shown that under well-defined conditions the term relevance can be reduced to an inverse document frequency factor of the form $\log((N-n)/n)$ [45–46]. The composite (tf × idf) term-weighting system is thus directly relatable to other theoretically attractive retrieval models.

A third term-weighting factor, in addition to the term frequency and the inverse document frequency, appears useful in systems with widely varying vector lengths. In many situations, short documents tend to be represented by short-term vectors, whereas much larger-term sets are assigned to the longer documents. When a large number of terms are used for document representation, the chance of term matches between queries and documents is high, and hence the larger documents have a better chance of being retrieved than the short ones. Normally, all relevant documents should be treated as equally important for retrieval purposes. This suggests that a *normalization factor* be incorporated into the term-weighting formula to equalize the length of the document vectors. Assuming that w represents the weight of term t, the final term weight might then be defined as $w \Big/ \sum_{\text{vector } i} w_i$, or $w \Big/ \sqrt{\sum_{\text{vector } i} (w_i)^2}$.

In the preceding discussion of term-weighting systems, both documents and queries were assumed to be represented by sets, or vectors, of weighted terms. Term-weighting systems have also been applied to Boolean query statements, and extended Boolean systems have been devised in which Boolean query statements are effectively reduced to vector form [47–54]. The previous considerations regarding term weighting thus apply to some extent also to Boolean query processing.

3. TERM-WEIGHTING EXPERIMENTS

A number of term-weighting experiments are described in the remainder of this note in which combinations of term frequency, collection frequency, and length normalization components are used with six document collections of varying size, covering different subject areas. In each case, collections of user queries are used for retrieval purposes and the performance is averaged over the number of available user queries. For each experiment, the average search precision is computed for three different recall points, including a low recall of 0.25, an average recall of 0.50, and a high recall of 0.75. This average search precision is then further averaged for all available user queries. In addition, to the precision measure, the rank of the weighting methods in decreasing performance order is used as an evaluation criterion. A total of 1800 different combinations of term-weight assignments were used experimentally, of which 287 were found to be distinct. A rank of 1 thus designates the best performance, and 287 the worst.

In the present experiments, each term-weight combination is described by using two triples, representing, respectively, the term frequency, collection frequency, and vector normalization factors for document terms (first triple), and query terms (second triple). The principal weighting components are defined in Table 1. Three different term-frequency components are used, including a binary weight (b), the normal term frequency (t), and a normalized term frequency (n) that lies between 0.5 and 1.0. The three collection frequency components represent multipliers of 1 (x) that disregards the collection frequency, a conventional inverse collection frequency factor (f), and a probabilistic inverse collection frequency (p). Finally, the length normalization factor may be absent (x as the third component) or present (c). (In the previously mentioned full set of 1800 different term-weight assignments, additional weighting components not included in Table 1 were also tried. These additional components did not supply any fundamentally new insights or advantages.)

Table 2 shows actual formulas for some well-known term-weighting systems. The

Table 1. Term-weighting components

Term Frequency Component		
b	1.0	binary weight equal to 1 for terms present in a vector (term frequency is ignored)
t	tf	raw term frequency (number of times a term occurs in a document or query text)
n	$0.5 + 0.5 \frac{\text{tf}}{\max \text{tf}}$	augmented normalized term frequency (tf factor normalized by maximum tf in the vector, and further normalized to lie between 0.5 and 1.0)
Collection Frequency Component		
x	1.0	no change in weight; use original term frequency component (b, t, or n)
f	$\log \frac{N}{n}$	multiply original tf factor by an inverse collection frequency factor (N is total number of documents in collection, and n is number of documents to which a term is assigned)
p	$\log \frac{N-n}{n}$	multiply tf factor by a probabilistic inverse collection frequency factor
Normalization Component		
x	1.0	no change; use factors derived from term frequency and collection frequency only (no normalization)
c	$1 \Big/ \sqrt{\sum_{\text{vector}} w_i^2}$	use cosine normalization where each term weight w is divided by a factor representing Euclidian vector length

