On Relevance, Probabilistic Indexing and Information Retrieval*

M. E. Maron

The RAND Corporation, Santa Monica, California

and

J. L. Kuhns

Ramo-Wooldridge, Canoga Park, California

Abstract. This paper reports on a novel technique for literature indexing and searching in a mechanized library system. The notion of *relevance* is taken as the key concept in the theory of information retrieval and a comparative concept of relevance is explicated in terms of the theory of probability. The resulting technique called "Probabilistic Indexing," allows a computing machine, given a request for information, to make a statistical inference and derive a number (called the "relevance number") for each document, which is a measure of the probability that the document will satisfy the given request. The result of a search is an ordered list of those documents which satisfy the request ranked according to their probable relevance.

The paper goes on to show that whereas in a conventional library system the cross-referencing ("see" and "see also") is based solely on the "semantical closeness" between index terms, statistical measures of closeness between index terms can be defined and computed. Thus, given an arbitrary request consisting of one (or many) index term(s), a machine can elaborate on it to increase the probability of selecting relevant documents that would not otherwise have been selected.

Finally, the paper suggests an interpretation of the whole library problem as one where the request is considered as a clue on the basis of which the library system makes a concatenated statistical inference in order to provide as an output an ordered list of those documents which most probably satisfy the information needs of the user.

1. *Introduction*

One of the really remarkable characteristics of human beings is their ability to communicate with and operate on information formulated in ordinary language. We somehow are able to determine the meanings of words and sentences so as to make judgments about sameness of meaning, redundancy, inconsistency, relevance, etc., in spite of the fact that ordinary language is extremely complex and fraught with vagueness and ambiguity. Since there are no strict rules which prescribe how words are to be put together to convey various kinds and shades of meanings, it is difficult indeed to think of using machines to perform the following kinds of operations on ordinary language: automatic analysis to detect and remove redundant information, automatic abstracting of relevant information, automatic verification of information (i.e., given some items of data,

* Received October, 1959. This research was carried out while both authors were at Ramo-Wooldridge and was supported by the United States Air Force with funds from Contract No. AF30(602)-1814, monitored by the Rome Air Development Center, New York.

deciding whether or not they are inconsistent with any other data already in storage), automatic deduction (i.e., logical derivation), automatic correlation of data so as to establish trends and deviations from trends, and so on. Yet, if the capabilities of digital computers are to be exploited to the fullest extent, we would hope that someday they can be programmed to operate on ordinary language on the basis of its meaning (content). It appears that as a first step in the direction of the automatic processing of ordinary language, as typified by the above examples, the problems of information identification and retrieval must be met and dealt with successfully. We therefore turn our attention to the problems of mechanizing a library.

There are a number of obvious difficulties associated with the so-called "library problem" (i.e., the problem of information search and retrieval). The one usually cited relates to the fact that documentary data are being generated at an alarming rate (the growth rate is exponential—doubling every 12 years for some libraries), and consequently considerations of volume alone make the problem appear frightening. However, the heart of the problem does not concern size, but rather it concerns meaning. That is to say, there have been a number of "hardware" solutions to the problem of library size (e.g., use of microfilm, microcards, minicards, magnacards, etc.), but the major difficulties associated with the library problem remain, namely, the identification of content, the problem of determining which of two items of data is "closer" in meaning to a third item, the problem of determining whether or not (or to what degree) some document is *relevant* to a given request, etc.

The jumping-off point for our approach to automatic information retrieval is the recognition that the core of the problem is that of adequately identifying the information content of documentary data. In the discussion that follows we introduce arithmetic (as opposed to logic alone) into the problem of indexing and thereby pave the way for the use of mathematical operations so as to compute a number, called the relevance number, which is a measure of the probable relevance of a document for a requestor. Thus, the fundamental notion which acts as a wedge to drive an opening into the basic problem of information retrieval is that of the relevance number, which provides a means of ranking documents according to their probable relevance. However, the solution to the problem of information retrieval involves more than ranking by relevance—it involves the proper selection of those documents which are to be ranked. In order to get at this "selection" problem, it is necessary to establish various measures of closeness of meaning, and an approach to this semantical problem is via statistics. We define various measures of closeness between documents and between requests for information so that given an arbitrary request a machine can automatically elaborate upon a search in order to retrieve relevant documents which would not otherwise have been selected.

We divide the paper into three parts: (a) a discussion of the conventional approach to the library problem, (b) an exposition of the solution given by Probabilistic Indexing, (c) a discussion of some preliminary experiments to test these new techniques and procedures.

2. *Conventional Approach to an Automatic Retrieval System*

2.1. *The Role of Indexes.* Because, at least for the immediate future, no machine can actually read a document and decide whether or not its subject matter relates to some given request subject, it is necessary to use some intermediate identifying tags, namely, an indexing system. An index to a document acts as a tag by means of which the information content of the document in question may be identified. The index may be a single term or a set of terms which together tag or identify the content of each document. The terms which constitute the allowable vocabulary for indexing documents in a library form the common language which bridges the gap between the information in the documents and the information requirements of the library users.

In principle, an indexer reads an incoming document, selects one or several of the index terms from the "library vocabulary," and then coordinates the selected terms with the given document (or its accession number). Thus, the assignment of terms to each document is a go or no-go affair—for each term either it applies to the document in question or it does not. Furthermore, the processes of indexing information and that of formulating a request for information are symmetrical in the sense that, just as the subject content of a document is identified by coordinating to it a set of index terms, so also the subject content of a request must be identified by coordinating to it a set of index terms. Thus, the user who has a particular information need identifies this need in terms of a library request consisting of one or several index terms or logical combinations thereof.

2.2. *The Mechanization.* Given a set of tags (index terms) which identify the content of each document and a set of tags which describe a request for information, the problem of automatic searching resolves itself to that of searching for and matching tags or combinations thereof. Once a set of index terms has been assigned to each document in the library, this information can be encoded in digital form, put on a suitable machine medium, and searched automatically. In the past, literature searching has been done automatically on punched cards using the IBM sorter, on magnetic tape using an electronic computer such as the IBM 709, on photographic film as a continuous strip such as in the case of the Rapid Selector, or in discrete records on photographic film as, for example, in the Minicard system. In each case a machine searches and retrieves (either copies of the document, abstracts, or a list of accession numbers) by matching document index terms with the terms and logic of the given request.

2.3. *The Notion of Semantic Noise.* The correspondence between the information content of a document and its set of indexes is not exact because it is extremely difficult to specify precisely the subject content of a document by means of one or several index words. If we consider the set of all index terms on the one hand and the class of subjects that they denote on the other hand, then we see that there is no strict one-to-one correspondence between the two. It turns out that given any term there are many possible subjects that it could denote (to a greater or lesser extent), and, conversely, any particular subject of knowledge (whether broad or narrow) usually can be denoted by a number of different terms. This situation may be characterized by saying that there is "semantic noise" in the index terms. Just as the correspondence between the information content of a document and its set of indexes is not exact, so also the correspondence between a user's request, as formulated in terms of one or many index words, and his real need (intention) is not exact. Thus there is semantic noise in both the document indexes and in the requests for information.

One of the reasons that the index terms are noisy is due to the fact that the meanings of these terms are a function of their setting. That is to say, the meaning of a term in isolation is often quite different when it appears in an environment (sentence, paragraph, etc.) of other words. The grammatical type, position and frequency of other words help to clarify and specify the meanings of a given term. Furthermore, individual word meanings vary from person to person because, to a large degree, the meanings of the words are a matter of individual experience. This is all to say that when words are isolated and used as tags to index documents it is difficult to pin down their meanings, and consequently it is difficult to use them as such to accurately index documents or to accurately specify a request.

2.4. *Conventional Stopgaps.* Many workers in the field of library science have attempted to reduce the semantic noise in indexing by developing specialized indexing systems for different kinds of libraries. An indexing system tailored to a particular library would be less noisy than would be the case otherwise. (In a sense, to tailor an index system to a specific library is to apply the principle of an ideoglossary, as it is used in machine language translation, to remove semantic ambiguity.) In spite of careful work in the developing of a "best" set of tags for a particular library, the problem of semantic noise and its consequences remain, albeit, to a lesser extent.

Another attempt to remove the semantic noise in request formulations has to do with the use of logical combinations of index terms. That is to say, if two or more index terms are joined conjunctively, it helps to narrow or more nearly specify a subject. On the other hand, the same set of terms connected disjunctively broadens the scope of a request.[1] Thus, using logical combinations of index terms, one would hope to either avoid the retrieval of irrelevant material or avoid missing relevant material. However, although a request using a set of index terms joined conjunctively does decrease the probability of obtaining irrelevant material, it also increases the probability of missing relevant material. The converse holds for a request consisting of a disjunction of index terms. This difficulty in the conventional approach is inherent in its go or no-go nature.

