

# The Measurement of Term Importance in Automatic Indexing\*

G. Salton and H. Wu

*Department of Computer Science, Cornell University, Ithaca, NY 14853*

C. T. Yu

*Department of Information Engineering, University of Illinois at Chicago Circle, Chicago, IL 60680*

The frequency characteristics of terms in the documents of a collection have been used as indicators of term importance for content analysis and indexing purposes. In particular, very rare or very frequent terms are normally believed to be less effective than medium-frequency terms. Recently automatic indexing theories have been devised that use not only the term frequency characteristics but also the relevance properties of the terms. The major term-weighting theories are first briefly reviewed. The term precision and term utility weights that are based on the occurrence characteristics of the terms in the relevant, as opposed to the nonrelevant, documents of a collection are then introduced. Methods are suggested for estimating the relevance properties of the terms based on their overall occurrence characteristics in the collection. Finally, experimental evaluation results are shown comparing the weighting systems using the term relevance properties with the more conventional frequency-based methodologies.

## 1. Introduction

In information retrieval the stored information items and the incoming search requests are normally represented by sets of content identifiers variously known as keywords, index terms, or simply terms. In most operational systems a binary mode is used to assign terms to the items of a collection: in particular, a term may be assigned to an item if it appears at least marginally useful for purposes of content representation; the term is rejected when it is clearly extraneous. Furthermore, the retrieval of an item is normally based on exact match strategy between documents and queries; that is, a document is retrieved when it exhibits the exact term combination specified in the corresponding search requests. In a binary indexing mode, all items that are approximately similar in content will be identified by

the same set of terms, and all those items will then be jointly retrieved in response to a particular query.

Binary indexing and exact match retrieval simplify the search and retrieval operations. The user's task in evaluating the search output is, however, considerably complicated because no distinctions are made within the set of retrieved items, or for that matter within the set of items that are not retrieved. More often than not, the user is unable to distinguish items exhibiting different degrees of relevance, and the items of greatest potential use are not brought to the user's attention ahead of other more marginal items.

A greater degree of discrimination can be achieved among the documents by attaching *importance indicators*, or *weights*, to the terms assigned to documents and search requests. Thus, given two documents  $D_1$  and  $D_2$ , both dealing with plums and pears, an assignment of (PLUM, 5; PEAR, 2) to  $D_1$  and (PLUM, 2; PEAR, 5) to  $D_2$  can distinguish  $D_1$  dealing principally with plums from  $D_2$  concerned mostly with pears. When weighted, instead of binary, terms are assigned to the items of a collection, the degree of agreement between a document and query can be ascertained by computing a similarity coefficient for each query-document pair. Such a computation can be used as a basis for ranking the documents retrieved in response to a given query in decreasing order of the query-document similarity. Items exhibiting the more highly weighted query terms can then be retrieved ahead of other items with query terms of lower weight.

Consider, for example, a user interested in items dealing more with plums rather than with pears. Such a user might propose a query  $Q$  formulated as (PLUM, 2; PEAR, 1). If the similarity between queries and documents is computed as the sum of the products of the weights for corresponding query and document terms (the so-called inner product), one obtains for the previously used sample documents a similarity of 12 ( $[2 \times 5] + [1 \times 2]$ ) between  $Q$  and  $D_1$ , and a similarity of 9 ( $[2 \times 2] + [1 \times 5]$ ) between  $Q$  and  $D_2$ . In such circumstances, a *ranked* retrieval system will retrieve document  $D_1$  (which like the query assigns a large weight to PLUM) ahead of document  $D_2$  that stresses PEAR instead of PLUM.

\*This study was supported in part by the National Science Foundation under Grant IST 79-05301.

Received June 4, 1980; accepted August 8, 1980

© 1981 by John Wiley & Sons, Inc.

The greater degree of discrimination between documents achievable by assigning weighted terms to documents and queries may be expected to improve retrieval effectiveness and enhance user satisfaction. By using the document ranking possibility and introducing appropriate retrieval thresholds to distinguish the retrieved from the nonretrieved items, it may then be possible to control the number of items retrieved in accordance with the wishes of individual users. Furthermore, the user effort required during the search process may be minimized because the most important items—those that appear to be most similar to the query statements—are likely to be retrieved early in a search ahead of more marginal items. This simplifies the task of constructing improved query formulations to be used in subsequent search efforts.

Several modern term-weighting systems are examined in the remainder of this study and their importance is assessed in the theory and practice of indexing.

## 2. Similarity, Ranking, and Thresholding Operations

In conventional retrieval situations based on exact query-document comparisons, documents are retrieved whenever all the query terms are also used as terms in the document representations. When weighted terms are assigned to documents and queries, a more elaborate system of query-document matching must be introduced. In particular, it becomes necessary to take into account not only the number of terms that jointly are assigned to a document and query, or that are jointly absent from both, but also the weight of the respective terms.

Consider as an example two particular objects  $D_i$  and  $Q_j$  representing the  $i$ th document and the  $j$ th query, respectively. If  $d_{ik}$  and  $q_{jk}$  represent the weights of the  $k$ th terms assigned to  $D_i$  and  $Q_j$ , the term assignment to the sample document and query may be represented in *vector* form as

$$D_i = (d_{i1}, d_{i2}, \dots, d_{it}), \quad (1)$$

$$Q_j = (q_{j1}, q_{j2}, \dots, q_{jt}),$$

where  $t$  terms in all are included in the indexing system and a weight of zero is assumed for terms absent from the respective vectors.

When a query is formulated by sets of index terms without interconnecting Boolean operators, any standard vector-matching function can be used to reflect the similarity between query and document vectors. Typically the inner product between the vectors serves adequately for the computation of similarity coefficients between documents and queries:

$$SIM(D_i, Q_j) = \sum_{k=1}^t d_{ik} q_{jk} \quad (2)$$

When  $D_i$  and  $Q_j$  do not have any term in common the computation of Eq. (2) produces zero as a result. The upper bound of Eq. (2) is, however, not well specified. To obtain a similarity measure ranging from zero for no match to one

for maximum agreement between the two vectors, it is customary to add a normalizing factor. Two typical vector similarity measures ranging from zero to one are the Jaccard and cosine measures of expressions (3) and (4), respectively [1,2]:

$$SIM_1(D_i, Q_j) = \frac{\sum_{k=1}^t d_{ik} q_{jk}}{\sum_{k=1}^t (d_{ik})^2 + \sum_{k=1}^t (q_{jk})^2 - \sum_{k=1}^t d_{ik} q_{jk}} \quad (3)$$

and

$$SIM_2(D_i, Q_j) = \frac{\sum_{k=1}^t d_{ik} q_{jk}}{\left[ \sum_{k=1}^t (d_{ik})^2 \cdot \sum_{k=1}^t (q_{jk})^2 \right]^{1/2}} \quad (4)$$

For the sample vectors previously introduced, term 1 corresponds to PLUM and term 2 to PEAR. In vector form the query and documents are then represented as  $Q = (2,1)$ ,  $D_1 = (5,2)$ , and  $D_2 = (2,5)$ , and three similarity coefficients  $SIM$ ,  $SIM_1$ , and  $SIM_2$  produce the following results when applied to  $(Q, D_1)$  and  $(Q, D_2)$ , respectively: 12 and 9; 12/22 and 9/25; and  $12/\sqrt{145}$  and  $9/\sqrt{145}$ . In each case  $D_1$  receives a higher similarity rating with  $Q$  than  $D_2$ .