Table 2. Typical term-weighting formulas

Weighting System	Document term weight	Query Term weight
Best fully weighted system *tfc·nfx*	$\frac{\text{tf} \cdot \log \frac{N}{n}}{\sqrt{\sum_{\text{vector}} \left(\text{tf}_i \cdot \log \frac{N}{n_i}\right)^2}}$	$\left(0.5 + \frac{0.5\ \text{tf}}{\max \text{tf}}\right) \cdot \log \frac{N}{n}$
Best weighted probabilistic weight *nxx·bpx*	$0.5 + \frac{0.5\ \text{tf}}{\max \text{tf}}$	$\log \frac{N-n}{n}$
Classical idf weight *bfx·bfx*	$\log \frac{N}{n}$	$\log \frac{N}{n}$
Binary term independence *bxx·bpx*	1	$\log \frac{N-n}{n}$
Standard tf weight: *txc·txx*	$\frac{\text{tf}}{\sqrt{\sum_{\text{vector}} (\text{tf}_i)^2}}$	tf
Coordination level *bxx·bxx*	1	1

coordination-level match, which simply reflects the number of matching terms present in documents and queries, respectively, is described by the sextuple *bxx·bxx*. Similarly, the probabilistic binary term independence system that uses binary document terms, but a probabilistic inverse collection frequency weight for the query terms, is represented as *bxx·bpx*. A typical complex term-weighting scheme, described as *tfc·nfx*, uses a normalized tf × idf weight for document terms, and an enhanced, but unnormalized tf × idf factor for the queries. (Since the query vectors remain constant for all documents of a collection, a query normalization simply adds a constant factor to all query-document similarity measurements, which leaves the final document ranking unaffected.)

The six collections used experimentally are characterized by the statistics of Table 3. The smallest collection is a biomedical (MED) collection, consisting of 1033 documents and

Table 3. Collection statistics (including average vector length and standard deviation of vector lengths)

Collection	Number of vectors (documents or queries)	Average vector length (number of terms)	Standard deviation of vector length	Average frequency of terms in vectors	Percentage of terms in vectors with frequency 1
CACM					
documents	3,204	24.52	21.21	1.35	80.93
queries	64	10.80	6.43	1.15	88.68
CISI					
documents	1,460	46.55	19.38	1.37	80.27
queries	112	28.29	19.49	1.38	78.36
CRAN					
documents	1,398	53.13	22.53	1.58	69.50
queries	225	9.17	3.19	1.04	95.69
INSPEC					
documents	12,684	32.50	14.27	1.78	61.06
queries	84	15.63	8.66	1.24	83.78
MED					
documents	1,033	51.60	22.78	1.54	72.70
queries	30	10.10	6.03	1.12	90.76
NPL					
documents	11,429	19.96	10.84	1.21	84.03
queries	100	7.16	2.36	1.00	100.00

30 queries, whereas the largest collection (INSPEC) comprises 12684 documents and 84 queries, covering the computer engineering areas. In all cases, the query vectors are much shorter than the corresponding document vectors.

The NPL (National Physical Laboratory) collection of 11429 documents and 100 queries was available in indexed form only (i.e., in the form of document and query vectors) and not in original natural language form. This may explain its somewhat peculiar makeup. Both the document and the query vector are much shorter in the NPL collection than in the other collections, and the variation in query length (2.36 for a mean number of 7.16 query terms) is very small. Furthermore, the term frequencies are especially low for the NPL collection: each query term appears precisely once in a query, and the average frequency of the terms in the documents is only 1.21. In these circumstances, the term frequency weighting and length normalization operations cannot perform their intended function. One may conjecture that the NPL index terms are carefully chosen, and may in fact represent specially controlled terms rather than freely chosen natural language entries.