The fact that conventional searching consists in matching noisy tags implies that the result of a search provides documents which are irrelevant to the real needs of the requestor, and, even worse, some of the really relevant documents are not retrieved. Thus, in spite of specialized indexing systems and in spite of the

[1] We use the words "conjunction" and "disjunction" to denote the logical connectives "and" and "or." This usage is continued when classes, instead of propositions, are under discussion (as is the case for a possible interpretation of the index terms mentioned above), although in this case "intersection" and "union" are more appropriate.

use of logical combinations of index terms, the major problem is still that of properly identifying the subject content of both documents and requests. The problem of accurately representing the information content of a document by means of some kind of tags in such a way that a machine can operate on these tags in order to search for documents with the same meaning, related meanings, relevant meanings, etc., is still unsolved.

In the following section we shall present the basic notions of the technique of Probabilistic Indexing and show that this approach to the library problem improves retrieval effectiveness both by reducing the probability of obtaining irrelevant documents and by increasing the probability of selecting relevant documents. Furthermore, the technique of Probabilistic Indexing provides as the result of a search an ordered list of those documents which satisfy the request, ranked according to relevance.

3. *Derivation of the Relevance Number*

3.1. *Initial Remarks.* To say that index tags are noisy is to say that there is an uncertainty about the relationship between the terms and the subjects denoted by the terms. That is to say, given a document indexed with its assigned term (or terms), there is only a probability that if a user is interested in the subject (or subjects) designated by the tag he will find that the document in question is relevant. Conventional indexing consists in having an indexer decide on a yes-no basis whether or not a given term applies for a particular document. Either a tag is applicable or it is not—there is no middle ground. However, since there is an uncertainty associated with the tags, it is much more reasonable and realistic to make this judgment on a probabilistic basis, i.e., to assert that a given tag may hold with a certain degree or weight. Given the ability to weight index terms, one can characterize more precisely the information content of a document. The indexer may wish to assign a low weight such as 0.1 or 0.2 to a term, rather than to say that the term does not hold for the document. Conversely, the indexer may wish to assign a weight of 0.8 or 0.9 to a term, rather than to say that it definitely holds for a document. Thus, given weighted indexing, it is possible to more accurately characterize the information content of a document. The notion of weighting the index terms that are assigned to documents and using these weights to compute relevance numbers is basic to the technique which we call *Probabilistic Indexing.*

3.2. *Notions of Relevance and Amount of Information.* One of our basic aims is to rank documents according to their relevance, given a library request for information. The problem then is to take relevance, which is a primitive notion, and explicate it—in the sense of making the concept precise—hopefully, to give a quantitative explication. If we cannot have a quantitative measure for relevance, at least we would like a comparative measure so that ranking of documents by relevance will be possible. In some sense the problem of explicating the notion of relevance (which is the basic concept in a theory of information retrieval) is similar to that of explicating the notion of amount of information (which is the basic concept of communication theory). In Shannon's work on information theory we find that one notion of amount of information has been explicated in terms of probabilities so that one can establish a quantitative measure of the amount of information in a message.[2] We approach the notion of relevance also in a probabilistic sense.

3.3. *First Step.* By "$P(A,B)$" we mean the probability of an event of class B occurring with reference to an event of class A. We shall be interested in the following classes of events:

(a) D_i : obtaining the *i*th document and finding it relevant.

(b) I_j : requesting information on the field of interest (subject, area of knowledge) designated by the *j*th index term I_j.

(c) A: requesting information from the library.

Thus

$P(A.I_j,D_i)$ = the probability that if a library user requests information on I_j, he will be satisfied with document D_i.

As the first step in the explication of relevance we assert:

If $P(A.I_j,D_1) > P(A.I_j,D_2)$, then D_1 is more relevant than D_2.

3.4 *Next Step.* In the elementary calculus of probability, one can immediately derive by the inverse inference schema (closely associated with Bayes' Theorem):

$$P(A.I_j,D_i) = \frac{P(A,D_i)\cdot P(A.D_i,I_j)}{P(A,I_j)}. \tag{1}$$

For any given request I_j, $P(A,I_j)$ is a constant, and consequently we may rewrite (1) as follows:

$$P(A.I_j,D_i) \sim P(A,D_i)\cdot P(A.D_i,I_j) \tag{2}$$

where $P(A,D_i)$ is the a priori probability of document D_i, and $P(A.D_i,I_j)$ is the probability that if a user wants information of the kind contained in document D_i he will formulate a request by using I_j. Thus, if we can obtain the values called for in the right-hand side of (2), then we can compute a quantity proportional to the value $P(A.I_j,D_i)$. We call this quantity the *relevance number* of the *i*th document with respect to the given request.

Immediately the problem of estimating these values confronts us. Consider first the estimate of $P(A.D_i,I_j)$. In principle, one could obtain an estimate of this probability via a statistical sampling process, but such a procedure would of course be extremely impractical, and it turns out that it is unnecessary. When an individual indexes a document (i.e., when he decides which terms to use to tag a document) he intuitively estimates this probability—in the conventional case automatically converting the probabilities to either 0 or 1. We now assert that

[2] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication,* The University of Illinois Press, Urbana, Illinois (1949).

the weight of a tag for a document, i.e., the degree with which the tag holds for a document, when properly scaled, can be interpreted as an estimate of $P(A.D_i,I_j)$. Thus we have from (1) and (2)

$$P(A.I_j,D_i) = \alpha_j \cdot P(A,D_i) \cdot \omega_{ij}, \tag{3}$$

where "ω_{ij}" denotes *the degree to which the j-th index term applies to the i-th document*, $P(A,D_i)$ is the a priori probability of document D_i, and α_j is the scaling factor times the reciprocal of $P(A,I_j)$. If we interpret the a priori probability so that it corresponds to statistics on document usage, i.e., so that $P(A,D_i)$ is the quotient of the number of uses of document D_i by the total number of document uses, then it is easy to show on the basis of (1) and (3) that ω_{ij} and $P(A.D_i,I_j)$ are related by

$$\omega_{ij} = \beta_j \cdot P(A.D_i,I_j), \tag{4}$$

where

$$\beta_j = \frac{\sum_i P(A,\, D_i) \cdot \omega_{ij}}{P(A,\, I_j)}. \tag{5}$$

In other words, in (4), β_j plays the role of an error factor in the estimation of $P(A.D_i,I_j)$ by the value ω_{ij} ; (5) tells us how to estimate, in turn, the value of this error factor by using all the weights for a given tag as well as the statistical data $P(A,I_j)$. An improved estimate, therefore, is given by redefining the weight of a tag as follows:

$$w_{ij} = \omega_{ij}/\beta_j = \text{improved estimate of } P(A.D_i,I_j). \tag{6}$$

We call this value "the modified weight." The reader is referred to the Appendix for the details on the extension of the weight of a single index term to the weight of any Boolean function of index terms.

To summarize, library statistics provide us with $P(A,D_i)$, the weights coordinated with the index terms, when properly scaled, give us estimates of $P(A.D_i,I_j)$, and consequently we can compute the value of the relevance number $P(A.I_j,D_i)$ by means of which documents can be ranked according to their probable relevance to the requestor.

3.5. *Question Concerning Estimation.* At this point one might raise the following question. If the indexer is required to estimate $P(A.D_i,I_j)$, why not have him estimate $P(A.I_j,D_i)$ directly, since this is the goal of the computations? Actually this is not quite correct since the general goal of the computations is the determination of $P(A.R,D_i)$, where R is any Boolean function of the index terms. Clearly the indexer cannot make a single estimate of $P(A.R,D_i)$ since the value depends on the particular request R and hence there would have to be as many estimates of $P(A.R,D_i)$ as there would be R's. In order to avoid this situation, it would be necessary to transform $P(A.R,D_i)$ so that R goes from the reference class to the attribute class. Once we do this the problem is to compute $P(A.D_i,R)$ given all of the values of $P(A.D_i,I_j)$ as j varies over the range of tags that are contained in R. Thus, it turns out that $P(A.R,D_i)$ cannot be estimated directly—we must obtain it from $P(A.D_i,R)$, which in turn must be computed from

$P(A.D_i,I_j)$.[3] That is to say, we need the value of $P(A.D_i,I_j)$ anyway, and any other estimates are superfluous.[4]

4. *Automatic Elaboration of the Selection Process*

4.1. *Initial Remarks.* The technique of Probabilistic Indexing, as we have seen, allows a computing machine, given a request for information, to derive a relevance number for each document. This relevance number is a measure of the document's relevance. The result of a search is an ordered list of those documents which satisfy the request, ranked according to their relevance numbers. We would prefer to have a technique which not only decides which of a given class of documents is most probably relevant, next most probably relevant, etc., but which also decides whether the class itself of retrieved documents is adequate (at least in the sense of determining whether or not it excludes some documents which are relevant to the user's *information needs* but are not computed as probably relevant due to inadequacies in the user's description of his information need). That is to say, if we consider the request as a *clue* which the user gives to the library to indicate the nature of his information needs, then we should raise the following question: Given a clue, how may it be used by the library system to generate a best *class* of documents (to be ranked subsequently by their relevance numbers)? Thus given the clue, how can we elaborate upon it automatically in order to produce a best class of retrieved documents? Let us turn our attention to this problem.