A great many similarity measures are discussed in the literature. They all exhibit one common property, namely that their value increases when the number of common terms or the weight of the common terms in two vectors increases. Measures of vector dissimilarity that are sometimes used instead of similarity measures have the opposite effect. Both the Jaccard and the cosine measures [Eqs. (3) and (4)] have been widely used in practice and they appear to be as effective in retrieval as other more complicated functions.

When the Boolean connectives AND, OR, and NOT are used to relate the query terms, the computation of item similarities becomes more complex. Consider as an example two arbitrary index terms designated as  $A$  and  $B$ , and let  $A$  and  $B$  represent the set of documents indexed by terms  $A$  and  $B$ , respectively. When unweighted query terms are used the standard Boolean queries receive the following interpretation:

- query " $A$  OR  $B$ " retrieves document set  $(A \cup B)$  consisting of documents indexed by term  $A$ , or by term  $B$ , or by both  $A$  and  $B$ ;
- query " $A$  AND  $B$ " retrieves document set  $(A \cap B)$  consisting of documents indexed by both terms  $A$  and  $B$ ;
- query " $A$  NOT  $B$ " retrieves document sets  $(A - B)$  consisting of documents indexed by term  $A$  that are not also indexed by  $B$ .

When weighted terms are included in Boolean query formulations a problem of interpretation arises [3,4]. It appears reasonable to use the weights attached to *document terms* as a *ranking* device as illustrated by the earlier examples. In that case documents identified by highly weighted query terms are retrieved ahead of items carrying query terms of lesser weight. The weights attached to the *query terms*, on the other hand, may be used for *thresholding* purposes. In particular, assuming that the term weights vary between zero and one, the normal Boolean operations may be carried out for query terms with weight equal to one. A query weight of zero correspondingly indicates that the operand may be disregarded. Query weights between zero and one produce intermediate results in the sense that a query term such as  $A_a$  for  $0 < a < 1$  will affect a subset of the documents indexed by  $A$ , instead of all of  $A$ .

When both the document and the query terms are weighted the thresholding and ranking operations are carried out simultaneously. Thus, in response to a query such as  $(A_a \text{ OR } B_b)$ , the set of documents retrieved consists of those having either term  $A$  with a weight at least equal to  $a$  or term  $B$  with a weight at least equal to  $b$ . These retrieved items are then ranked according to the sum of the weights  $a + b$  in the document vectors. When only the documents are weighted but not the queries, the full document sets  $A$  and/or  $B$  are retrieved using the appropriate Boolean combination, and the ranking applies as before [5]. When the query terms alone are weighted but document terms are not, the thresholding operation applies but not the ranking. A query such as  $(A_a \text{ OR } B_b)$  then affects some fraction of set  $A$  and some fraction of  $B$ . A possible interpretation for that case is described in the Appendix.

### 3. Term-Weighting Systems

#### A. Historical Developments

It has long been recognized that when users or search intermediaries are charged with the manual assignment of term weights to documents and query vectors the weighting operation is difficult to control. The main problem is that a

correct assignment of weights requires a great deal of know-how about the collection makeup and about the operations of the retrieval system. For this reason it has been conjectured that an effective term-weighting system is best carried out by using objective term characteristics to generate the term weights automatically.

Luhn [6] was the first to suggest that the frequency of occurrence of the terms in a collection had something to do with the usefulness of the terms for indexing purposes. In particular, he conjectured that very-high-frequency terms—those that occur in many documents of a collection—would be too broad and would lead to losses in search *precision*. On the other hand, very-low-frequency terms—those assigned to only very few documents—would be too narrow and thus lead to *recall* losses.\* The best terms, it was suggested, would be medium-frequency terms that are assigned to neither too many nor too few items. Luhn drew a graph exhibiting the “resolving power” of a term as a function of the document frequency, that is, the number of documents in a collection, to which a term is assigned. Such a graph is shown in Figure 1. Two threshold frequencies, labeled  $A$  and  $B$  in the figure, are used to eliminate the rare as well as the frequent terms, and the remaining medium-frequency terms with the highest resolving power are then used for indexing purposes [6].

The basic automatic indexing proposals by Luhn left something to be desired, first because the required threshold frequencies ( $A$  and  $B$ ) are difficult to determine, and second because the simple deletion of high- and low-frequency terms is likely to lead to recall and precision losses, respectively. The shape of the resolving power curve was also left unspecified by Luhn; hence the resolving power could not immediately be used as an indication of term importance, or weight. Nevertheless, Luhn’s automatic indexing proposals proved to be reasonably perceptive, as will be seen in the remainder of this section.

\*Precision is the proportion of the retrieved items that are actually relevant; recall is the proportion of the relevant items actually retrieved. An effective retrieval system is normally expected to produce adequate recall, as well as reasonably high precision.

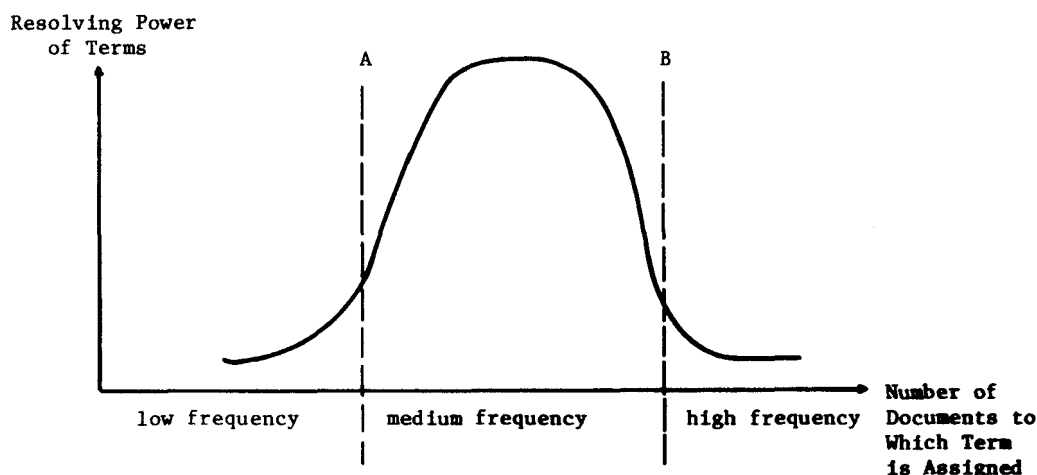


FIGURE 1. Resolving power of terms as a function of frequency.

## B. Frequency-Based Weighting Systems

The early frequency-based term-weighting systems were based on the assumption that documents and search requests would be available in natural language form as abstracts or text excerpts. In that case, the words occurring in the texts could be used for indexing purposes and the occurrence frequencies of these words could lead to an indication of term importance. In particular, for each term  $k$  and document  $i$  (or query  $j$ ), it is then possible to compute the *frequency of occurrence*,  $f_{ik}$ , of term  $k$  in document  $i$  (or in query  $j$ ). From these initial frequency counts it is then possible to compute the total *collection frequency*  $F_k$  of term  $k$  for the  $N$  documents of a collection as

$$F_k = \sum_{i=1}^N f_{ik}.$$

Similarly, the *document frequency*  $B_k$ , defined as the number of documents in a collection to which a term is assigned, is obtained as

$$B_k = \sum_{i=1}^N b_{ik},$$

where  $b_{ik} = 1$  whenever the corresponding  $f_{ik} \geq 1$  and  $b_{ik} = 0$  when  $f_{ik} = 0$ .