Typical evaluation output is shown in Tables 4 and 5. With a few minor exceptions, the results for the five collections of Table 4 are homogeneous, in the sense that the best results are produced by the same term-weighting systems for all collections, and the same holds also for the poorest results. The results of Table 4 do however differ substantially from those obtained for the NPL collection in Table 5. Considering first the results of Table 4, the following conclusions are evident:

1. Methods 1 and 2 produce comparable performances for all collections, the length normalization is important for the documents, and the enhanced query weighting is effective for the queries. These methods are recommended for conventional natural language texts and text abstracts.
2. Method 3 does not include the normalization operation for vector length, nor the enhanced query weights. This unnormalized (tf × idf) weighting method is poor for collections such as CRAN and MED where very short query vectors are used with little deviation in the query length. In such cases, enhanced query weights (*n* factor) prove important.
3. Method 4 represents the best of the probabilistic weighting systems. This method is less effective than the enhanced weighting schemes of methods 1 and 2. It fails especially for collections such as CISI and INSPEC where long query vectors are used and the term discrimination afforded by query term weighting is essential.
4. Methods 5 to 7 represent, respectively, the classical inverse document frequency weighting, the probabilistic binary term independence system, and the classical term frequency weighting. As can be seen, these methods are generally inferior for all collections.
5. The coordination level matching of binary vectors represents one of the worst possible retrieval strategies.

Table 4. Performance results for eight term-weighting methods averaged over 5 collections

Term-weighting methods	Rank of method and ave. precision	CACM 3204 docs 64 queries	CISI 1460 docs 112 queries	CRAN 1397 docs 225 queries	INSPEC 12,684 docs 84 queries	MED 1033 docs 30 queries	Averages for 5 collections
1. Best fully weighted (*tfc·nfx*)	Rank	1	14	19	3	19	11.2
	P	0.3630	0.2189	0.3841	0.2626	0.5628	
2. Weighted with inverse frequency *f* not used for docs (*txc·nfx*)	Rank	25	14	7	4	32	16.4
	P	0.3252	0.2189	0.3950	0.2626	0.5542	
3. Classical tf × idf No normalization (*tfx·tfx*)	Rank	29	22	219	45	132	84.4
	P	0.3248	0.2166	0.2991	0.2365	0.5177	
4. Best weighted probabilistic (*nxx·bpx*)	Rank	55	208	11	97	60	86.2
	P	0.3090	0.1441	0.3899	0.2093	0.5449	
5. Classical idf without normalization (*bfx·bfx*)	Rank	143	247	183	160	178	182
	P	0.2535	0.1410	0.3184	0.1781	0.5062	
6. Binary independence probabilistic (*bxx·bpx*)	Rank	166	262	154	195	147	159
	P	0.2376	0.1233	0.3266	0.1563	0.5116	
7. Standard weights cosine normalization (original Smart) (*txc·txx*)	Rank	178	173	137	187	246	184
	P	0.2102	0.1539	0.3408	0.1620	0.4641	
8. Coordination level binary vectors (*bxx·bxx*)	Rank	196	284	280	258	281	260
	P	0.1848	0.1033	0.2414	0.0944	0.4132	

Table 5. Performance results for NPL collection (11429 docs, 100 queries)

Evaluation	Best fully weighted *tfc·nfx*	Weighted restricted *f* *txc·nfx*	Classical tf × idf *tfx·tfx*	Best probabilistic *nxx·bpx*	Classical idf system *bfx·bfx*	Binary independence *bxx·bpx*	Standard weight *txc·txx*	Coordination level *bxx·bxx*
Rank	116	62	149	2	23	8	172	83
Average precision	0.1933	0.2170	0.1846	0.2752	0.2406	0.2596	0.1750	

The results of Table 5 for the NPL collection differs markedly from those of Table 4. Here the probabilistic schemes using binary query weights and unnormalized document vectors are preferred. This is a direct result of the special nature of the queries and documents for that collection: the very short queries with little length deviation require fully weighted query terms ($b = 1$), and the normally effective term frequency weights should be avoided because many important terms will then be downgraded in the short document vectors. An enhanced term frequency weight (*n* factor), or a full weight ($b = 1$) is therefore preferred. Retrieval results obtained for NPL were used earlier to claim superiority for the probabilistic term weighting system [55]. The results of Tables 4 and 5 do not support this contention for conventional natural language documents and queries.

