MATHEMATICS AND INFORMATION RETRIEVAL

GERARD SALTON

TR78-332

Department of Computer Science
Cornell University
Ithaca, NY 14853

Mathematics and Information Retrieval

Gerard Salton

Department of Computer Science

Cornell University

Abstract

The development of a given discipline in science and technology often depends on the availability of theories capable of describing the processes which control the field and of modelling the interactions between these processes. The absence of an accepted theory of information retrieval has been blamed for the relative disorder and the lack of technical advances in the area.

The main mathematical approaches to information retrieval are examined in this study, including both algebraic and probabilistic models, and the difficulties which impede the formalization of information retrieval processes are described. A number of developments are covered where new theoretical understandings have directly led to the improvement of retrieval techniques and operations.

1. Information Retrieval Operations

Three different cultures currently share the information retrieval field. In data base retrieval, simple structured files are normally processed, using a small number of well-defined attributes to characterize each record, and a restricted set of prespecified query types to access the data base. Fewer restrictions exist in reference retrieval where the records represent books, documents and other library materials, and the number of different attributes available for the identification of the information items is effectively unlimited. In that case, the queries often refer to the information content of individual documents. In the most general case, a retrieval system might be designed to handle any kind of query, and the system might furnish direct replies to such queries. In such question answering, or fact retrieval systems a wide variety of different types of information identifiers may be needed, and the answers may have to be based not only on a deep analysis of each individual information item, but also on general world knowledge and other extraneous factors.

No matter what retrieval environment is actually involved, four main system components must be taken into account in any mathematical formulation of the retrieval problem:

a) first, the objects, documents, or records themselves which in the aggregate constitute the information files to be processed;

b) second, the information identifiers, terms, index terms, key words, attributes, etc., which characterize the records and represent the information content in each case;

c) third, the information requests which enter into the system and are to be compared with the stored records prior to retrieval;

d) finally, the relevance information often supplied by the users of the system connecting the information requests to the stored information items.

To describe a particular retrieval system and relate the system components to each other, certain characteristics of the system may be quantified. Among these characteristics are the number, sizes, types and joint-distributions of the term occurrences in the data base, and various costs associated with response time, record access time, and the quality of the retrieved data.

In principle, it is possible to use mathematical models in a retrieval environment for descriptive purposes only. Such an approach provides an explicit system formulation and a careful specification of the assumptions which govern the system design. In general, it seems more valuable to go beyond pure description, and to relate the mathematical treatment to actual system design parameters. Ideally this makes it possible to perform changes in the operating environment while observing the corresponding effects on system performance. Eventually optimum design parameters may be generated by such an abstract treatment of the retrieval problem.

The following processing operations must participate in any comprehensive retrieval formulation:

a) indexing, that is the assignment of terms and content identifiers to records and information requests;

b) classification, that is the specification of affinity groups between records, or record identifiers;

c) term matching, that is the comparison of analyzed information requests with stored records;

d) searching, that is the utilization of available file access paths to locate designated records;

e) retrieving, that is extracting answers in response to the information requests;

f) query and record updating through appropriate interactions between users and system;

g) evaluating systems performance in terms of user satisfaction, search effort, and retrieval cost.

Certain retrieval operations naturally lend themselves to mathematical formulations. In particular, since records and information items are often represented by sets or vectors of terms, or sometimes by higher order structures such as graphs, algebraic models can be utilized to describe indexing, searching, and retrieving. Probabilistic considerations may also apply if one assumes that system characteristics such as the term assignment to the records, or the relevance properties of the records are probabilistic in nature. Other mathematical techniques that have been used include decision theory, information theory, pattern classification, mathematical linguistics, and feature selection methods.

The first formal approaches to information retrieval problems were initiated about twenty years ago. [1-6] However, successful implementations of formal results in operational environments are quite recent. In the remainder of this study, various mathematical developments are described with emphasis on algebraic and probabilistic procedures. The improvements in retrieval techniques and operations which may result are stressed whenever possible.