The foregoing measurements immediately give rise to several possible term-weighting systems. The *term frequency* (TF) weighting system is based on the notion that language constructs (words, phrases, word groups) that occur in the text of documents or search requests with sufficient frequency have some bearing on the content of the corresponding items. Hence the weight of term  $k$  in document  $i$  (or query  $j$ ),  $w_{ik}$ , might be determined by the frequency of occurrence of word construct  $k$  in document  $i$  (or query  $j$ ); that is

$$w_{ik} = f_{ik}. \quad (5)$$

The formula of Eq. (5) simply reflects the fact that when a document contains a term—say, PLUM or PEAR—many times the document is likely to deal with plums or pears.

Formula (5) unfortunately says nothing about the role of term  $k$  in any document other than document  $i$ . The formula fails in particular when term  $k$  occurs in many documents of a collection. In that case, the term is unable to distinguish one document from another and may not therefore be very useful for indexing purposes. When all documents carry the term PLUM, this term is not helpful for indexing purposes. The *inverse document frequency* (IDF) system postulates that a good term—one that should be assigned to a given query or document with a high weight—exhibits a high-occurrence frequency in a specific document, while overall the collection and document frequencies of the term are low. A high-occurrence frequency in a particular document indicates that the term carries a

good deal of importance in that document; a low overall document frequency indicates at the same time that the term is not equally important in all documents of a collection. Indeed, its importance in the remainder of the collection may be small, so that the term can actually distinguish the document to which it is assigned from the remainder of the collection. This may then render that document actually retrievable when wanted. The inverse document term-weighting function may then be defined as

$$w_{ik} = f_{ik}/B_k, \quad (6)$$

where  $w_{ik}$  again represents the weight of term  $k$  in document  $i$ . There exists substantial evidence that IDF weights offer superior performance in retrieval [7].

## C. Sparse Vector Space Theory

It was noted earlier that a useful indexing system can distinguish the information items from each other. The *term discrimination* theory is a direct approach designed to create document representations that are easily distinguishable [8,9]. Consider, in particular, a collection of documents, or other information items. If the documents are represented in vector form as in Eq. (1), one of the previously mentioned similarity measures can be used to obtain an indication of affinity between pairs of items, say  $D_i$  and  $D_j$ . For example, the cosine coefficient of Eq. (4) measures the similarity between  $D_i$  and  $D_j$  in the form

$$\text{SIM}_2(D_i, D_j) = \frac{\sum_{k=1}^t d_{ik}d_{jk}}{\left[ \sum_{k=1}^t (d_{ik})^2 - \sum_{k=1}^t (d_{jk})^2 \right]^{1/2}}. \quad (7)$$

The space *density*  $\overline{\text{SIM}}$  can be defined as the average similarity between all distinct document pairs in a collection of  $N$  documents as

$$\overline{\text{SIM}} = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{\substack{j=1 \\ i \neq j}}^N \text{SIM}(D_i, D_j). \quad (8)$$

The discrimination value now specifies that a good content term is one which decreases the space density when it is assigned to the items of a collection because in that case the documents are most easily distinguishable from their neighbors. This suggests that the discrimination value of a given term  $k$ ,  $DV_k$ , be computed as the difference between the two space densities  $\overline{\text{SIM}}_k$  and  $\overline{\text{SIM}}$ , respectively, where  $\overline{\text{SIM}}_k$  is the space density for documents from which term  $k$  has been removed:

$$DV_k = \overline{\text{SIM}}_k - \overline{\text{SIM}}. \quad (9)$$

If term  $k$  is a useful term that decreases the space density when it is assigned to the documents, then  $\overline{SIM}_k$ , representing the space density without term  $k$ , will be greater than  $\overline{SIM}$ ; hence the discrimination value is positive in that case. The reverse obtains for poor discriminators that increase the space density when assigned to a collection. An appropriate term-weighting function analogous to the one of Eq. (6) but based on the discrimination value is

$$w_{ik} = f_{ik} \cdot DV_k \quad (10)$$

A general relationship exists between the discrimination value  $DV_k$  of a term and the corresponding document frequency  $B_k$  [8,9]:

(a) Terms with a high document frequency that are assigned to many documents in a collection increase the space density when used; hence  $DV_k < 0$ .

(b) Terms with a low document frequency that are assigned to very few documents in a collection leave the space density unchanged when used; hence  $DV_k = 0$ .

(c) Terms with medium document frequency that are assigned to some items but not to others generally decrease the space density when used; hence  $DV_k > 0$ .

The variation of the discrimination value weights with document frequency of the terms is illustrated in Figure 2. The exact shape of the graph of Figure 2 is not precisely known. However, a comparison of the graphs of Figures 1 and 2 confirms the basic Luhn theory that the most useful terms exhibit medium document frequencies. Low-frequency terms are not as good, and high-frequency terms are even worse since the discrimination value weights then turn negative.

#### D. Term Relevance Theory

No distinctions are made in the computation of the IDF and term discrimination values between occurrences of terms in the relevant and in the nonrelevant documents with respect to certain queries. That is, terms occurring in

a given proportion of nonrelevant items will receive the same weight as terms occurring in an equivalent proportion of the relevant documents. The *term relevance* theory is designed to overcome this shortcoming by introducing distinctions between term occurrences in the relevant as opposed to the nonrelevant documents of a collection. Two particular measurements of term value, known as the term precision and term utility values, make use of this distinction.

The *term precision value* is based on probabilistic considerations by postulating that a given document  $D$  should be retrieved provided its probability of being relevant to a given user query exceeds its probability of being nonrelevant, that is

$$P(D|\text{rel}) > P(D|\text{nonrel}) \quad (11)$$

Assuming that  $D$  is represented by a set of terms  $(d_1, \dots, d_r)$  and that the terms are assigned independently to the items in the collection, the previous inequality can be transformed into

$$P(d_1|\text{rel})P(d_2|\text{rel}) \dots P(d_r|\text{rel}) > P(d_1|\text{nonrel})P(d_2|\text{nonrel}) \dots P(d_r|\text{nonrel}) \quad (12)$$

It remains to determine the individual probabilities of occurrence of each term in the relevant and nonrelevant documents. This problem can be approached by replacing probabilities by frequencies. Consider in particular a sample collection of  $N$  documents, and assume that  $R$  items out of  $N$  are relevant with respect to a certain query, while  $N-R$  are nonrelevant. If the assignment of the terms to the documents is binary in the sense that a given term is either assigned with a weight equivalent to one, or is not assigned (with a weight of zero), the term occurrence characteristic of a given term  $d_k$  in the relevant and nonrelevant documents can be represented as shown in Table 1.