4. RECOMMENDATIONS

The following conclusions may be drawn from the experimental evidence reported in this study:

4.1 *Query vectors*

1. Term-frequency component
   - For short query vectors, each term is important; enhanced query term weights are thus preferred: first component *n*.
   - Long query vectors require a greater discrimination among query terms based on term occurrence frequencies: first component *t*.
   - The term-frequency factor can be disregarded when all query terms have occurrence frequencies equal to 1.
2. Collection-frequency component
   - Inverse collection frequency factor *f* is very similar to the probabilistic term independence factor *p*: best methods use *f*.
3. Normalization component
   - Query normalization does not affect query-document ranking or overall performance; use *x*.

4.2 *Document vectors*

1. Term-frequency component
   - For technical vocabulary and meaningful terms (CRAN, MED collections), use enhanced frequency weights: first component *n*.
   - For more varied vocabulary, distinguish terms by conventional frequency weights: first component *t*.
   - For short document vectors possibly based on controlled vocabulary, use fully weighted terms: first component $b = 1$.
2. Collection-frequency component
   - Inverse document-frequency factor *f* is similar to probabilistic term independence weight *p*: normally use *f*.
   - For dynamic collections with many changes in the document collection makeup, the *f* factor requires updating; in that case disregard second component: use *x*.
3. Length-normalization component
   - When the deviation in vector lengths is large, as it normally is in text indexing systems, use length normalization factor *c*.
   - For short document vectors of homogeneous length, the normalization factor may be disregarded; in that case use *x*.

The following single-term weighting systems should be used as a standard for comparison with enhanced text analysis systems using thesauruses and other knowledge tools to produce complex multiterm content identifications:

Best document weighting *tfc, nfc* (or *tpc, npc*)

Best query weighting *nfx, tfx, bfx* (or *npx, tpx, bpx*)