4.2. *Search Strategies and the Notion of Closeness.* A library request (a clue) is a Boolean function whose variables are index terms, each of which, in turn, selects a class of documents via a logical match. That is to say, all of those documents whose index terms are logically compatible with the logic and the tags of a request R constitute the class of retrieved documents C. Our goal is to extend the class C in the most probable "direction," and this can be done in two ways. One method involves the transforming of R into R', where R' in turn will select a class of documents C' which is larger than C and contains more relevant documents. A second method does not modify R but rather uses the class C to define a new class C''. A set of rules which prescribe how to go from a given request R to a class of retrieved documents is called a strategy. A strategy, in turn, involves the use of several different techniques for measuring the "closeness" between index terms and between documents. Before proceeding, let us introduce some further notations to make more precise what we have been saying.

[3] See Appendix.

[4] Even if every request were of the elementary form $P(A.I_j, D_i)$, it would be better to estimate $P(A.D_i,I_j)$ and compute $P(A.I_j,D_i)$ rather than to estimate the latter directly. This second argument in favor of the estimation of $P(A.D_i,I_j)$ over $P(A.I_j,D_i)$ appears when we consider the consistency of the comparative values. The indexer looks at each document, then runs through the various possible index terms which apply. In general $P(A.D_i,I_j)$ will vary over a much larger range than $P(A.I_j,D_i)$ as j varies, and therefore it is easier psychologically for the indexer to rank correctly the values over the larger range.

We understand by "basic selection process" the rule which uses the request to select the class of documents whose tags are logically compatible with the logic and tags of the request, and we denote this basic selection process by the functional notation "f". Thus f is the transfer function from inputs (requests) to output (class of retrieved documents) and we write

$$f(R) = C \tag{7}$$

where, again, R is the request and C is the class of retrieved documents. The problem is to enlarge C so as to increase the probability that it will contain relevant documents and to decrease the probability that it will contain irrelevant documents. We approach this problem in the following way: Suppose R' is a request similar in meaning to R; then we can take as a possible modification of f, say f',

$$f'(R) = f(R) \vee f(R') = C \vee C' \tag{8}$$

(where "$\vee$" designates class union). This modification can be made precise if we are able to invent a closeness measure on the request space to measure similarity in meaning. Since we are not sure what "meaning" is and much less able to assign a numerical quantity to it, this is rather difficult; but we shall show later that statistics can provide such measures. For the present, suppose we actually do have such a measure; then we can generate a modified selection function f' by defining $f'(R)$ to be the union of all classes $f(R')$ where the "closeness" between R and R' exceeds some specified number, say ϵ. Symbolically, this is written

$$f'(R) = \bigcup_{[Q(R,\,R') > \epsilon]} f(R'). \tag{9}$$

Analogously, if we have a "distance" function in the document space[5] which gives "distance" as a numerical measure of dissimilarity of information content, then a completely different modification f'' of f arises via

$$f''(R) = C'' \tag{10}$$

where C'' consists of all documents whose distance from $C = f(R)$ is less than ϵ.

Thus, we see that a machine strategy can elaborate upon the basic selection process in order to improve the search in one of two different ways. The first is to establish a measure for "closeness" in request space so as to formulate R', given R. The other way is to use the class of documents C, obtained by the initial request R, to define a new class C''. Both of these methods are discussed below.

4.3. *Notion of Index Space.* Geometrically speaking, one may think of the set of n index terms which constitute the library catalog "vocabulary" as points in an n-dimensional space. The points in this space are not located at random, but rather, they have definite relationships with respect to one another, depending on the meanings of the terms. For example, the term "logic" would be much

[5] We use "distance" in document space, "closeness" in request space. The reason is that our measures in request space have the nature of "coefficients of association" rather than the properties of mathematical distance functions.

closer to "mathematics" than to "music." One always finds, when looking up index terms in the catalog of a conventional library, other terms listed under "see" and "see also." This cross-indexing ("see/see also") aspect of a library indicates some of the relationships that index terms have for one another; i.e., it indicates some of the relationships between points in index space.

The numerical evaluation of relationships between index terms can be made explicit by formulating probabilistic weighting factors between them. Once numerical weighting factors are coordinated with the distances, the cross-indexing aspect of a library can be mechanized so that, given a request involving one (or many) index terms, a machine could compute other terms for which searches should be made. That is to say, a request places one at a point, or several points, in index space, and, once the "closeness" measures between points are arithmetized,a machine could determine which other points to go to in order to improve the request. Thus, the elaboration of a request on the basis of a probabilistic "association of ideas" could be executed automatically.

4.4. *Automatic Groping in Index Space.* There are at least two different kinds of relationships that can exist between the points in index space, viz., semantical relationships and statistical relationships. The most elementary semantical relationship is that of synonymity, but in addition to synonymity there are other semantical relationships such as "partially implied by" and "partially implies." Such relationships between terms are based strictly on the meanings of the terms in question—hence the word "semantical." Another class of relationships is statistical, i.e., those based on the relative frequency of occurrence of terms used as indexes. The distinction between semantical and statistical relationships may be clarified as follows: Whereas the semantical relationships are based solely on the meanings of the terms and hence independent of the "facts" described by those words, the statistical relationships between terms are based solely on the relative frequency with which they appear and hence *are* based on the nature of the facts described by the documents. Thus, although there is nothing about the meaning of the term "logic" which implies "switching theory," the nature of the facts (viz., that truth-functional logic is widely used for the analysis and synthesis of switching circuits) "causes" a statistical relationship. (Another example might concern the terms "information theory" and "Shannon"—assuming, of course, that proper names are used as index terms.)

Once the various "connections" between the points of index space have been established, rules must be formulated which describe how one should move in the maze of connected points. We call such rules "heuristics." They are general guides for groping in the "maze" in the attempt to create an optimal output list of documents for any arbitrary request. The heuristics would enable a machine to decide, for a given set of request terms, which index terms to "see" and "see also," and how deep this search should be and when to stop, etc. Generally speaking, the heuristics would decide which index terms to look at next, on the basis of the semantical and statistical connections between terms, and the heuristics would decide when to stop looking, on the basis of the number of documents that would be retrieved and the relevance numbers of those documents. (Remember that each point in index space defines a class of documents, viz., all of those documents which have been assigned the index terms in question with a nonzero weight.) Given this understanding of heuristics, we see that an over-all search strategy is made up of components, some of which are heuristics; i.e., the sequence of devices, rules, heuristics, etc., which lead from inputs (requests) to outputs (classes of retrieved documents) is the strategy.

4.5. *Three Measures of Closeness in Index Space.* In order to clarify the notion of developing heuristics which would determine how a computer should "grope" in index space, consider the following example. Assume that we compute the frequency, $N(I_j)$, with which each term is used to tag a document, and also that we compute the frequency, $N(I_j.I_k)$, with which pairs of terms are assigned to documents. We can then compute the conditional probability $P(I_j,I_k)$ that if a term I_j is assigned to a document then I_k also will be assigned:

$$P(I_j, I_k) = \frac{N(I_j \,.\, I_k)}{N(I_j)} \tag{11}$$

We do this for all pairs I_j, I_k.

Assume now that I_j' is the index term which has the highest conditional probability given I_j; i.e., I_j' is the index term for which $P(I_j,I_k)$ is a maximum. Then given a request, $R = I_j$, for all documents tagged with I_j, we form a new request, $R' = I_j \vee I_j'$, which searches for all documents tagged with either I$_j$ or I_j'. Thus, the rule is now to consider R' instead of R.

This procedure tells us which tags are closest (in one sense) to given ones, but we still have no *measure* of the "closeness" (hereafter written without quotes) and such a measure is needed as a part of the associated computation rule. That is to say, we elaborate upon R and obtain R' by searching for documents indexed under tags closely related to those in the original request, but clearly the relevance numbers that we derive for these additional documents should be weighted down somewhat in order to indicate that they were obtained only from tags which are close to those in the original request. We measure the closeness as follows: Let $p_j = P(I_j,I_j')$ and normalize p_j over the set of tags used in the request so that

$$\bar{p}_j = \frac{p_j}{\sum p_j} \,.$$

Now, instead of using $w_i(I_j')$ (the weight assigned to I_j' for the ith document) in the search computation, we replace it by $\bar{p}_j \cdot w_i(I_j')$. The extended search that we have just described is an elementary form of only one of a class of possible heuristics based on the statistical relationships between tags.