If the data for the sample collection of Table 1 are

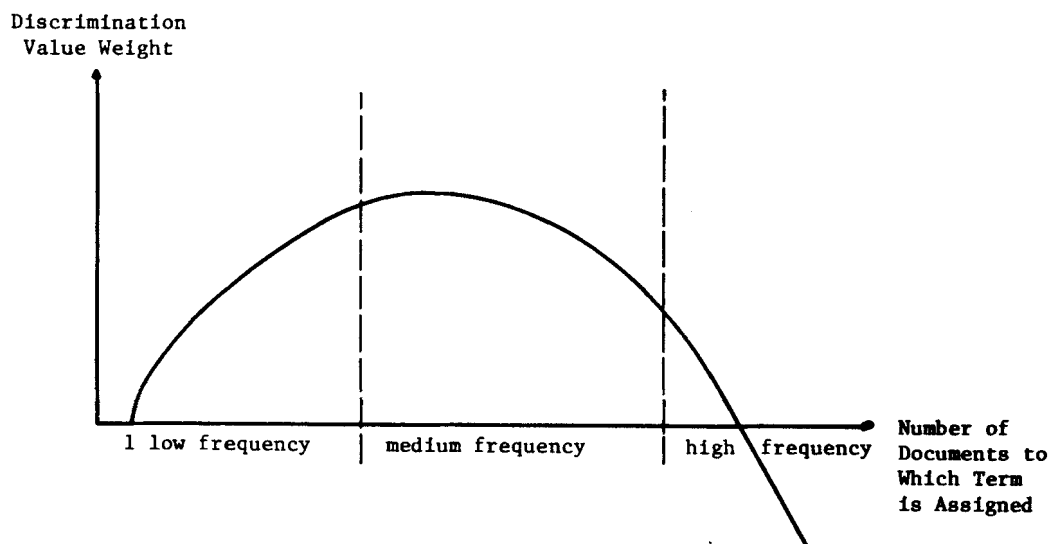


FIGURE 2. Variation of discrimination value weighting with document frequency.

TABLE 1. Occurrence characteristics for term  $k$  in a collection of  $N$  documents of which  $R$  are relevant.

	Number of relevant	Number of nonrelevant	
Term $k$ assigned	$r_k$	$s_k$	$B_k$
Term not assigned	$R - r_k$	$I - s_k$	$N - B_k$
	$R$	$I$	$N$

assumed to be typical of term occurrence frequencies at large, the probabilities of Eq. (12) can be transformed into frequencies as follows:

$$P(d_k | \text{rel}) = r_k/R$$

and

$$P(d_k | \text{nonrel}) = s_k/I. \quad (13)$$

By inserting expressions (12) and (13) into (11) and taking the logarithm one obtains a term importance factor  $L$ , known as the *term precision*, where

$$L_k = \log \frac{r_k/(R - r_k)}{s_k/(I - s_k)}. \quad (14)$$

The term precision represents the proportion of relevant items in which a term occurs divided by the proportion of nonrelevant items in which the term occurs. An appropriate term-weighting function for term  $k$  in document  $D_i$  can then be defined as

$$w_{ik} = f_{ik} \cdot L_k. \quad (15)$$

There is no immediate way of relating the term precision formula to the document frequency of each term as was done earlier for the IDF and for the term discrimination weights. However, assuming that the values of  $r_k$ , the number of relevant items in which a term occurs is known, and that the total number of relevant items  $R$  with respect to some query can be determined, the precision values are easy to compute and may be shown to produce optimal query weight assignments under the stated conditions of term independence and binary term assignment to the documents [10-12]. Methods for estimating values of  $r_k$  and  $R$  are introduced later in this study.

Other avenues exist for obtaining term-weighting functions using the distribution of terms in the relevant and nonrelevant items of a collection. One such method is based on the concept of *search utility* [13,14]. In the utility theoretic approach, a value parameter, say  $v_1$ , is assigned to each relevant item that is properly retrieved, and another value, say  $v_2$ , is attached to each nonrelevant document that is properly rejected. In the same way, cost parameters of  $c_1$  and  $c_2$  are attached to the retrieval of nonrelevant

items and to the nonretrieval of relevant ones, respectively. The total utility of a given search is then defined as the sum of the values obtained by retrieving the relevant and rejecting the nonrelevant items minus the sum of the costs incurred by retrieving the nonrelevant and rejecting the relevant ones.

For the example of Table 2 where a query  $Q$  is processed against a collection of  $N$  documents of which  $R$  are assumed relevant, the following utility value is obtained for retrieval threshold  $t$  (that is assuming that an item is retrieved whenever its similarity with the query exceeds the value  $t$ ):

$$U(R, Q, t) = v_1 r + v_2 (I - s) - c_1 s - c_2 (R - r). \quad (16)$$

If  $\text{SIM}(D, Q)$  is a random variable, denoting a similarity function between document  $D$  and query  $Q$ , the utility expression may be rewritten in terms of probabilities as [14]

$$\begin{aligned} U(R, Q, t) = & v_1 NP \{ D \text{ rel and } \text{SIM}(D, Q) \geq t \} \\ & + v_2 NP \{ D \text{ nonrel and } \text{SIM}(D, Q) < t \} \\ & - c_1 NP \{ D \text{ nonrel and } \text{SIM}(D, Q) \geq t \} \\ & - c_2 NP \{ D \text{ rel and } \text{SIM}(D, Q) < t \}. \end{aligned} \quad (17)$$

By assuming that all values of the retrieval threshold  $t$  are equally likely, and integrating (17) for values of  $t$  between zero and infinity, one obtains a new utility expression:

$$U(R, Q) = N \sum_{k=1}^t \frac{q_k}{[\sum (q_k)^2]^{1/2}} (\alpha_k - \beta_k) - c_2 R + v_2 I, \quad (18)$$

where  $\text{SIM}(D, Q)$  has been replaced by the actual cosine computation of Eq. (4), and where  $\alpha_k$  and  $\beta_k$  are assumed equal, respectively, to the expected value of term  $k$  in a relevant and a nonrelevant item. If  $E$  denotes the expectation,  $\alpha_k$  and  $\beta_k$  are given as

$$\alpha_k = (v_1 + c_2) P \{ D \text{ rel} \} E \left[ \frac{d_k}{\left[ \sum_{i=1}^t (d_i)^2 \right]^{1/2}} \middle| D \text{ rel} \right]$$

and

$$\beta_k = (v_2 + c_1) P \{ D \text{ nonrel} \} E \left[ \frac{d_k}{\left[ \sum_{i=1}^t (d_i)^2 \right]^{1/2}} \middle| D \text{ nonrel} \right] \quad (19)$$

Equation (18) is in the form of an inner product between a query vector  $(q_1, \dots, q_t)$  and a document vector  $[(\alpha_1 - \beta_1), \dots, (\alpha_t - \beta_t)]$ . Hence the optimum query  $Q_{\text{opt}}$  will exhibit weights proportional to

Table 2. Retrieval characteristics for a typical query  $Q$ .

	Number of relevant	Number of nonrelevant	
Number retrieved	$r(v_1)$	$s(c_1)$	
Number not retrieved	$R - r(c_2)$	$I - s(v_2)$	
	$R$	$I$	$N$

$$U_k = (\alpha_k - \beta_k)K \quad (20)$$

for some constant  $K$ . The probabilities and expected values in Eq. (19) can be replaced with the corresponding frequencies from Table 1 as follows:

$$\begin{aligned} P\{D \text{ rel}\} &= R/N; P\{D \text{ nonrel}\} = I/N, \\ E[d_k | D \text{ rel}] &= r_k/R, \\ E[d_k | D \text{ nonrel}] &= s_k/I. \end{aligned}$$

If the average length of a document vector is set equal to  $L$ , a final *utility weighting function* for term  $k$  then becomes

$$U_k = (v_1 + c_2) \frac{r_k}{N\sqrt{L}} - (v_2 + c_1) \frac{s_k}{N\sqrt{L}}. \quad (21)$$

An appropriate term-weighting function for term  $k$  in document  $D_i$  could then be defined as

$$w_{ik} = f_{ik} \cdot U_k. \quad (22)$$

The term precision and term utility weighting systems of Eqs. (15) and (22) may be expected to be more powerful than the alternative weighting schemes based on simple term frequency characteristics [Eqs. (6) and (10)] in view of the distinctions made between term occurrences in the relevant and nonrelevant documents that are made in the former but not in the latter formulations.