REFERENCES

1. Luhn, H.P. A statistical approach to the mechanized encoding and searching of literary information. IBM Journal of Research and Development 1:4; 309–317; October 1957.
2. Salton, G., ed. The Smart Retrieval System—Experiments in Automatic Document Retrieval. Englewood Cliffs, NJ: Prentice Hall Inc.; 1971.
3. Salton, G., McGill, M.J. Introduction to Modern Information Retrieval. New York: McGraw-Hill Book Co.; 1983.
4. van Rijsbergen, C.J. Information Retrieval, 2nd ed. London: Butterworths; 1979.
5. Luhn, H.P. A new method of recording and searching information. American Documentation 4:1; 14–16; 1955.
6. Taube, M.; Wachtel, I.S. The logical structure of coordinate indexing. American Documentation 3:4; 213–218; 1952.
7. Perry, J.W. Information analysis for machine searching. American Documentation 1:3; 133–139; 1950.
8. van Rijsbergen, C.J. A theoretical basis for the use of cooccurrence data in information retrieval. Journal of Documentation 33:2; 106–119; June 1977.
9. Salton, G.; Buckley, C.; Yu, C.T. An evaluation of term dependence models in information retrieval. Lecture Notes in Computer Science, In: Salton, G.; Schneider, H.J., eds. 146, Berlin: Springer-Verlag, 151–173; 1983.
10. Yu, C.T.; Buckley, C.; Lam, K.; Salton, G. A generalized term dependence model in information retrieval. Information Technology: Research and Development 2:4; 129–154; October 1983.
11. Lesk, M.E. Word-word associations in document retrieval systems. American Documentation 20:1; 27–38; January 1969.
12. Klingbiel, P.H. Machine aided indexing of technical literature. Information Storage and Retrieval 9:2; 79–84; February 1973.
13. Klingbiel, P.H. A technique for machine aided indexing. Information Storage and Retrieval 9:9; 477–494; September 1973.
14. Dillon, M.; Gray, A. Fully automatic syntax-based indexing. Journal of the ASIS 34:2; 99–108; March 1983.
15. Sparck Jones, K.; Tait, J.I. Automatic search term variant generation. Journal of Documentation 40:1; 50–66; March 1984.
16. Fagan, J.L. Experiments in automatic phrase indexing for document retrieval: A comparison of syntactic and non-syntactic methods. Doctoral thesis, Report 87-868, Department of Computer Science, Cornell University, Ithaca, NY; September 1987.
17. Smeaton, A.F. Incorporating syntactic information into a document retrieval strategy: An investigation. Proc. 1986 ACM-SIGIR Conference on Research and Development in Information Retrieval, Pisa, Italy, Association for Computing Machinery, New York; 103–113; 1986.
18. Sparck Jones, K. Automatic Keyword Classification for Information Retrieval. London: Butterworths; 1971.
19. Salton, G. Experiments in automatic thesaurus construction for information retrieval. Information Processing 71, North Holland Publishing Co., Amsterdam; 115–123; 1972.
20. Dattola, R.T. Experiments with fast algorithms for automatic classification. In: Salton, G., ed. The Smart Retrieval System-Experiments in Automatic Document Processing. Chapter 12, pp. 265–297. Englewood Cliffs, NJ: Prentice Hall Inc.; 1971.
21. Walker, D.E. Knowledge resource tools for analyzing large text files. In: Nirenburg, Sergei, ed. Machine Translation: Theoretical and Methodological Issues, pp. 247–261. Cambridge, England: Cambridge University Press; 1987.
22. Kucera, H. Uses of on-line lexicons. Proc. First Conference of the U.W. Centre for the New Oxford English Dictionary: Information in Data, pp. 7–10. University of Waterloo; 1985.
23. Amsler, R.A. Machine-readable dictionaries. In: Williams, M.E., ed. Annual Review of Information Science and Technology, Vol. 19, pp. 161–209. White Plains, NY: Knowledge Industry Publication Inc.; 1984.
24. Fox, E.A. Lexical relations: Enhancing effectiveness of information retrieval systems. ACM SIGIR Forum, 15:3; 5–36; 1980.
25. Croft, W.B. User-specified domain knowledge for document retrieval. Proc. 1986 ACM Conference on Research and Development in Information Retrieval, pp. 201–206; Pisa, Italy. New York: Association for Computing Machinery; 1986.
26. Thompson, R.H.; Croft, W.B. An expert system for document retrieval. Proc. Expert Systems in Government Symposium, pp. 448–456. Washington, DC: IEEE Computer Society Press; 1985.
27. Croft, W.B. Approaches to intelligent information retrieval. Information Processing & Management 23:4; 249–254; 1987.
28. Sparck Jones, K. Intelligent retrieval. In: Intelligent Information Retrieval: Proc. Informatics, Vol. 7; 136–142; Aslib; London: 1983.
29. Fox, E.A. Development of the coder system: A testbed for artificial intelligence methods in information retrieval. Information Processing & Management 23:4; 341–366; 1987.
30. Salton, G. On the use of knowledge based processing in automatic text retrieval. Proc. 49th Annual Meeting of the ASIS, Learned Information, Medford, NJ: 277–287; 1986.
31. Swanson, D.R. Searching natural language text by computer. Science 132:3434; 1099–1104; October 1960.
32. Cleverdon, C.W.; Keen, E.M. Aslib-Cranfield research project, Vol. 2, Test Results. Cranfield Institute of Technology, Cranfield, England; 1966.
33. Cleverdon, C.W. A computer evaluation of searching by controlled languages and natural language in an experimental NASA database, Report ESA 1/432. European Space Agency, Frascati, Italy; July 1977.
34. Lancaster, F.W. Evaluation of the Medlars demand search service, National Library of Medicine, Bethesda, MD; January 1968.
35. Blair, D.C.; Maron, M.E. An evaluation of retrieval effectiveness for a full-text document retrieval system. Communications of the ACM 28:3; 289–299; March 1985.
36. Salton, G. Another look at automatic text retrieval systems. Communications of the ACM, 29:7; 648–656; July 1986.
37. Salton, G. Recent studies in automatic text analysis and document retrieval. Journal of the ACM 20:2; 258–278; April 1973.
38. Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28:1; 11–21; March 1972.
39. Salton, G.; Yang, C.S. On the specification of term values in automatic indexing. Journal of Documentation 29:4; 351–372; December 1973.
40. Salton, G. A theory of indexing. Regional Conference Series in Applied Mathematics, No. 18, Society for Industrial and Applied Mathematics, Philadelphia, PA; 1975.