A second elementary heuristic which looks even more promising is called the "inverse conditional" search, and it involves measuring closeness of tags to I_j in terms of the conditional probability from I_k to I_j (instead of conversely as described above). That is to say, we compute the $P(I_k,I_j)$ which is maximum as I_k varies, and this provides the tag which most strongly implies the given tag

I_j. Thus, instead of asking for that tag which is most strongly implied (statistically) by an arbitrary tag in the request, we ask for the tag which most strongly *implies* (statistically) the given tag. Using this method to determine the closeness of tags we establish a measure for the closeness by normalizing the probability as before. That is, define

$$p_j = P('I_j, I_j),$$
$$\bar{p}_j = \frac{p_j}{\sum p_j}$$

and, again, the corresponding computation rule is now $\bar{p}_j \cdot w_i('I_j)$, where $'I_j$ is the I_k which makes $P(I_k,I_j)$ a maximum for a given I_j.

Having discussed two possible measures of closeness, viz., the conditional probability $P(I_j,I_k)$ and the inverse condition probability $P(I_k,I_j)$, we now consider a third statistical measure which appears to be the most promising of the three. This is one of several possible *coefficients of association* between predicates.[6]

The particular coefficient we have chosen arises in the following way. Consider the tags I_j and I_k, and partition the library by four classifications, viz., documents indexed under both I_j and I_k, those indexed under I_j but not I_k, those indexed under I_k but not I_j, and those not indexed under either. Letting "$\bar{I}_j$" denote the complement of the class I_j, etc., these four classes are given by $I_j.I_k$, $I_j.\bar{I}_k$, $\bar{I}_j.I_k$, $\bar{I}_j.\bar{I}_k$, respectively. The classification and the number of documents is shown most conveniently in a table:

	I_k	$\bar{I}_k$	
I_j	$x = N(I_j.I_k)$	$u = N(I_j.\bar{I}_k)$	$N(I_j)$
$\bar{I}_j$	$v = N(\bar{I}_j.I_k)$	$y = N(\bar{I}_j.\bar{I}_k)$	$N(\bar{I}_j)$
	$N(I_k)$	$N(\bar{I}_k)$	n

We have adjoined to the table the row and column sums and n (the total number of documents).

Now, using the notation of formula (11), we say that I_j is *statistically independent* of I_k if

$$P(I_j,I_k) = P(I_k). \tag{15}$$

This can be shown to be equivalent to

$$P(I_j.I_k) = P(I_j) \cdot P(I_k); \tag{16}$$

so that rewriting in terms of frequencies we have an additional equivalence:

$$N(I_j.I_k) = N(I_j) \cdot N(I_k)/n. \tag{17}$$

[6] G. U. YULE, On measuring association between attributes, *J. Royal Stat. Soc.*, *75* (1912), 579–642.

For any pair I_j, I_k, (17) suggests that we look at the excess of $N(I_j.I_k)$ over its independence value; i.e., the quantity

$$\delta(I_j,I_k) = N(I_j.I_k) - N(I_j)\cdot N(I_k)/n. \tag{18}$$

It can be shown that this function δ has the property

$$\delta(I_j,I_k) = \delta(\bar{I}_j,\bar{I}_k) = -\delta(\bar{I}_j,I_k) = -\delta(I_j,\bar{I}_k), \tag{19}$$

and thus δ is associated with the difference over independence values in all four classifications. Yule[7] lists some basic properties that a coefficient of association between I_j and I_k should have. We call this coefficient "$Q(I_j,I_k)$". (1) $Q(I_j,I_k)$ should be zero when $\delta(I_j,I_k) = 0$ and, moreover, $Q(I_j,I_k)$ should vary as $\delta(I_j,I_k)$ for fixed n and fixed row and column totals; (2) the maximum of $Q(I_j,I_k)$ should occur when I_j is contained in I_k ($u = 0$), or I_k is contained in I_j ($v = 0$), or I_j and I_k give the same class ($u = v = 0$); (3) the minimum of $Q(I_j,I_k)$ should occur when I_k is contained in $\bar{I}_j$ ($x = 0$), or $\bar{I}_j$ is contained in I_k ($y = 0$), or I_j is the complement of I_k ($x = y = 0$); (4) it should have a simple range of values, say from -1, to 1. A coefficient[8] that has all of these properties is:

$$Q(I_j,I_k) = (xy - uv)/(xy + uv), \tag{20}$$

where the intimate connection with δ is indicated by the fact that the numerator of Q is, in fact, $n\delta$.

The generation of a heuristic now proceeds by the rules for the previous measures. Given $R = I_j$, we select the term I_k (different from I_j) with the maximum coefficient $Q(I_j,I_k)$. This value will be between 0 and 1, or no term will be selected. Then R is extended to

$$R' = I_j \vee I_k,$$

and in the search computation we multiply the weight $w_i(I_k)$ by $Q(I_j,I_k)$.

We now have the possibility of generating more elaborate heuristics. The heuristics just described can be called "one-deep." Applying the procedure again, we arrive at "two-deep" heuristics. At this second level, however, several possibilities arise. Having gone from I_j to $I_j \vee I_k$, we can now find the term most closely associated to I_k, say I_l, thus obtaining (two-deep chain search):

$$R'' = I_j \vee I_k \vee I_l.$$

Alternately, we can choose the term of second highest association with I_j, say I_m, thus obtaining (two-deep hub search)

$$R'' = I_j \vee I_k \vee I_m.$$

We also have the possibility of changing the measure of closeness for the second search, thus building as complex a search strategy as we wish.

[7] Loc. Cit.

[8] The coefficient recommended by Yule, *loc. cit.*, is not Q, but $Z = (\sqrt{xy} - \sqrt{uv})/(\sqrt{xy} + \sqrt{uv})$. The range of variation of both Q and Z is the same and since both lead to equivalent heuristics we have chosen Q for its computational simplicity. For refined work we might adopt Z.

4.6. *Heuristics in the Document Space.* Having shown how to generate heuristics by elaborating on the original requests, let us now look at the problem of implementing formula (10). We call such heuristics "extension heuristics"; i.e., we extend an initially retrieved class of documents by considerations concerning this class itself—holding that this class gives clues as to the meaning of the original request. Now we would prefer not only to extend this class by measures of distance between documents but also introduce such measures into a "generalized" relevance number computation. That is to say, we would like to combine heuristics in such a way that documents with associated ranking numbers are retrieved, not just classes of documents. We would also like to use the values $w_i(R)$ in the computation.

First, we note that the Pythagorean distance between two rows of the probabilistic matrix gives a measure of dissimilarity of information content (as well as dissimilarity of distribution of information) between documents corresponding to these rows. Call this distance, $\Delta(D_i,D_j)$. We can use this distance function to compute the distance of any document from the class C of documents retrieved by the basic selection process. This is all the theory required to implement formula (10).[9]

Next is the problem of defining the generalized relevance number. There are infinitely many possibilities here, and which is the "best" is still an open problem. However, an extremely natural one arises as follows: We consider the values $w_i(R)$ as measures of closeness between R and the documents. To combine these values with $\Delta(D_i,D_j)$, we convert closeness to "distance" by some device such as considering the negative of the logarithm of $w_i(R)$. We define

$$d(R,D_i) = -\log w_i(R). \tag{21}$$

D_i will be retrieved by R if and only if $d(R,D_i)$ is finite; thus this characterizes the class of retrieved documents. Now take that document D_i in the class of retrieved documents such that $\Delta(D_i,D_j)$ is a minimum.[10] Then we take

$$g(R,D_j) = \sqrt{\Delta^2(D_i,D_j) + \log^2 w_i(R)} \tag{22}$$

as the measure of "distance" between R and D_j.[11] Note that if D_j is a retrieved document, then $\Delta(D_i,D_j)$ is zero and

$$g(R,D_j) = -\log w_j(R).$$

Furthermore, if D_j has not been retrieved (initially),

$$g(R,D_j) > -\log w_i(R),$$

where i is the accession number of the document nearest to D_j. Thus the ranking by the g-function will always put an adjoined document below its associated

[9] Another measure of dissimilarity is to take, not the Pythagorean distance, but the sum of the absolute values of the differences between rows; i.e., $\sum_k | w_{ik} - w_{jk} |$; in fact, several other measures of dissimilarity appear worthy of study. Our discussion is perfectly general, and the reader may take $\Delta(D_i,D_j)$ as any such measure.

[10] If D_i is not unique, choose the one in the minimal set with the largest $w_i(R)$.

[11] It may be preferable to "weight" each of the components $\Delta(D_i,D_j)$, $\log w_i(R)$ in (22). Suitable values of these weights are to be determined by experimentation.

document in the class C. We may now finish the computation by subtracting the logarithm of the *a priori* probability of a document from its g-value (analogous to multiplying $w_i(R)$ by $P(A,D_i)$ to obtain the relevance number). The final heuristic is then obtained by choosing a suitable cutoff point in the list of adjoined documents—taking only those with generalized relevance numbers less than some specified value.