#### E. Relationship between $r_k$ and $B_k$

To generate the term precision and term utility coefficients,  $L_k$  and  $U_k$ , respectively, it becomes necessary to identify the number of relevant and nonrelevant documents,  $r_k$  and  $s_k$ , in which a given term occurs. Furthermore, for the utility weights, actual numeric values must be chosen for the cost and value parameters of Eq. (16). In practice, the occurrence characteristics of the terms in the relevant and nonrelevant documents are not available before a search is actually conducted, although relevance information may be obtainable for certain documents retrieved in the early parts of an interactive search process. However, the total document frequency of term  $k$ ,  $B_k$

( $= r_k + s_k$ ), is given and that in turn might be used to estimate the number of relevant documents  $r_k$  including term  $k$ .

The document frequency of a term varies from zero for a term not assigned to any document in the collection to a maximum of  $N$  for a term assigned to all  $N$  items. The parameter  $r_k$ , on the other hand, varies from zero for a term not assigned to any relevant item to a maximum of  $R$ , the total number of relevant with respect to the given query  $Q$ . When the value of  $R$  is not known *a priori*, it could be chosen simply as the number of documents which a user wishes to retrieve in response to a given query.

The following relationships may exist between the total document frequency  $B_k$  of a term, and the frequency  $r_k$  in the relevant documents [16]:

- as  $B_k$  increases, so will  $r_k$ ; thus given two terms  $j$  and  $k$ ,  $B_j > B_k$  generally implies  $r_j > r_k$ ;
- for most query terms, the number of relevant documents in which a term occurs is relatively larger for lower-frequency terms than for higher-frequency terms; mathematically, one can say that when  $B_j > B_k$ , one finds that  $r_k/B_k > r_j/B_j$ . (For example, the one document in which a frequency-one term occurs is more likely to be relevant than the two documents for a term of frequency two.)

A possible functional relationship between  $r_k$  and  $B_k$  for a given term  $k$  is shown in Figure 3. Here for document frequencies between zero and  $R$ , one assumes that a straight-line relationship which exists between  $B$  and  $r$  is given as  $r = aB$  for some constant  $a < 1$ , and represented by line segment OA. For frequencies  $B$  between  $R$  and  $N$ , another straight-line relationship is assumed expressed as  $r = d + eB$  and represented by segment AC. It may be noted that in accordance with assumption (b) above the slope of line AC (represented by parameter  $e$ ) is smaller than the slope of line OA (parameter  $a$ ). As a result the proportion  $r_k/B_k$  is relatively larger for terms of smaller frequency  $B_k$  than for terms of larger frequency. An alternative functional relationship between  $r$  and  $B$  which also obeys assumptions (a) and (b) is  $r = a + b \log B$ .

It can be shown that if the relationship between  $B_k$  and  $r_k$  is the one shown in Figure 3, the term precision function  $L_k$  exhibits a variation with increasing document frequency of the kind represented in Figure 4 [16]. In particular, the precision weight of a term starts with some constant value  $a$  for terms of frequency one. The weight then increases as the document frequency of a term increases to  $R$ , the number of relevant documents which the user wishes to retrieve in response to a query. As the document frequency increases still further, the term becomes less important and the term weight decreases. Eventually, for terms of document frequency near  $N$ , the weight decays to 0. The frequency spectrum of Figure 4 shows again that the medium-frequency terms in a collection are the most important for purposes of document indexing.

The utility weighting function  $U_k$  of Eq. (21) exhibits variations with document frequency that are very similar to

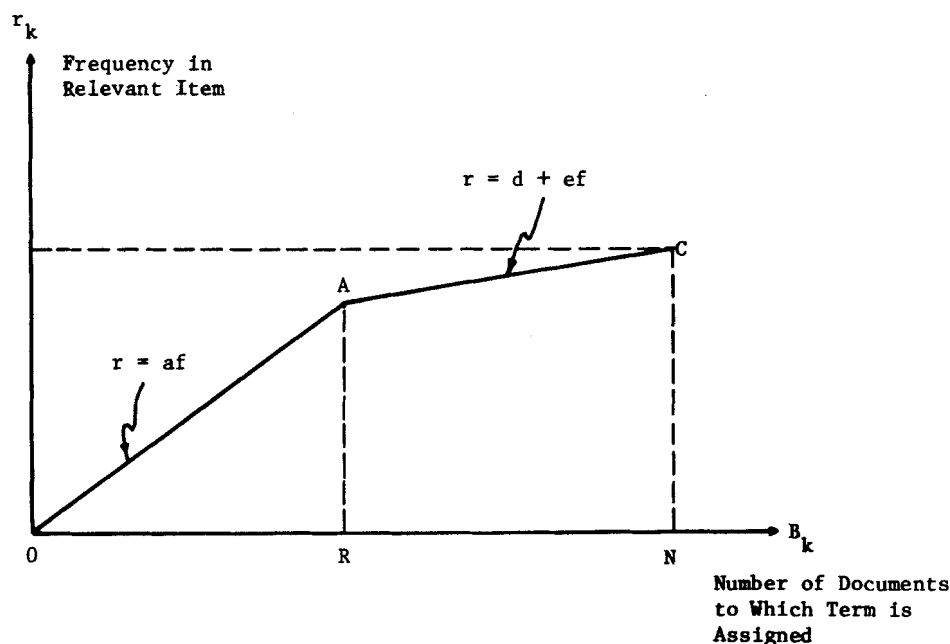


FIGURE 3. Variation of number of relevant items to which term  $k$  is assigned ( $r_k$ ) with document frequency  $B_k$ .

those demonstrated for the precision function [14]. In particular, assuming a logarithmic relationship between  $r$  and  $B$ , such as  $r = a + b \log B$ , one obtains by differentiation from Eq. (21):

$$\frac{dU}{dB} = \frac{1}{N\sqrt{L}} \left[ (v_1 + c_2 + v_2 + c_1) \frac{dr}{dB} - (v_2 + c_1) \right]$$

and

$$\frac{dr}{dB} = \frac{b}{B} (\log_2 e).$$

One concludes that  $dU/dB \geq 0$  when

$$B \leq \frac{(v_1 + c_2 + v_2 + c_1) b (\log_2 e)}{v_2 + c_1} = f_0$$

and  $dU/dB < 0$  when  $B > f_0$ . The variation of the utility term weight with the document frequency of the terms is summarized in Figure 5.