41. Salton, G.; Yang, C.S.; Yu, C.T. A theory of term importance in automatic text analysis. Journal of the ASIS 26:1; 33–44; January–February 1975.
42. Bookstein, A.; Swanson, D.R. A decision theoretic foundation for indexing. Journal of the ASIS 26:1; 45–50; January–February 1975.
43. Cooper, W.S.; Maron, M.E. Foundation of probabilistic and utility theoretic indexing. Journal of the ACM 67–80; 25:1; 1978.
44. Robertson, S.E.; Sparck Jones, K. Relevance weighting of search terms. Journal of the ASIS 27:3; 129–146; 1976.
45. Croft, W.B.; Harper, D.J. Using probabilistic models of information retrieval without relevance information. Journal of Documentation 35:4; 285–295; December 1975.
46. Wu, H.; Salton, G. A comparison of search term weighting: Term relevance versus inverse document frequency. ACM SIGIR Forum 16:1; 30–39; Summer 1981.
47. Noreault, T.; Koll, M.; McGill, M.J. Automatic ranked output from Boolean searches in SIRE. Journal of the ASIS 27:6; 333–339; November 1977.
48. Radecki, T. Incorporation of relevance feedback into Boolean retrieval systems. Lecture Notes in Computer Science, 146, G. Salton and H.J. Schneider, eds., pp. 133–150. Berlin: Springer-Verlag; 1982.
49. Paice, C.D. Soft evaluation of Boolean search queries in information retrieval systems. Information Technology: Research and Development 3:1; 33–41; 1983.
50. Cater, S.C.; Kraft, D.H. A topological information retrieval system (TIRS) satisfying the requirements of the Waller-Kraft wish list. Proc. Tenth Annual ACM-SIGIR Conference on Research and Development in Information Retrieval, C.T. Yu and C.J. van Rijsbergen, eds. pp. 171–180. Association for Computing Machinery, New York: 1987.
51. Wong, S.K.M.; Ziarko, W.; Raghavan, V.V.; Wong, P.C.N. Extended Boolean query processing in the generalized vector space model, Report, Department of Computer Science, University of Regina, Regina, Canada; 1986.
52. Wong, S.K.M.; Ziarko, W.; Wong, P.C.N. Generalized vector space model in information retrieval. Proc. Eighth Annual ACM-SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, New York; 18–25; 1985.
53. Wu, H. On query formulation in information retrieval. Doctoral dissertation, Cornell University, Ithaca, NY; January 1981.
54. Salton, G.; Fox, E.A.; Wu, H. Extended Boolean information retrieval. Communications of the ACM 26:11; 1022–1036; November 1983.
55. Croft, W.B. A comparison of the cosine correlation and the modified probabilistic model. Information Technology: Research and Development 3:2; 113–114; April 1984.