4.7. *Extension of the Request Language.* It is obvious that the richer the request language the more precise is the user's description of his information need. However, as the language becomes descriptively richer, the processing of retrieval prescriptions becomes more complex because of the difficulties discussed in section 1. A rather simple extension of our request language does, however, present itself. This richer language also has the virtue of adaptability to the automatic procedure we envisage. That is, we permit the requestor to assign numerical weights to index terms according to how important a role he wishes them to play in the processing of his request. These request weights can be used to scale down the index term weights and/or to serve as control numbers in search strategies.

4.8. *Search Strategies.* We have presented some of the heuristics that appear to have the best possibility of being useful components of a search strategy. We also have formulated some principles for a general approach to the problem of automatic elaboration of the selection process. Let us now illustrate these ideas by constructing an over-all search strategy.

First we list the variables involved:

1. Input
   a. The request R
   b. The request weights
2. The Probabilistic Matrix $[w_{ij}]$
   a. Dissimilarity measures between documents (e.g., Δ-values)
   b. Significance measures for index terms (An index term applied to every document in the library will have no significance, while an index term applied to only one document will be highly significant. Thus significance measures are related to the "extension number" for each term, i.e., to the number of documents tagged with the term—the smaller this number, the greater the significance of the index term.)
   c. "Closeness" measures between index terms (e.g., Q-values, etc.)
3. The a priori Probability Distribution
4. Output (by means of the basic selection process, i.e., the logical match plus the inverse inference schema (1) with all of its ramifications and refinements)
   a. The class of retrieved documents, C
   b. n, the number of documents in C
   c. Relevance numbers
5. Control Numbers
   a. n_0, the maximum number of documents that we wish to retrieve
   b. Relevance number control; e.g., we may ignore documents with relevance number less than a specified value.

c. Generalized relevance number control. Similar to the above, but this applies to the computation described in section 4.6.
d. Request weight control; i.e., we elaborate on index terms in the request if their request weight is higher than some specified value.
e. Significance number of index term control; i.e., we give index terms of certain significance (defined in terms of their extensions) special attention.

6. Operations
a. Basic selection process; denote this by "*f*".
b. Elaboration of the request by using "closeness" in the request space. Denote this by "*H*". Thus the operation H will transform the request R into a new request R'. More precisely H is the heuristic: elaborating the index terms in R with request weights greater than the request weight control number and/or index term significance greater than a specified value.
c. Adjoining new documents to the class of retrieved documents by using "distance" in the document space. Denote this by "*h*". Thus the operation h will transform the class C of retrieved documents into a new class, say D. More precisely, h is the heuristic: trim C to documents having relevance number greater than the control number and then annex to C all of the documents with generalized relevance number in a certain range.
d. Merge: any merging operation between two classes; e.g., forming their intersection, their union, trimming by using relevance number and then forming union, etc.

Next we combine these to obtain the strategy shown in figure 1. This strategy is to be regarded as a particularly simple example, its goal to obtain a specified number of documents (say n_0) having the best chance of satisfying the request. Thus, the decision to elaborate, centers on answering the question: Is the number of documents selected greater than or equal to n_0 ? In figure 1 we refer to the heuristic H as simply "elaborate the request." The actual transfer function H involves using control numbers to limit the elaboration. Furthermore, these control numbers can be varied from one application of H to the next. Similarly we refer to the heuristic H as simply "extend the class," but we point out that this too involves control numbers. Finally, a word about the classes C, C', D, etc. These are actually lists of documents ranked by relevance numbers. Thus the instruction "trim C to n_0" means "cut off the list to the n_0 documents with highest relevance numbers." The output of the system will be an ordered list of document accession numbers.

5. *Experimental Results*

5.1. *An Initial Remark.* The previous discussion (sections 3 and 4) indicates that there are two basic hypotheses that we wish to verify. The first hypothesis asserts that the relevance number which we compute for each document, given

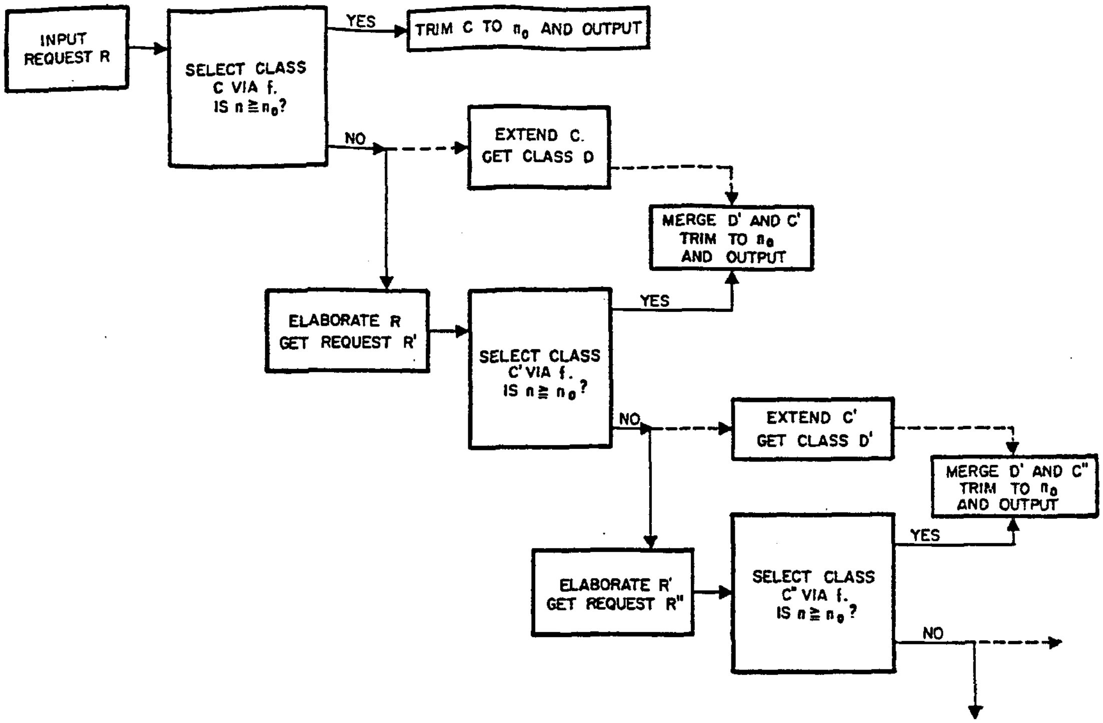
INPUT REQUEST R
SELECT CLASS C VIA f. IS n ≧ n₀?
YES
TRIM C TO n₀ AND OUTPUT
NO
EXTEND C. GET CLASS D
MERGE D' AND C' TRIM TO n₀ AND OUTPUT
ELABORATE R GET REQUEST R'
SELECT CLASS C' VIA f. IS n ≧ n₀?
YES
NO
EXTEND C' GET CLASS D'
MERGE D' AND C" TRIM TO n₀ AND OUTPUT
ELABORATE R' GET REQUEST R"
SELECT CLASS C" VIA f. IS n ≧ n₀?
YES
NO

FIG. 1. A search strategy based on heuristics of elaboration and extension

a request, is, *in fact*, a measure of the probable relevance of the document. The second hypothesis asserts that the automatic elaboration of a search does, *in fact*, produce relevant documents which are not retrieved by the original request. In order to empirically verify the above hypotheses and to gain further insights into these problems, we conducted some preliminary experiments, the results of which we now summarize.

5.2. *Experimental Setup.* A collection of 110 articles from *Science News Letter*[21] formed the experimental library. Our choice of articles from *Science News Letter* was dictated to a large extent by the fact that these articles are relatively brief, pithy, clearly written, interesting, and easy to index by nonexperts. This made not only the indexing but the subsequent evaluation of retrieved documents a reasonably uncomplicated task.

Since the methods of Probabilistic Indexing are applicable to any indexing system, we were not limited in our choice of a set of tags to be used. The only constraint was that the number of tags in the index list be comparable with the size of the library. Instead of "truncating" an existing index system and using its tags to index the documents, we adopt the following procedure: Each document was read and the key content-bearing words were selected and listed. There were a total of 577 different keywords (as they are called) in the list. These words were sorted into categories on the basis of their meanings. It turned out that the keywords could be sorted into 47 fairly well-defined categories. In many cases a particular keyword could belong to more than one category; consequently, there were 919 occurrences of the keywords in the 47 categories. The names of the 47

[12] Published by Science Service, Inc., 1719 N Street, N.W., Washington, D. C.

categories became the tags that constituted an index term list. These 47 index terms were then assigned to the documents by working backwards as follows: For each category we determined which keywords it contained and each document which contained the keyword in question was coordinated to the corresponding category. That is to say, given the categories, the keywords in each category, and the documents associated with each of the keywords, we then were able to determine which documents should be coordinated with each category, and thus the documents were indexed by assigning to each the names of the corresponding categories. Next, Probabilistic Indexing requires that we indicate the degree with which each tag holds for the document by assigning weights to the index terms. In order to assign the corresponding weight, each document was reread and then the indexer decided for each of the tags coordinated to each document the degree with which it held. We had decided previously that an adequate set of values for the weights was eighths, i.e., the values 1/8, $\cdots$, 8/8.