2. Quantitative Considerations

Quantitative models have been used to describe a wide variety of phenomena in document processing environments, including the characteristics of existing file access mechanisms, and the criteria leading to the generation of optimum indexing vocabularies. In particular, indicators such as the size of a useful indexing vocabulary, the characteristics of the frequency distribution of the terms across the documents of a collection, and the statistical characteristics of the indexing vocabulary have been described by using algebraic and probabilistic formulations.

Typically, the quantitative aspects of certain parameters are expressed mathematically, and conclusions are drawn concerning useful operating modes or optimum system conditions. Thus, the Poisson distribution is believed to apply whenever rare events occur independently of each other at constant average rates. Since most library items circulate only rarely among library patrons, and since the circulation characteristics of the items are normally unrelated to each other, a Poisson distribution has been used to describe the number of books and the circulation characteristics of books in a library within a given period.

Of particular interest in information dissemination are phenomena whose behavior is hyperbolic in nature, that is, for which the product of fixed powers of the variables is constant. [7] In their simplest discrete formulation such laws apply to situations in which an input increasing geometrically produces a yield increasing arithmetically, the objects in question--letters of the alphabet, words in the language, authors of documents, journals in a given subject, etc.--being chosen from a finite repertoire of elements whose co-occurrences are only weakly correlated and whose quantitative properties are additive. In all these cases, all items are presumably equally open to selection, but a success-breeds-success mechanism appears to operate in the sense that when an item has been successfully selected, its chances of being selected again increase. Thus, certain letters of the alphabet tend to occur much more frequently than some others; so do certain words in written text.

In information processing, the quantitative yield of the technical journal articles written by a specified author provides a case in point: the appearance of a journal article by a certain author makes it more likely that additional articles will be written by the same author. The class of events obeying such success-breeds-success mechanisms have been characterized mathematically by the so-called cumulative advantage distribution, which is derived by rewarding success by increasing the chance of further success, while keeping the chance of failure constant when failure occurs (lack of publication is a nonevent instead of being counted against the particular author). [8]

If the items under consideration are arranged in decreasing order of yield--for example, authors of journal articles in decreasing order of the number of articles published, or words in a long running text in decreasing order of their frequency of occurrence--the total yield from the most productive fraction x of the items is expressed as

$$F(x) = A \log(1+Bx) \;, \qquad (1)$$

while the yield of the item with rank x is

$$F(x) = C/1+Bx \qquad (2)$$

where A, B, and C are suitable constants.

The latter formula is a form of the well-known law by Zipf which states that the frequency of an item (letters of the alphabet, words in running text, etc.) is inversely related to the rank of the item in decreasing frequency order. [9] Zipf-type frequency distributions are often assumed to characterize a variety of phenomena of interest in retrieval system design and simulation, for example, the occurrence probabilities of the index terms assigned to the documents of a collection. [10,11]

Another quantitative indicator in document processing is the log-normal distribution where the logarithm of the dependent variable is normally distributed about the mean value of the independent variable. Such a distribution is believed to characterize, for example, the number of documents indexed by term sets of a certain size, a fact that might be utilized in deciding on an optimum size for the indexing vocabulary used in a given document environment. [12,13] Studies of many document accessing devices, such as directories, indexes, catalogs, titles, and so on, also indicate that in each case the size distributions (in terms of the number of words in titles, the number of entries in a back-of-the-book index, etc.) are log-normal of the form

$$f(x) = \frac{1}{xs\sqrt{2\pi}} \; \exp(-1/2\left[\frac{\log x-m}{s}\right]^2\} \qquad (3)$$

where m is the mean and s the standard deviation of the distribution. [14,15]

Log-normal distributions can be transformed into a form a Zipf's law by letting m and s approach infinity in such a way that m/s^2 approaches a single parameter k. This may explain the fact that both types of distributions appear useful in the same general context.

Various theoretical justifications have been advanced for the regularities expressed by the several quantitative laws in the document processing field. Appeal is made in particular to file search theories and to information theoretical considerations relating to optimum communication procedures across information channels, and to the information content of the indexing vocabulary. [16,17] The practical usefulness of the quantitative distributions is in any case clear when it comes to designing indexing and dictionary processing systems, and the size and distribution of these mechanisms is known in advance.