The foregoing development shows that all the term-weighting theories, except the IDF weighting of Eq. (6), follow the same general pattern in the sense that low-frequency terms carry relatively modest weight, medium-

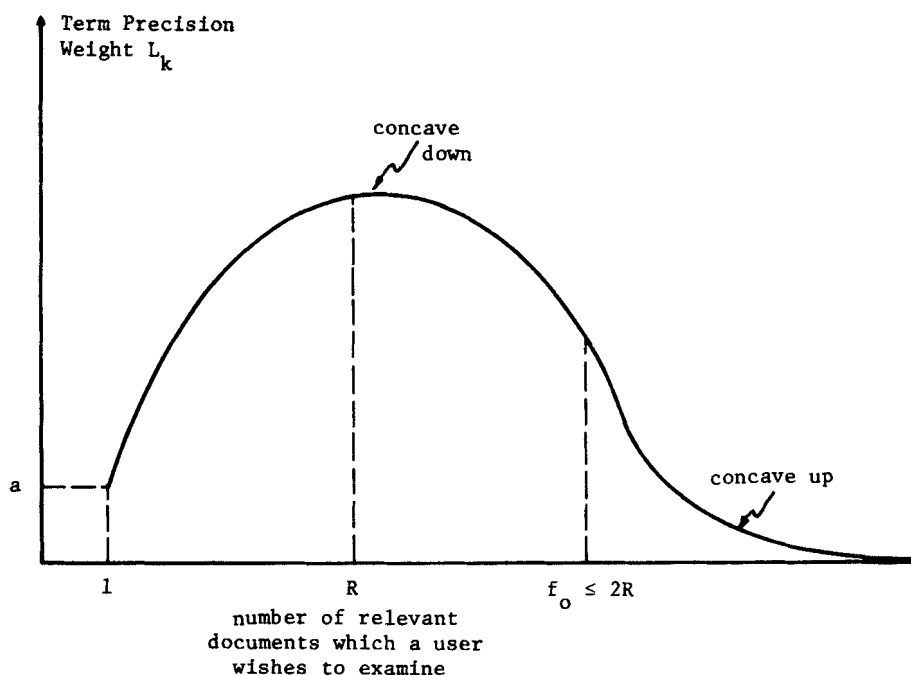


FIGURE 4. Variation of precision weight with document frequency.



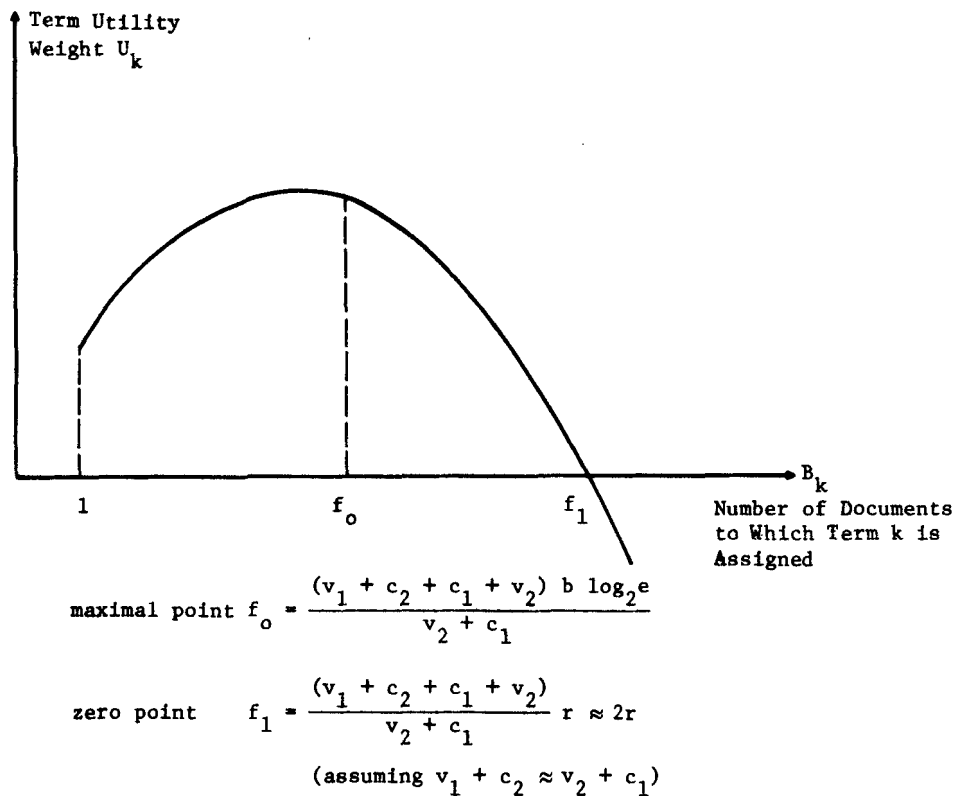


FIGURE 5. Variation of term utility weight with frequency.

frequency terms are best, and high-frequency terms are least attractive. For the IDF system, the weight decreases monotonically with document frequency; even that system follows the general pattern for all except the very-low-frequency terms.

#### 4. Experimental Evaluation of Term-Weighting Functions

Two collections of documents are used as examples to evaluate the various term-weighting systems, including the Cranfield collection of 424 documents in aerodynamics, and the Medlars collection of 450 documents in biomedicine. Each collection is used with 24 search requests in aerodynamics and medicine, respectively. Table 3 contains evaluation output for the standard TF weightings (5), the IDF weights (6), the term discrimination weighting (10), the term precision weights (15), and finally the term utility weights (22). In each case precision values are shown in Table 3 for ten recall levels varying from 0.1 to 1.0, averaged for the 24 search requests. It may be seen first that the IDF weights that are based on frequency characteristics over all documents regardless of relevance produce average improvements of about 27 and 15% in precision over the standard TF weights for the sample collections. The term discrimination weights appear somewhat less effective. When the utility weights based on document relevance are used, the average advantage in precision increases to 37 and 48%, respectively, while an even greater advantage of 46 and 74% is obtained for the term precision weights. The output of Table 3 is based on the assumption that the cost and value parameters associated with the relevant docu-

ments ( $v_1$  and  $c_2$ ) are given a total weight equal to 20 times the value and cost parameters associated with the nonrelevant documents ( $v_2$  and  $c_1$ ). The utility function of Eq. (21) is thus computed as  $20r - s$ . The precision weight formula (14) is slightly modified by addition of 0.5 factors to avoid division by zero. The precision weight used for the output of Table 3 is thus obtained as  $\{(r + 0.5)/[(R - r) + 0.5]\} \div \{s + 0.5\}/[(I - s) + 0.5]$ .

The problem of generating the  $r_k$  and  $s_k$  values was bypassed in the experiments of Table 3 by using the actual values found in the two sample collections for these parameters. That is, the unrealistic assumption was made that the occurrence characteristics of all terms in the relevant and nonrelevant documents were known in advance for all queries. This, of course, accounts for the excellent performance of the term utility and term precision weighting systems in the output of Table 3.

When the parameter values for  $r_k$  and  $s_k$  are no longer assumed to be available, one of the previously mentioned functional relationships between  $r_k$  and  $B_k$  can be used to obtain estimates for  $r_k$  given a particular value of the document frequency  $B_k$  of term  $k$ . Evaluation results for the term utility and term precision weighting systems based on estimated  $r_k$  values are included in Table 4 for the two document collections previously used in Table 3. For the utility weights a logarithmic relationship is assumed between  $r$  and  $B$  (that is,  $r = a + b \log B$  for suitably chosen parameter values  $a$  and  $b$ ). A hybrid function modeled on the relationship of Figure 3 is used to relate  $r$  and  $B$  for the precision weight calculation: in particular, a straight

TABLE 3. Evaluation of term-weighting systems.