5.3. *Experimental Evaluation of the Measure of Relevance.* The problem which we now consider is: How well does our relevance number perform in ranking documents according to relevance? First, we must be careful to distinguish a user's information need N from his request formulation R. In a real library system N will never be known, only its description R in a rather artificial language, viz., a Boolean function whose variables are index terms. The library indexing system only relates documents to this request language, but we want to relate documents to information need. A bridge between request language and information need is through statistical data relating library users with the utility they derive from documents. Such statistical data is given by the a priori probability distribution. This is shown by the theoretical development which states that the probability of a document satisfying the request, i.e., the probability of a document giving the desired information item N, is proportional to the product of the a priori probability ($P(A,D_i)$) and the value ($w_i(R)$) of the extended weight function[13] for the request describing that need. In a sense then, this quantity is a measure of the degree of relevance of a document for the information need of the requestor. We say "in a sense" because of its unavoidable probabilistic nature. It is a *probabilistic estimate* of the relevance of a document for the information need of the requestor. With this qualification in mind we call this quantity "relevance to information need" or "probable relevance." We have:

$$\text{(relevance to information need)} \sim P(A,D_i) \cdot w_i(R) \tag{23}$$

We conjecture that our computational procedure for computing the values of the extended weight function is a measure of "relevance to request formulation." We will present the supporting data in subsequent sections, but in anticipation of this we state the result. The inverse inference theorem plus experimental data implies:

$$\text{(relevance to information need)} \sim P(A,D_i) \cdot \text{(relevance to request formulation)} \tag{24}$$

[13] See Appendix.

5.3.1. *The result predicted by the inverse inference theorem.* The content of the inverse inference theorem (formula (23)) can be illustrated by the following hypothetical experiment: Consider a document in the experimental library. It consists of many information items. Select one of these. Let a library user describe this item in the library request language. Let the library system now operate on the request, producing a collection of documents. (If the library indexing system is adequate and the formulation of the request is accurate, the original document from which the information item was derived should appear in this retrieved collection.) We now ask the library user to prepare a list:

L_1 : the retrived documents ranked according to relevance to the information item

We ask another person to prepare a second list:

L_2 : the retrieved documents ranked according to relevance to the request formulation

To facilitate the processing of this comparative data, we ask that the documents be classified into five categories: I. Very Relevant; II. Relevant; III. Somewhat Relevant; IV. Only Slightly Relevant; V. Irrelevant.

Suppose now we simulate an a priori probability distribution and repeat the above experiment many times, each time selecting a document (from which to obtain an information item) by using the simulated distribution. For each request we obtain lists L_1 and L_2 and third list L_3 :

L_3 : the retrieved documents ranked according to the magnitude $P(A,D_i) \cdot w_i(R)$

The inverse inference theorem now tells us what we may expect to find, namely, that the list L_3 will agree with list L_1 *in the long run.* More precisely, for each instance of a particular request R there are many information items (or needs) that would be formulated by R, one for each requestor who uses R. If for each list L_1 that originated with these requestors we computed the mean relevance evaluation for each document in L_1 by using the category numbers I, II, III, IV, V, then the resulting ranking should agree with list L_3.

5.3.2. *The experimental design.* The result predicted above is difficult to test empirically because it would require such a large sample, but an experiment designed on a much smaller scale can give us some valuable information. Since it is designed primarily to test both the basic selection process and the search strategy by elaborating the request as well as the computational schema for $w_i(R)$, a flat a priori probability distribution is assumed, i.e., all $P(A,D_i)$ are taken to be equal. The significance of this for the probable relevance concept is clear by looking at formula (23).

$$\text{(relevance to information need)} \sim w_i(R). \tag{25}$$

The interpretation of this by the phrase "in the long run" still holds, however. That is, we would not expect a single list of type L_1 to compare with its corresponding list of type L_3 (this last being the ranking of documents by the values $w_i(R)$ in the case of equal $P(A,D_i)$). This can be seen by noting that as the information need becomes more specific the evaluations in a list of type L_1 would tend to split into the two classifications of Very Relevant or Irrelevant, but the ranking by the values of $w_i(R)$ always varies gradually. On the other hand, a list of type L_2 might conceivably be expected to agree with the list L_3 in a single case. This is the content of the experimental result stated in subsection 5.3.1.

5.3.3. *Hypothesis to be tested.* We can formulate our goal as that of attempting to confirm that the value $w_i(R)$ that we compute for each document selected by a given request is, in fact, a measure of relevance with respect to the request formulation. If our basic notion is correct, it implies the following hypothesis which we call H_1.

H_1 : if a document is relevant to a request, then a high number $w_i(R)$ will be derived for it.

How to verify, confirm, test this hypothesis empirically? We did the following: A number of documents from our experimental library were selected at random and for each document a question was formulated which could be answered by reading the corresponding document. Several persons who acted as test subjects were briefed as to the nature of the library, the indexing system, etc., given a set of questions and asked to formulate a library request for information on the basis of which, hopefully, relevant documents would be retrieved (as as to answer the question). Given the library requests that these test subjects formulated, we proceeded to search and select the accession numbers of those documents satisfying the logic of the request. For each request a list of documents (i.e., a list of the corresponding accession numbers) was generated, and the documents in the lists were ranked according to the number $w_i(R)$ that was computed for each. We then examined each list to determine whether or not the so-called "answer" document was on the list, and if it was we recorded its relative position on the list. We made the (natural) assumption that the answer document (i.e., the document on the basis of which the question was formulated) would be relevant to the request. We then determined the number of times that the correct answer document retrieved was associated with a high number $w_i(R)$. The results can be summarized as follows: 40 library requests were made, and in 27 cases the answer document was retrieved. The number of documents on the output lists ranged from a minimum of one (in four cases) to a maximum of 41. In the majority of the 23 cases which contained more than a single document, the answer document appeared towards the top of the list.

The results showed that if the answer document was on a list, then it was computed to have a high number $w_i(R)$ in most of the cases. This evidence thus supports the hypothesis H_1, which asserts that if a document is relevant a high value $w_i(R)$ will be computed for it. However, it was not always the case that the answer document was computed to have the highest number; i.e., there were documents other than the answer document for which a high number was derived. Thus, the question arises: "If a document has a high number $w_i(R)$, is it relevant to the request?" This represents the converse of the original hypothesis H_1. We shall form this as an hypothesis and call it H_2.

H_2 : if a document has a high number $w_i(R)$, then it is relevant to the corresponding request.

If we can confirm H_2 as well as H_1 we will have, in fact, confirmed an hypothesis H^* which is stronger than each.

H^*: the methods of Probabilistic Indexing will derive a high number $w_i(R)$ for an arbitrary document *if* and *only if* the document in question is relevant to the request.

In order to determine if there were relevant documents other than the answer document on a list, we had to have evaluation data from the users themselves. We obtained a sample of this information from the test subjects in the following way: Four test subjects were given the actual documents corresponding to the retrieval lists, and they were asked to read each document and decide whether they considered it to be Very Relevant, Relevant, Somewhat Relevant, Only Slightly Relevant or Irrelevant. Thus for each document retrieved they would judge to which of these five categories it belonged, and we in turn compared their judgments with the numbers $w_i(R)$ which we had computed for each document. A fifth person prepared control lists, i.e., evaluations for the same requests.

In order to facilitate the comparison we standardized the values $w_i(R)$; i.e., we multiplied each value by the reciprocal of the highest value to force the numbers on each list to vary from 1 to 0,—0 being the value assigned to unretrieved documents. We also divided the numbers into three categories: *high* (value equal to or greater than 0.75, *medium* (value between 0.75 and 0.25), and *low* (value equal to or less than 0.25). The results show quite definitely that if a document had a high number $w_i(R)$ that document was judged by the evaluator as Very Relevant or Relevant, in most cases. Conversely, if the number $w_i(R)$ was low, the evaluators rated the corresponding document as either Only Slightly Relevant or Irrelevant in most cases.

Thus the data support the following: "If a document is relevant to a request, then there is a strong probability that the document will have a high number $w_i(R)$ computed for it." Furthermore, the data support the converse; viz., "If a document is computed to have a high number $w_i(R)$, there is a strong probability that it is relevant to the request." Thus the data support both H_1 and H_2, and taken jointly we see that the data do support and confirm the stronger hypothesis H^*; viz., a high number $w_i(R)$ will be derived if and only if the document in question is relevant to the request. The details of the analysis show how the values $w_i(R)$ associated with these documents were distributed among the five categories. Computing the average value and the variance in each of the five categories, we obtained the following results.