Furthermore, the memory space requirements needed to implement various search and accessing procedures can also be determined more easily when the corresponding quantitative parameters obey certain definite rules. In general, however, it remains to be seen whether the quantitative models will ever take on more than peripheral importance in the study of information retrieval operations.

3. Set Theoretic Models

The set theoretical view of information retrieval is based on the recognition that information requests are normally formulated by choosing collections or sets of item identifiers, or keywords. The keyword sets in turn lead to the retrieval of record subsets chosen from among the stored collection of records. The fundamental data of retrieval theory are provided in this view by the relations which exist between the set of item descriptions and the corresponding record sets.

In the simplest case, one could start with a query set R and a set of stored records D. The retrieval operations may then be expressed as a mapping $T:R\rightarrow 2^D$ which assigns to each query $r\epsilon R$ some element of 2^D, the set of all subsets in the record space. This is illustrated in the schema of Fig.1(a). [18] The main task then consists in studying the properties of the mapping between descriptor and record subsets. Since the concept of record relevance is absent from the model, an attempt is made to relate the abstract model to search efficiency rather than to retrieval effectiveness. Of particular interest in this connection are theories of file organization and access speed considerations.

The mapping between queries and records may be studied by noting that an order relation is normally definable in the query and record subspaces. The normal set inclusion relations between subsets of records provides the order in the record space D; furthermore, a partially ordered relation ($\geq$) is automatically also defined in the query space R when each request is formulated by choosing subsets from among the set of query identifiers. In these circumstances, it is easy to show that for queries $r,s\epsilon R$

$$r \geq s \Rightarrow T(r) \subseteq T(s) \ ,$$

that is, the more comprehensive the query, the smaller the record set actually retrieved. [18] In the limit, if nothing is specified, that is if the set of records to be obtained is left completely unrestricted, one retrieves everything; on the contrary, when every possible topic area is specified in the query, nothing may be expected to be retrieved.

In most retrieval systems, the physical records are not the direct images of the retrieval function, but the system operates on some description of the records, for example keyword sets. Furthermore the set of record descriptions may not be identical with the set of all possible queries. An indexing function X may then be defined from the record space D into the space C of all possible record descriptions which assigns to each element $d \epsilon D$ a value X(d) called the description of d. The mapping $X:D \rightarrow C$ defines an equivalence relation on D, where a given element $d_i \epsilon D$ is equivalent to d_j if and only if $X(d_i) = X(d_j)$. In such a system, the set of documents T(r) retrieved in response to some query $r \epsilon R$ is then a union of equivalence classes in D under the equivalence relation defined by X.

Specifically, the complete retrieval system may now be taken as a query language R, a record set D, and an indexing language C, together with the functions $X:D \rightarrow C$, and $F:R \rightarrow 2^C$. The retrieval function $T:R \rightarrow 2^D$ is now defined as

$$T(r) = \{d:X(d) \epsilon F(r)\} .$$

Thus F maps queries into sets of possibly relevant record descriptions, and T so defined retrieves all items corresponding to one of these assigned descriptions. The model is represented in Fig. 1(b). [18]

A substantial amount of effort has been devoted to the study of mapping systems of the kind represented in Fig. 1. In particular, various indexing techniques can be simulated, by using for example an equivalence relation within the descriptor set C to represent term grouping or thesaurs functions. Descriptor phrase generation methods can be similarly simulated in the abstract model. [19-21]

The set theoretical models can be related to questions of retrieval efficiency by considering different types of storage organizations for the record set D. Two main problems arise in practice: first one would like to obtain rapid access to the set of records corresponding to individual user queries; second one would give preference to a storage organization in which the basic retrieval methodology is not grossly affected when small perturbations are introduced in the make-up of the record and query sets. Both of these requirements translate themselves into a file organization in which related record sets are stored in close proximity to each other.

A typical investigation in this connection concerns the so-called consecutive retrieval property which obtains whenever all records pertinent to a certain set of information queries are stored in consecutive, or adjacent storage positions. [22,23] Records with that property can normally be retrieved in a single file access in a linear storage device such as a drum or a tape. It is easy to show that when the records pertinent to a given query set form a nested set ($\rho(Q_i) \subseteq \rho(Q_{i+1})$, $i = 1,2,\ldots,n$, where $\rho(Q_i)$ is the set of records pertinent to query Q_i), the consecutive retrieval property necessarily obtains; the same is true for any set of two arbitrary queries, but not unfortunately for an arbitrary set of three or more queries.