Recall	Term Frequency	Inverse Document Frequency		Term Discrimination Value		Utility Weight $w = 20r - s$ (actual values)		Precision Weight (actual values)	
(a) Cranfield aerodynamics collection (424 documents, 24 queries)									
0.1	0.455	0.566	+24%	0.455	0%	0.568	+25%	0.571	+25%
0.2	0.410	0.530	+29%	0.484	+18%	0.540	+32%	0.558	+36%
0.3	0.391	0.476	+22%	0.454	+16%	0.503	+29%	0.503	+29%
0.4	0.301	0.421	+40%	0.361	+20%	0.474	+57%	0.479	+59%
0.5	0.280	0.364	+30%	0.330	+18%	0.416	+49%	0.434	+55%
0.6	0.233	0.301	+29%	0.277	+19%	0.328	+41%	0.352	+51%
0.7	0.189	0.254	+34%	0.236	+25%	0.272	+44%	0.324	+71%
0.8	0.155	0.195	+26%	0.174	+12%	0.211	+36%	0.220	+42%
0.9	0.121	0.150	+24%	0.129	+ 7%	0.162	+34%	0.199	+48%
1.0	0.112	0.132	+18%	0.121	+ 8%	0.143	+29%	0.164	+46%
			+27.6%		+16%		+37.6%		+46.2%
(b) Medlars biomedical collection (450 documents, 24 queries)									
0.1	0.543	0.611	+13%	0.460	+ 1%	0.676	+24%	0.707	+30%
0.2	0.528	0.601	+14%	0.439	+ 7%	0.676	+28%	0.707	+34%
0.3	0.467	0.541	+16%	0.422	+ 8%	0.639	+37%	0.705	+51%
0.4	0.421	0.467	+11%	0.316	+ 5%	0.609	+45%	0.672	+60%
0.5	0.384	0.438	+14%	0.288	+ 3%	0.558	+45%	0.633	+65%
0.6	0.346	0.396	+14%	0.252	+ 8%	0.510	+47%	0.616	+78%
0.7	0.316	0.347	+10%	0.208	+10%	0.459	+45%	0.573	+81%
0.8	0.211	0.245	+16%	0.174	+12%	0.374	+79%	0.462	+119%
0.9	0.171	0.193	+13%	0.138	+14%	0.277	+62%	0.354	+107%
1.0	0.120	0.154	+28%	0.127	+13%	0.204	+70%	0.259	+116%
			+14.9%		+ 8%		+48%		+74.1%

line similar to line segment OA of Figure 3 ( $r = aB$ ) is used for document frequency values  $B$  up to  $B = 8$ ; for larger values of  $B$  a logarithmic relationship ( $r = d + e \log B$ ) is assumed between  $r$  and  $B$ .

A comparison between the output of Tables 3 and 4 indicates that the utility and precision weighting systems are not as powerful when the parameter values must be estimated than when actual values are available. However, the precision weighting system appears to be more effective

than the IDF even when the relevance parameters are estimated. In many operational retrieval situations, several partial searches may be undertaken before a final, improved query formulation is generated [17,18]. In such circumstances, relevance assessments are often available for certain documents retrieved by the initial partial searches. These relevance assessments can then be used to estimate the  $r_k$  and  $s_k$  values for certain terms. Alternatively, the assumed functional relations between  $r_k$  and  $B_k$  can be used for this

TABLE 4. Estimated term utility and precision weight evaluation.

Cranfield Aerodynamics (424 documents, 24 queries)				Medlars Biomedical (450 documents, 24 queries)			
Utility Weight (estimated values)		Precision Weight (estimated values)		Utility Weight (Estimated values)		Precision Weight (estimated values)	
0.531	+17%	0.552	+21%	0.592	+ 9%	0.629	+16%
0.501	+22%	0.520	+27%	0.579	+10%	0.629	+19%
0.450	+15%	0.461	+18%	0.511	+ 9%	0.601	+29%
0.388	+29%	0.421	+40%	0.440	+ 5%	0.536	+27%
0.332	+19%	0.369	+32%	0.396	+ 3%	0.512	+33%
0.288	+24%	0.303	+30%	0.333	- 4%	0.456	+32%
0.234	+24%	0.259	+37%	0.309	- 2%	0.409	+29%
0.184	+19%	0.192	+24%	0.233	+10%	0.296	+40%
0.138	+14%	0.159	+31%	0.186	+ 9%	0.218	+27%
0.128	+14%	0.131	+17%	0.139	+16%	0.169	+41%
	+19.7%		+27.7%		+6.5%		+29.3%

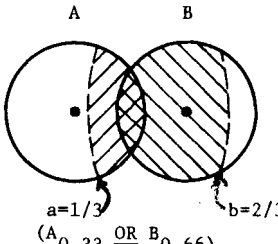
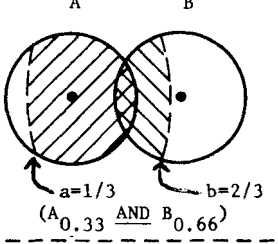
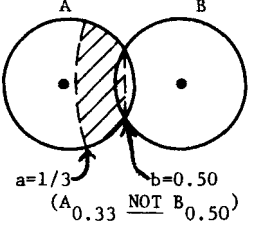
Boolean Operator	Query Statement	Output Document Set	Graphical Representation of Sample Queries
<u>OR</u>	$A \text{ OR } B_0$	$\underline{A}$	
	$A \text{ OR } B_1$	$\underline{A} \cup \underline{B}$	
	$A \text{ OR } B_b$	As $b$ increases from 0 to 1 output document set <u>expands</u> from $\underline{A}$ to $\underline{A} \cup \underline{B}$	
<u>AND</u>	$A \text{ AND } B_0$	$\underline{A}$	
	$A \text{ AND } B_1$	$\underline{A} \cap \underline{B}$	
	$A \text{ AND } B_b$	As $b$ increases from 0 to 1 output document set <u>shrinks</u> from $\underline{A}$ to $\underline{A} \cap \underline{B}$	
<u>NOT</u>	$A \text{ NOT } B_0$	$\underline{A}$	
	$A \text{ NOT } B_1$	$\underline{A} - \underline{B}$	
	$A \text{ NOT } B_b$	As $b$ increases from 0 to 1 output document set <u>shrinks</u> from $\underline{A}$ to $\underline{A} - \underline{B}$	

FIGURE 6. Interpretation of weighted Boolean operations.

purpose. In cases where the relevance weighting systems become too cumbersome, the simple IDF weighting still provides an acceptable standard of performance.

#### Appendix: Thresholding Operation for Weighted Query Terms

When weighted query terms are included in a Boolean query formulation, but the document terms are unweighted, it is necessary to identify the proportion of indexed documents to which the retrieval operation applies. One possible approach consists in modeling the operations as closely as possible on the case that applies to the normal unweighted Boolean query terms.

Consider, as an example, query statements of the form  $\{ \dots [(A_a * B_b) * C_c] \dots * Z_z \}$ , where  $*$  stands for one of the operators AND, OR, NOT, and where  $a, b, \dots, z$  represent weights attached to terms  $A, B, \dots, Z$ , respectively, such that  $0 \leq a \leq 1, \dots, 0 \leq z \leq 1$ . The general case involving a multiplicity of binary  $*$  operators may be reduced to that of a single binary operator with two operands by assuming that the search process is carried out iteratively, one operator at a time.