Document Rating	Mean	Variance
I. Very Relevant	0.81	0.043
II. Relevant	0.72	0.053
III. Somewhat Relevant	0.54	0.043
IV. Only Slightly Relevant	0.40	0.110
V. Irrelevant	0.18	0.013

Thus we see that the values of the numbers that we computed decrease, on the average, as we go from category I (Very Relevant) to category V (Irrelevant).

Although this result tends to confirm our hypotheses H_1, H_2, and H^*, we prefer to look deeper into the situation. Let us call categories I or II simply "Relevant" and category V, as before, "Irrelevant." Note that Relevant and Irrelevant are not negations of each other since we have the intermediate categories III and IV consisting of documents neither totally Relevant nor Irrelevant. Now the hypotheses H_1, H_2, H^* say two things:

1. Relevant is equivalent to High.
2. Irrelevant is equivalent to Low.

Also these hypotheses imply two weaker statements:

3. Relevant implies not-Low.
4. Irrelevant implies not-High.

The statistical confirmation of these statements can be accomplished by using the theory of the coefficient of association between predicates as outlined in section 2.6. That is to say, each of the statements above calls for a study of a matrix of the kind defined on p. 227, i.e., a sorting of the total class of retrieved documents according to the properties:

1. Relevant and High
2. Irrelevant and Low
3. Relevant and Low
4. Irrelevant and High

We would expect to find the Q-values in (1) and (2) to be near $+1$ (maximum positive association), and the Q-values in (3) and)4) near -1 (maximum negative association). These values are in fact:

$$\begin{aligned} Q(\text{Relevant, High}) &= +0.70 \\ Q(\text{Irrelevant, Low}) &= +0.90 \\ Q(\text{Relevant, Low}) &= -0.92 \\ Q(\text{Irrelevant, High}) &= -1.00 \end{aligned}$$

Since these values are fairly sensitive, we introduce a control on the study by assuming that these predicates are statistically independent, then computing the probabilities of the Q-values having been as close or closer to the anticipated values by chance. For the four distributions we calculate these control probabilities to be 0.041, 0.006, 0.010, 0.059, respectively.[14]

5.4. *Elaboration of the Selection Process.*

5.4.1. *Initial remarks.* The relevance number, as we have seen, provides a means of ranking documents according to their probable relevance. However, the solution to the problem of retrieval effectiveness involves more than ranking by relevance—it involves the proper selection of those documents which are to be ranked. Before we describe the results of the experiments that were conducted to test our methods for improving the selection process, let us take one more look at the relevance number as a filter to eliminate low relevance documents. In

[14] Precisely, if Relevant and High are independent, then the probability of their Q-value having the property $0.70 \leqq Q \leqq 1.00$ is 0.041 (similarly for the other classifications).

particular, let us consider the usefulness of the relevance number on unelaborated requests.

In our experiments, 40 different library requests were made and a total of 379 documents were retrieved (using the basic process of selecting those documents whose tags are logically compatible with the logic and tags of the request). Let us compare the results of probabilitsic searching and so-called "binary" or conventional searching. We can do this by assuming that all the tags which are assigned to documents with a nonzero weight are, in fact, assigned to the corresponding documents in the conventional system. Thus when the basic selection process is the same (viz., the unelaborated logical matching process), the same documents will be retrieved in both cases; however, in the conventional system the retrieved documents are not ranked by any criteria of relevance. For each of the retrieval lists if n documents have been retrieved and the answer document is present, then using the conventional search technique the requestor must read, on the average, $(n + 1)/2$ documents. If the answer document is not present, then all of the retrieved documents must be read (in order to determine that no relevant information was retrieved). These considerations (inadequate though they be, since they presuppose that only an answer document produces a satisfactory search result) give us a criterion with which to compare the probabilistic and binary searches. This criterion is the total number of documents that would have to be read for all 40 searches in order to find the answer documents. The results are as follows:

Type of Search	Total Number of Documents Retrieved	Total Number of Documents that would have to be Read
Binary	**379**	**235**
Probabilistic[15]	**379**	**181**

Thus we see that a conventional system would require the user to read approximately 30 percent more retrieved documents to obtain the same number of answer documents. These two different searches, each using the basic selection process, produced 27 answer documents out of a possible 40. (Note that the binary search as defined above is more extensive than might be expected in the sense that we have used all the tags with nonzero weights as binary tags. In an actual conventional system those tags with a low weight would probably not be coordinated with documents. That is to say, the use of weighted tags encourages more tags to be applied to a given document than would be the case if weights were not allowed. In a previous study where documents were indexed independently by two different indexers, one using probabilistic indexing, the other using binary indexing (i.e., either a tag holds for a document or it does not), it was found than 70 percent more answer documents were retrieved in the probabilistic search and only 32 percent more documents had to be read.

The above comparison presupposes that the user is looking for some specific information (viz., the answer document) and that he knows when he has found it. It might be more realistic to make no such assumption; therefore, let us consider the following comparison. Given a request for information, a probabilistic search

[15] A flat a priori probability distribution was used.

is made, but beforehand we tell the user to read only those documents which have a computed relevance number greater than 0.5. That is to say, "before the facts" we give the requestors a guide to use in reading the 40 lists presented to them. It turns out that of the 379 documents in the 40 lists there are only 225 which have a relevance number greater than 0.5. Furthermore, it turns out that if the users had adopted the strategy of reading only those retrieved documents which had relevance numbers greater than 0.5, then they would have found 25 of the 27 answer documents.[16] Now compare this with the case of conventional retrieval where the users would have to read all of the 379 retrieved documents (since there is no way to distinguish between any two documents in the same list). In this latter case the users, of course, would find all 27 answer documents, but again at the "cost" of reading all 379 documents. Thus we see that a conventional system would require that users read 68.5 percent more documents than for the probabilistic system and they would gain only 7.4 percent in increased number of answer documents.

These considerations indicate that the relevance number can be used to filter out irrelevant material. That is to say, if we use the relevance number associated with documents to separate the relevant from the irrelevant, we are providing the user with a valuable tool.

5.4.2. *Automatic elaboration.* We have described two methods for automatically elaborating upon the selection process which is involved in information searching. One method establishes a measure of distance in document space and the other method involves measures of closeness in request space. We shall not consider the former since as yet no experimental tests have been completed. For closeness in request space we have described three different statistical measures, viz., forward conditional probabilities, inverse conditional probabilities, and coefficients of association. We now raise the questions: "How good are the proposed statistical measures of closeness in elaborating upon a request?" and "Which of the three measures that have been discussed is the best?" Again, in the case of the automatically elaborated request we generate the new request R' given the initial request R by formulating the following type of disjunction for each tag in R:

$$\text{if } R = I_j, \quad \text{then} \quad R' = I_j \vee (\alpha) I_j'$$

where α is the measure of closeness between I_j and I_j', and I_j' is the term that gives maximum α with respect to I_j. We would like to be able to establish the following:

(1) That the elaborated request catches relevant documents which are not selected by the original (unelaborated) request.

(2) That, although the elaborated request catches more documents, the relevance number can be used as a guide for eliminating the ones with low probable relevance.

[16] In one of the two remaining cases the relevance number of the answer document was just under 0.5, and in the other case the answer document had a rather low number but it was third in a list of only three.

5.4.3. *Some testing (evaluation) problems.* Since we are really interested in the over-all retrieval effectiveness of the selection process, we would like to know how many of the relevant documents in the entire library have been caught by the elaborated requests. In order to determine this it would be necessary for us to present to the requestor the entire library so that he, in turn, could judge which relevant documents, if any, were not retrieved. That is to say, in order that a user properly judge whether or not he did, in fact, receive all relevant documents as the result of a search, he would have to be familiar with the entire contents of the library. Because of this difficulty, we see that such an evaluation would be impractical to conduct. We must, therefore, lower our sights and look for a substitute type of evaluation. The substitute that we have adopted consists in, again, using the answer documents as a measure of retrieval effectiveness. That is to say, since we *know* that the answer documents are relevant, we can automatically elaborate upon those original requests which did not catch the answer document in order to see whether the elaborated request succeeds in retrieving it. Such a test would allow us to establish some measure of the retrieval effectiveness of the automatic elaboration procedures. We can compare the total number of documents for the elaborated requests with what would be the case for the unelaborated request. This we have done and the results are discussed in the following section.

5.4.4. *Results and their evaluation.* Of the 40 requests that were made, the answer document was retrieved in 27 cases and it was not retrieved in 13 cases. We conducted three different types of elaborated requests for each of the 40 cases. The results are as follows:

(1) Using the method of request elaboration via forward conditional probabilities between index tags, we retrieved the correct answer document in 32 cases out of the 40.

(2) Elaborating the requests via the inverse conditional probability heuristic, we retrieved the correct document in 33 of the 40 cases.