While the set mapping model produces many interesting formulations and research problems, practical results are not easy to come by, and the aim of relating the mathematical work to actual system design parameters may be difficult to fulfill.

4. Retrieval as Vector Matching Operations

A) Attribute and Record Space

Consider a collection D of n records, and a set A of t attributes or properties used to identify the records. A particular record D_i can then be represented by an attribute vector

$$D_i = (a_{i1}, a_{i2}, \ldots, a_{it}) \qquad (4)$$

where a_{ij} represents the weight, or degree of importance of attribute A_j in D_i. A complete collection of records is characterized in such a formulation by an attribute-record matrix C of dimension n by t as shown in Fig. 2(a). Each row of the matrix identifies a record D_i and each column corresponds to the assignment of a particular attribute to the records of the collection. In practice, the term vectors are all sparse because most attributes will be absent from each particular record; when an attribute is absent, the corresponding attribute weight is assumed to be zero.†

Consider now a standard retrieval operation consisting of a comparison between the attribute vectors identifying queries and stored information records, respectively. If a query is represented by a t-dimensional property vector

$$Q = (q_1, q_2, \ldots, q_t)$$

where q_i is the weight of the ith query attribute, a similarity measure r_i is computable between query Q and record D_i as

$$r(Q, D_i) = \sum_{j=1}^{t} a_{ij} q_j \ . \qquad (5)$$

†The vector representation can be applied to business-type records where a given attribute (for example, the age of a given person, or the salary of an employee) can take on a variety of values. In that case, the properties then represent the individual attribute-values, and a given a_{ij} is the weight, or degree of importance of the j^{th} attribute-value in vector i. For each attribute, all values except one are then assigned a zero weight.

When the property vectors representing the queries and records are binary, that is, when the attribute weights are restricted to 0 and 1, expression (5) measures the number of matching attributes between query Q and record D_i. Other well-known vector similarity measures are the cosine function

$$r_i'\ (Q,D_i) = \frac{\sum_{j=1}^{t} a_{ij}q_j}{\sqrt{\sum_{j=1}^{t} q_j^2}\sqrt{\sum_{j=1}^{t} a_{ij}^2}} \qquad (6)$$

or the correlation function

$$r_i''\ (Q,D_i) = \frac{\sum_{j=1}^{t} a_{ij}q_j}{\sum_{j=1}^{t} q_j^2 + \sum_{j=1}^{t} a_{ij}^2 - \sum_{j=1}^{t} q_j a_{ij}} \qquad (7)$$

For retrieval purposes it suffices to compute one of these similarity coefficients between queries and records, and to withdraw from the file those records which exhibit sufficiently high similarity with the given queries.

It should be pointed out that the use of similarity measures such as those shown in equations (5) to (7) is based on the premise that the underlying basis vectors of which the attribute vectors are linear combinations, are orthogonal. In actual fact, however, this may not be true because relationships may exist between individual vector attributes. In particular, from the attribute-record matrix C it is possible to compute an attribute similarity matrix T of dimension t by t, by pair-wise comparison of the columns of C. The ij^th matrix element t_{ij} of T then represents the similarity coefficient between the i^th and j^th attributes, expressed by similarities in the attribute assignments to the records of the collection.

In an analogous manner, record similarities can be obtained by similarity computations between pairs of rows of matrix C. This produces an n by n similarity matrix V shown in Fig. 2, where V_{ij} represents the similarity coefficient between records D_i and D_j;

An extended retrieval system is now envisaged in which attribute and record similarities are taken into account by making use of matrices T and V respectively. [18] Consider first the standard retrieval system of equation (5). In vector notation this becomes

$$r = Cq \quad . \tag{8}$$

An alternative retrieval function may be written as

$$r = VCTq \quad . \tag{9}$$

Here the t-dimensional query vector is first premultiplied by matrix T to produce a new query vector q' which takes into account the attribute similarities contained in T (q' = Tq). The altered query vector q' is then further processed with the attribute record matrix C, producing an initial n-dimensional response vector r (r = CTq). The latter finally is improved by incorporating the record similarities of matrix V thereby changing the record coefficients in the response vector r. [18]

The similarity matrices T and V reflect first order similarities between pairs of attributes and pairs of records respectively. Higher order similarities between triples, quadruples, etc., of items can also in principle be utilized, although the practical usefulness may be expected to diminish rapidly as the similarity order increases.