The problem then consists in interpreting query statements of the form  $(A_a * B_b)$  where  $*$  is a binary connective and  $a$  and  $b$  are the term weights. When the weighting factors are equal to one, the normal Boolean operations are

implied, so that  $A_1 * B_1 \equiv A * B$ , and  $A_a * B_1 \equiv A_a * B$ . On the other hand, when the weighting factors are equal to zero, the corresponding operands are disregarded, that is,  $A_a * B_0 \equiv A_a$  and  $A * B_0 \equiv A$ .

The operations of the three Boolean connectives may be described by considering the special case where only one of the two operands carries a weight less than one, that is, where the queries have the form  $(A * B_b)$ . Extensions to the general case where both query terms carry weights less than unity will then be immediate. Remembering that query term  $B_0$  can be disregarded, whereas  $B_1$  covers the full set  $B$  of documents indexed by  $B$ , it becomes clear that query  $(A \text{ OR } B_b)$  expands the output documents set from  $A$  to  $A \cup B$  as the weight of  $b$  increases from zero to one.  $A \cup B$  comprises the full set of items that are either  $A$ 's or  $B$ 's. A query such as  $(A \text{ OR } B_{0.33})$  must then retrieve all the  $A$ 's plus a third of the  $B$ 's. Correspondingly,  $(A \text{ AND } B_b)$  decreases the size of the output from  $A$  to  $A \cap B$ , i.e., to the set of items that are both  $A$ 's and  $B$ 's. This suggests that  $(A \text{ AND } B_{0.33})$  covers about two-thirds of the  $A$ 's, including all the  $A$ 's that are also in  $B$ . Finally,  $(A \text{ NOT } B_b)$  decreases the output from  $A$  to  $A - B$ , i.e., to the items in  $A$  that are not also in  $B$ . The query  $(A \text{ NOT } B_{0.33})$  would then cover all  $A$ 's that are not in  $B$  plus two-thirds of the items in  $A \cap B$ .

It remains to determine how the partial set of items that are either included in, or excluded from, the answering

document set is to be determined. The following mode of operation suggests itself:

(a) OR operation: as  $b$  increases from zero to one, the items in  $B$  not already in  $A$  that are *closest* to the set  $A$  are successively added to  $A$  to generate  $A \cup B$  in answer to query ( $A$  OR  $B$ ).

(b) AND operation: as  $b$  increases from zero to one, the items in  $A - B$  that are *furthest* from  $A \cup B$  are successively subtracted from  $A$  until only  $A \cap B$  remains in answer to query ( $A$  AND  $B$ ) when  $b = 1$ .

(c) NOT operation: as  $b$  increases from zero to one, the items in  $A \cap B$  that are *furthest* from  $A - B$  are successively subtracted from  $A$  until only  $A - B$  remains in answer to the query ( $A$  NOT  $B$ ).

To determine the closeness of a particular document to another document, or to a set of documents, the similarity coefficients previously introduced to compare queries and documents [Eqs. (2), (3), and (4)] can be used to obtain affinity indicators between pairs of documents, or between a particular document and a set of documents. In the latter case, a typical document is chosen to represent the given set, such as, for example, the *centroid*  $C$  of the set, and for each document  $D_i$  the size of the coefficient  $SIM(C, D_i)$  is used to indicate whether  $D_i$  is to be retrieved [1,2].

Consider as a typical example the operations for the query ( $A_{0.33}$  OR  $B_{0.66}$ ) illustrated in Figure 6 together with other examples:

- (a) compute the centroids of sets  $A$  and  $B$ ;
- (b) remove from set  $A$  two-thirds of the documents including all those exhibiting the largest distance to the centroid of  $B$ ;
- (c) remove from set  $B$  one-third of the documents including all those exhibiting the largest distance to the centroid of  $A$ ;
- (d) the response set is then the union of the remaining items from  $A$  and  $B$ .

To summarize, when weighted terms are used for queries that do *not* include Boolean operators, a query-document similarity computation can be used directly to obtain a retrieval value, or ranking, for each document. Documents may then be retrieved in decreasing order of their retrieval values. When Boolean operators are included in the queries, a similarity computation is first carried out between certain documents and the centroids, or representatives, of certain document sets. The size of the corresponding similarity coefficients then determines which documents are to be added to (for the OR operation) or subtracted from (for the AND and NOT operations) the basic answering set.

## References

1. Salton, G. *Automatic Information Organization and Retrieval*. New York: McGraw-Hill; 1968.
2. Van Rijsbergen, C. J. *Information Retrieval*, 2nd ed. London: Butterworths; 1979.
3. Bookstein, A. "Fuzzy Requests: An Approach to Weighted Boolean Searches." Paper presented at Joint Meeting of the Operations Research Society of America and the Institute of Management Science, New Orleans, May 1979.
4. Bookstein, A. "On the Perils of Merging Boolean and Weighted Retrieval Systems." *Journal of the American Society for Information Science*. 29(3):156-158; May 1978.
5. Noreault, T.; Koll, M.; McGill, M. J. "Automatic Ranked Output from Boolean Searches in SIRE." *Journal of the American Society for Information Science*. 28:333-341; November 1977.
6. Luhn, H. P. "A Statistical Approach to Mechanized Encoding and Searching of Literary Information." *IBM Journal of Research and Development*. 1(4):309-317; October 1957.
7. Sparck Jones, K. "A Statistical Interpretation of Term Specificity in Retrieval." *Journal of Documentation*. 28(1):11-21; March 1972.
8. Salton, G.; Yang, C. S. "On the Specification of Term Values in Automatic Indexing." *Journal of Documentation*. 29(4):351-372; December 1973.
9. Salton, G.; Yang, C. S.; Yu, C. T. "A Theory of Term Importance in Automatic Text Analysis." *Journal of the American Society for Information Science*. 26(1):33-44; January-February 1975.
10. Robertson, S. E.; Sparck Jones, K. "Relevance Weighting of Search Terms." *Journal of the American Society for Information Science*. 27(3):129-146; May-June 1976.
11. Yu, C. T.; Salton, G. "Precision Weighting—An Effective Automatic Indexing Method." *Journal of the ACM*. 23(1):76-88; January 1976.
12. Van Rijsbergen, C. J. "A Theoretical Basis for the Use of Co-occurrence Data in Information Retrieval." *Journal of Documentation*. 33(2):106-119; June 1977.
13. Bookstein, A.; Kraft, D. "Operations Research Applied to Document Indexing and Retrieval Decisions." *Journal of the ACM*. 24(3):418-427; July 1977.
14. Cooper, W. S.; Maron, M. E. "Foundations of Probabilistic and Utility-Theoretic Indexing." *Journal of the ACM*. 25(1):67-80; January 1978.
15. Salton, G.; Wu, H. "A Term Weighting Model Based on Utility Theory." Paper presented at Symposium on Research and Development in Information Retrieval, Cambridge, England, June 1980.
16. Yu, C. T.; Lam, K.; Salton, G. *Optimum Term Weighting in Information Retrieval Using the Term Precision Model*. Ithaca, NY: Computer Science Department, Cornell University; 1980.
17. Rocchio, J. J., Jr. "Relevance Feedback in Information Retrieval." In: Salton, G., Ed. *The Smart System—Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice-Hall; 1971: Chap. 14.
18. Salton, G. "Relevance Feedback and the Optimization of Retrieval Effectiveness." In: Salton, G., Ed. *The Smart Retrieval System—Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice-Hall; 1971: Chap. 15.