(3) Using the coefficient of association to obtain the elaborated request, we obtained success in 33 cases of the 40.

Thus we see that the automatic elaboration of a request does, in fact, catch relevant documents that were not retrieved by the original request.

We now raise the question: "Because of the small size of the library and the large percent of the total library that is selected by the elaborated request, are the above results statistically significant?" That is to say, what is the probability of doing as well or better just by selecting at random, for each of the 13 requests for which the answer document was not originally retrieved, a sample of size equal to that given by the elaborated requests. We have made the corresponding calculations and it turns out that probability of doing as well or better by chance is less than 0.034 for both the forward and inverse conditional probability elaborations and less than 0.001 for the coefficient of association search. Thus the above results are indeed statistically significant.

Could the number of answer documents have been improved? That is, could 40 out of 40 answer documents have been retrieved? We looked at the seven cases for which the answer document was not retrieved when elaborating via the coefficient of association, and in three cases the indexing was at fault. That is to say, in three of the seven cases the answer document was poorly indexed (a fact of life that must be faced by all libraries). In one case the request formulation was very poor and no reasonable elaboration would help. In one case the answer document was caught by a different heuristic (viz., the forward conditional), and in the remaining two cases, again, the requests suffered by being poorly formulated.

Now consider the fact that, although the automatic elaboration of a request does catch relevant documents that would not otherwise have been selected, it also increases the total number of retrieved documents. (We point out at this time that of the three heuristics which we considered, the one which elaborated via the coefficient of association gave the greatest ratio of answer documents to total documents retrieved.) In order to have the advantages of an elaborated request (namely, the relevant documents that it obtains) and in order to avoid the disadvantages (namely, the larger number of total documents), we now introduce the relevance number to truncate the output lists. That is to say, we use the relevance numbers to separate out the highly relevant from the less relevant documents by adopting the following rule: Only those documents which are selected by the elaborated request and which have a standardized relevance number greater than 0.5 are to be retrieved. Our experiments with the coefficient of association heuristic show that of a total of 661 documents that were selected by the elaborated requests only 446 (or 67.5 percent) have a standardized relevance number greater[17] than 0.5. Furthermore, if we adopt this rule, then 32 out of the 33 (or 97 percent of the) answer documents which are selected by the automatic elaboration would still be retrieved; i.e., 32 of the 33 answer documents had relevance numbers greater than 0.5.

We conclude by observing that to a very large degree the procedures for automatically elaborating upon a request are empirical; i.e., their development and refinement must rest on further empirical testing and experimentation. Hopefully the results of further tests will shed light on and provide new insights into the difficult and intriguing problems of information identification, search and retrieval.

APPENDIX

Extension of the Weight Function

Before looking at the computional procedure for deriving the relevance number given any arbitrary request R, we must explain the meaning of the language of the request. We allow two logical operations between index terms, viz., "and" and "or". We abbreviate "I_1 or I_2" by "$I_1 \vee I_2$", "I_1 and I_2" by "$I_1.I_2$"; the first is called a disjunctive request, the second, a conjunctive request. The different interpretations of the logical combinations $I_1.I_2$, $I_1 \vee I_2$, as used in request

[17] For these computations we used a flat a priori probability distribution.

TABLE 1

Interpretation of Logical Connectives

Request:	$I_1.I_2$	$I_1 \vee I_2$
Logical meaning	User requests information on the "subject" designated by $I_1.I_2$	User requests information on the "subject" designated by $I_1 \vee I_2$
Retrieval instruction meaning	Search for documents indexed under I_1 and I_2	Search for documents indexed under I_1 *and* search for documents indexed under I_2
Class meaning	User obtains documents indexed under both I_1 and I_2	User obtains documents indexed under I_1 or I_2 or both

formulations are shown in table 1. Note how the "$\vee$" inside a retrieval prescription becomes an "and" in the retrieval instructions. We can say that a disjunctive request is actually several requests, but the searches are to be conducted simultaneously.

By extending the notation for a request to include logical combinations of tags, we can consider every request R (i.e., every Boolean function of index terms) as an event class. For example, $R = I_j$, $R = I_j.I_k$, $R = I_j \vee I_k$, etc. We have shown that if it is possible to compute $P(A.D_i,R)$ then we can rank documents according to probable relevance by taking the relevance number to be

$$P(A,D_i)\cdot P(A.D_i,R);$$

for, by the inverse probability calculation

$$P(A.R, D_i) = \left(\frac{1}{P(A, R)}\right)\cdot P(A, D_i)\cdot P(A.D_i, R), \qquad (1)$$

so that $P(A.R,D_i)$ is proportional to $P(A,D_i)\cdot P(A.D_i,R)$. Now we note that $P(A.D_i,R)$ is an *extension* of the modified weight function in the sense that:
If $R = I_j$, then

$$w_{ij} = w_i(R) = P(A.D_i,R). \qquad (2)$$

Thus the problem is to extend the function $w_i(I_j)$, whose values are given only for $I_1, \cdots, I_n$, to any Boolean function of these terms. We denote this extension by $w_i(R)$ and we require this extension to satisfy the rules of probability since we intend for it to be an estimate of $P(A.D_i,R)$. In particular, we require:

$$0 \leqq w_i(R) \leqq 1, \qquad (3)$$

$$w_i(I_1.I_2) \leqq w_i(I_1), \qquad (4)$$

$$w_i(I_1 \vee I_2) + w_i(I_1.I_2) = w_i(I_1) + w_i(I_2). \qquad (5)$$

We note the important fact that (5) allows us to compute the weight of a disjunction if the weight of a conjunction is known. Successive applications of (5), combined with logical transformations, allow the weight of any request to be written as sums and differences of weights of single terms or conjunctions. Thus the problem of the extension of the weight function is reduced to the extension to conjunctions. For these weights we also have certain restrictive conditions. If we let $p = w_i(I_1)$ and $q = w_i(I_2)$, then it can be shown that $w_1(I_1.I_2)$ must be less than or equal to the minimum of the two numbers p and q and must be greater than or equal to $p + q - 1$ if this is positive, otherwise it must be greater than or equal to 0. We write this conditions as

$$\max[0, p+q-1] \leqq w_i(I_1.I_2) \leqq \min[p, q]. \tag{6}$$

We have decided to take as the initial w-value of a conjunction its independence value, i.e.,

$$w_i(I_1.I_2) = w_{i1} \cdot w_{i2}. \tag{7}$$

The relevance number for a conjunction $I_1.I_2$ is then given by

$$P(A,D_i) \cdot w_{i1} \cdot w_{i2},$$

and the relevance number for a disjunction $I_1 \vee I_2$ becomes by (5)

$$P(A,D_i) \cdot [w_{i1} + w_{i2} - w_{i1} \cdot w_{i2}].$$

Several remarks need to be made about use of the independence value. Note that we do not say that the tags are independent—in fact they are not—but the word "estimate" is useful to avoid making a false assumption. First, we estimate $w_i(I_1.I_2)$ by $w_{i1} \cdot w_{i2}$. Second, we use the independence value relative to the class D_i, that is, we take

$$P(A.D_i.I_1,I_2) = P(A.D_i,I_2), \tag{8}$$

but not

$$P(A.I_1,I_2) = P(A,I_2). \tag{9}$$

We believe the former estimate is more accurate than the latter. In section 4.5 we discussed a coefficient of association between index terms. This coefficient Q lies in the interval $[-1, 1]$ with $Q = 0$ being the point of independence. The joint occurrence of two events will have a probability in excess of its independence value only if the corresponding value of Q is positive. We have two intervals to schematize this situation (p and q are the probabilities of the separate events and Q their coefficient of association):

$$\overline{0 \qquad Q \rightarrow \qquad 1}$$

Interval of Positive Association

$$p \cdot q\!\uparrow \quad \overline{\text{Probability} \rightarrow} \quad \uparrow\!\min[p, q]$$

Corresponding Interval of Probability Value

An investigation of the statistical correlation between tags via the computation of Q and then a subsequent study of which pairs of tags were used in requesting shows that Q had positive values for almost all of these pairs. This indicated that computations were called for with estimates of $w_i(I_1.I_2)$ taken at the upper end of the scale, i.e., where

$$w_i(I_1.I_2) = \min[w_{i1},w_{i2}]. \tag{10}$$

The results were not as successful as when using the independence value. A possible explanation lies in noting that independence is a three-term relation as formulas (8) and (9) show. It could well be that the probability value for tags I_1 and I_2 relative to the reference class A lies closer to the maximum value ($\min[p, q]$), while the probability value for I_1 and I_2 relative to $A.D_i$ lies closer to its independence value. In our computations we have assumed this to be the case.

Acknowledgment

The authors are indebted to Mr. L. C. Ray of Ramo-Wooldridge for his work in the planning and execution of the experiments on Probabilistic Indexing. We would also like to acknowledge the excellent computational work done by Mrs. T. Hamada.