B) Indexing Theory

Of all the basic information processing operations carried out in a retrieval environment, the term assignment, or indexing operations is most crucial, since the retrieval results and all subsequent processes are directly based on the attributes identifying the information items. The generation of a viable theory of indexing is then of considerable importance in the formalization of the retrieval process. [24-26] The vector processing model which characterizes each record by a t-dimensional attribute vector may be directly translated into a useful indexing theory. [27-28]

Consider a t-dimensional attribute space. Since each record is assigned a unique position in the space based on the attributes that are present in each case, the indexing problem can be translated into a question of space configuration by asking what type of record space leads to the best retrieval results. The first thought is to construct a clustered record space which collects in adjacent positions those records jointly identified as relevant to individual user queries. A typical example of a clustered object space is given in Fig. 3 (a) showing the envelope of the t-dimensional space and the relative positions of the records identified by x's. The distance between two x's in Fig. 3 (a) is inversely proportional to the similarity between the corresponding vectors; hence vectors appearing in close proximity to each other in the space may be expected to be jointly retrievable in response to certain queries.

While the space representation of Fig. 3(a) may lead to ideal retrieval conditions, the space based on relevance clusters is difficult to construct a priori because the set of records jointly relevant to the queries is unknown at indexing time. The next best space configuration may then be a separated record space in which each record exhibits the widest possible separation from all its neighbors. Such a separated record space is shown in Fig. 3(b). The notion of a separated record space may be explained by reference to the parameters used to measure retrieval effectiveness. A dual objective is normally pursued in retrieval: on the one hand one wants to achieve reasonably high recall by retrieving a substantial portion of relevant materials; at the same time one would like to reject a large proportion of the extraneous items thereby obtaining also a high precision.[†]

When adequate separation is achieved between the records in the record space, it is possible in principle to retrieve a given item in response to a query without at the same time retrieving its immediate neighbors. The clustered space of Fig. 3(a) may then be recall-oriented since it favors the retrieval of clusters of adjacent records that may be either relevant or nonrelevant; the separated record space of Fig. 3(b) on the other hand favors search precision since it leads to the rejection of extraneous items that may be intermingled with the relevant.

The notion of record space separation produces an indexing model known as the discrimination value theory. Specifically, the value of an index term is assumed to depend on its ability to effect separation in the space when assigned to the records of a collection. Thus a useful term (a good discriminator) is one which spreads out the space when assigned to the records, as illustrated in Fig. 4. An indifferent discriminator leaves the space density unchanged, whereas a poor discriminator compresses the space. [27,28,29]

[†] Recall is the proportion of relevant items retrieved, while precision is the proportion of retrieved items that are relevant. Normally, most relevant items should be retrieved, while most nonrelevant should be rejected, leading to high recall, as well as high precision.

If the density of the record space is computed as the sum of the similarity coefficients between all pairs of distinct records (or alternatively, as the sum of the similarity coefficients between each record and a calculated centroid record located at the center of space), a term discrimination value can be computed for each term k as the difference in space densities obtained first with term k present in the record vectors and then with term k removed.

In particular, consider a set of n record vectors of the type shown in equation (4). A centroid K of the record space can be computed as the average record in such a way that k_i the ith vector element of K is

$$k_i = \frac{1}{n} \sum_{j=1}^{n} a_{ij} \quad . \tag{10}$$

The space density Y is now computed as

$$Y = \sum_{j=1}^{n} r(K,D_j) \tag{11}$$

where r is the standard similarity coefficient between records D_j and centroid K. When $0 \leq r \leq 1$, then $0 \leq Y \leq n$. If Y_k represents the space density Y with term k removed from all record vectors, the discrimination value DV_k of term k is defined simply as $Y_k - Y$. Obviously, for good discriminators $Y_k - Y$ is positive, because removal of term k will cause the space to become more dense; hence $Y_k > Y$. For poor discriminators the reverse obtains.

Term discrimination value measurements can be used as an indexing aid by choosing as record identifiers terms which exhibit sufficiently high discrimination values. Furthermore the discrimination values can be incorporated into a term weighting function by defining

$$w_{ij} = a_{ij} \cdot DV_j \tag{12}$$

where w_{ij} is the weight of term A_j in record D_i, a_{ij} is the old, record-dependent factor--for example, the frequency in occurrence of A_j in D_i--and

DV_j is the collection-dependent discrimination value.

Experimental output is shown in Table 1 for three sample collections of research documents in the areas of aerodynamics (CRAN), medicine (MED), and newspaper articles in world affairs (TIME). In each case the search precision is given at certain specified levels of the recall, averaged over 24 user queries for each collection. Collection sizes vary from 424 documents for aerodynamics to 450 items for medicine. The improvement obtainable with the DV factor is obvious in each case. [28,30]

C) Retrieval with Fuzzy Sets

In the vector theoretic model of retrieval weights are assigned to the vector attributes and similarity coefficients are computed between attributes or between records. Obviously the term weights represent the degree to which the attributes may be pertinent to the information items. Degrees of relationship can also be recognized between different attributes and different records, degrees of relevance between queries and records, and degrees of vector similarity between queries and information items. This suggests that the set theoretic model may be supplemented by fuzzy set considerations. [31-34]

A fuzzy set may be regarded as an extension of a conventional set with the added provision that each element of the set carries a parameter specifying the degree of membership of the element in the set. Specifically, if $U = \{u\}$ is a collection of objects, then a fuzzy subset W of U is a set of ordered pairs $\{(u,\mu_W(u)\}$, $u\varepsilon U$, where $\mu_W(u)$ represents the grade of membership of u in W, and μ_W is the membership function.

Using a set membership function, it is now possible to define the retrieval system components in fuzzy set terms. [34] In particular let t represent a term or attribute, and let x be a record or an information request, then a fuzzy relation F can be defined of the form

$$F = \{<x,t,\mu_F(x,t)> |x\varepsilon DuQ, t\varepsilon A\} \qquad (13)$$

where $\mu_F(x,t)$ is a function determining for each pair <x,t> a real number from the interval [0,1] representing the degree of importance, or weight, of term t in the attribute vector of x. The normal term vector for a given record or query vector $x\varepsilon DuQ$ is now defined as a fuzzy set F_x, where

$$F_x = \{<t,\mu_F(x,t)> |t\varepsilon A\} \quad . \qquad (14)$$

Additional fuzzy relations are easily introduced giving rise to new fuzzy sets. Thus similarities between records and queries can be modelled by using a fuzzy association relation G between pairs of fuzzy sets F_x and F_y, specifying the degree of membership of F_x and F_y in G:

$$G = \{<F_x,F_y,\mu_G(F_x,F_y) > x,y\varepsilon DuQ\} \quad . \qquad (15)$$

As before, μ_G is a membership function which for each pair F_x,F_y assigns a real number $\mu_G(F_x,F_y)\varepsilon[0,1]$ representing the degree of membership of the pair in G.

If a fuzzy relevance relation R is defined to express the degree of relevance of some record $d\varepsilon D$ with respect to an information query $q\varepsilon Q$, that is

$$R = \{<F_q,F_d,\mu_R(F_q,F_d)\} \quad , \qquad (16)$$

the ideal response of a retrieval system is a set of items $d\varepsilon D$ for which the membership degree of ordered pairs $<F_q,F_d>$ in the relevance relation R exceeds some threshold λ. [34]

It is not difficult to see that other retrieval operations fit comfortably under the fuzzy set umbrella. Fuzzy set models have become popular in recent years, but fundamentally new insights have not so far materialized by using this approach.

5. Probabilistic Retrieval Models

A) Retrieval as a Decision Theoretic Model

In the vector processing models of retrieval similarities are computed between attributes and records, and between records and queries. It is obvious that the degree of similarity between the various entities of interest in retrieval can also be expressed probabilistically; indeed probabilistic retrieval models have received extensive use in the past. [35,36]

From a decision-theoretic viewpoint, the basic retrieval task may be expressed in terms of three main parameters: P(rel), the probability of relevance of a record; a_1, a loss parameter associated with the retrieval of a nonrelevant or extraneous record; and a_2, a loss associated with the nonretrieval of a relevant record. A loss minimizing rule can be devised by noting that the retrieval of an extraneous item causes a loss of $[1-P(rel)]a_1$, whereas the rejection of a relevant item produces a loss of $P(rel)a_2$. In these circumstances the total loss is minimized by opting for retrieval of an item whenever

$$P(rel)a_2 \geq [1-P(rel)]a_1 \ . \qquad (17)$$

Equivalently a discriminant function g may be defined, and an item may be retrieved whenever $g \geq 0$, [10,37,38]

where
$$g = \frac{P(rel)}{1-P(rel)} - \frac{a_1}{a_2} \ . \qquad (18)$$

A retrieval rule of the kind produced by equation (18) is not useful in practice because the relevance properties of the individual records cannot of course be divorced from other system parameters. Thus, it becomes necessary to relate the discriminant function to other design parameters, and most notably to the indexing process. This can be done by defining two conditional probability parameters:

$P(x_i|w_1)$ the probability of term x_i occurring in a record given that the record is relevant to a given query

and

$P(x_i|w_2)$ the probability of term x_i given that the record is not relevant to the query. [38,39]

Using Bayes' formula, a retrieval function $P(w_i|\underline{x})$ can be obtained, where w_1 and w_2 indicate relevance and nonrelevance of the record, and $\underline{x}$ is a vector of one or more terms x_i:

$$P(w_i|\underline{x}) = \frac{P(\underline{x}|w_i)\ P(w_i)}{P(\underline{x})} \quad . \qquad i = 1,2 \tag{19}$$

Here $P(w_i)$ is the a priori probability of relevance or nonrelevance of an item, and

$$P(\underline{x}) = \sum_{i=1}^{2} P(\underline{x}|w_i)\ P(w_i) \quad . \tag{20}$$

If one assumes that the two loss parameters are equal to 1 ($a_1 = a_2 = 1$), the obvious retrieval rule now calls for retrieval whenever

$$P(w_1|\underline{x}) \geq P(w_2|\underline{x}) \quad , \tag{21}$$

or whenever the discriminant function $g \geq 1$ where

$$g(\underline{x}) = \frac{P(w_1|x)}{P(w_2|\underline{x})} = \frac{P(\underline{x}|w_1)\ P(w_1)}{P(\underline{x}|w_2)\ P(w_2)} \quad . \tag{22}$$

The discriminant function may also be linearized by taking logarithms as follows

$$g(\underline{x}) = \log \frac{P(\underline{x}|w_1)}{P(\underline{x}|w_2)} + \log \frac{P(w_1)}{P(w_2)} \quad . \qquad (23)$$

The foregoing rule relates the retrieval of the records to the occurrence characteristics of the terms in both the relevant and the nonrelevant items. For pratical application, it is necessary to specify how the probabilities $P(\underline{x}|w_i)$ are to be determined. The problem is two-fold in that first one must determine the occurrence characteristics for each term separately, and next the interactions between terms must be specified. In most abstract retrieval models the second problem is settled either by considering single-term queries only, where term interactions are obviously of no consequence [38,40], or more drastically by disregarding term interactions altogether, and assuming that terms occur independently of each other in the records of the collection. The first question relating to the term occurrences can be handled either by using a simple probability distribution, such as the Poisson distribution to characterize the occurrence characteristics of the terms, or by studying the actual occurrences of the terms in a typical sample record collection and applying the findings to other collections at large.

Consider the case where term independence is assumed and where the occurrence characteristics are obtained from a sample collection. In such circumstances one can write

$$P(\underline{x}|w_i) = P(x_1|w_i)\ P(x_2|w_i)\ \ldots.\ P(x_n|w_i) \quad . \qquad (24)$$

Assuming for convenience that the information vectors are binary, that is $x_i=1$ whenever the ith term is present in a vector, and $x_i=0$ otherwise, equation (24) becomes

$$P(\underline{x}|w_1) = \prod_{i=1}^{n} p_i^{x_i} (1-p_i)^{1-x_i}$$

and

$$P(\underline{x}|w_2) = \prod_{i=1}^{n} q_i^{x_i} (1-q_i)^{1-x_i}$$

where

$$p_i = P(x_i=1|w_1) \text{ and } q_i = P(x_i=1|w_2) \ . \qquad (25)$$

The discriminant function g of equation (23) is now rewritten

$$g(x) = \sum_{i=1}^{n} \{x_i \log\frac{p_i}{q_i} + (1-x_i) \log (\frac{1-p_i}{1-q_i})\} + \log \frac{P(w_1)}{P(w_2)} \ . \qquad (26)$$

It remains to determine the occurrence probabilities of each term x_i separately in both the relevant and the nonrelevant items in a collection. Consider for this purpose a sample collection of N records and assume that R records out of N are relevant to a given query Q and N-R items are nonrelevant. The term occurrence characteristics for a term x_i are listed in Table 2.

If one assumes that the term occurrences in the sample record collection of Table 2 are typical of the term occurrences at large, one can postulate that

$$p_i = r_i/R \text{ and } q_i = n_i-r_i/N-R \ ,$$

so that the retrieval function $g(\underline{x})$ of equation (26) becomes

$$g(\underline{x}) = \log \frac{P(w_1)}{P(w_2)} + \sum_{i=1}^{n} \log \frac{1-p_i}{1-q_i} + \sum_{i=1}^{n} x_i \log\frac{r_i/R-r_i}{n_i-r_i/N-n_i-R+r_i} . \qquad (27)$$

The first two terms in (27) are constant for a given query. Only the last factor involves term x_i. In particular, for each query term x_i a weight proportional to

$$L_i = \frac{r_i/R-r_i}{n_i-r_i/N-n_i-R-r_i} \tag{28}$$

is added to the discriminant function g. The expression L_i represents the proportion of the relevant items in which x_i occurs divided by the proportion of nonrelevant items in which the term occurs; L_i may be called the term relevance of term x_i. Since L_i expresses in some sense the value of term x_i in the retrieval environment, the term relevance may be usable in interactive retrieval to adjust the term weights as explained later.

B) Poisson Model

The earlier development shows that adequate information must be available concerning the occurrence probabilities of the terms in the records of a collection if probabilistic models are to be used in retrieval. When actual term occurrence data of the kind shown in Table 2 for simple records and queries are available, the problem can be circumvented. Unfortunately the complete retrieval and relevance information necessary for the construction of a full contingency table is generally not known a priori. It becomes necessary then to use information derived from theoretical probability distributions while assuming that the actual data in fact follow the theoretical model.

The Poisson distribution has often been used for this purpose because that distribution is easy to treat mathematically, and more importantly because its properties appear to reflect accurately the occurrence characteristics of many words or terms in written documents and records. The Poisson distribution in fact reflects a random scattering of a word, or term, throughout a running text, with text units of equal length having equal probability of containing an occurrence of the word.

Specifically, if p occurrences or tokens of a term are scattered over records or documents of approximately equal length, the probability P(k) that a given document receives k tokens of that term is

$$P(k) = \frac{1}{k!} \left(\frac{p}{n}\right)^k e^{-\frac{p}{n}} \quad . \qquad (29)$$

Since $p/n = \lambda$ represents both the mean and the variance of the distribution, the total frequency of occurrence p of a term is proportional to the variance. This fact has been used in many of the early automatic indexing models by noting that specialty words capable of representing information content tend to be clustered in a few documents, whereas nonspecialty words exhibit more uniform occurrence characteristics. By computing a measure of deviation from the Poisson model, it is then possible to separate the content words which provide effective indexing elements from others that do not. [40-45]

The Poisson model can be extended by assuming that the distribution reflects the occurrence data not only for the common nonspecialty words, but also for content and specialty terms. In the latter case, however, a distinction must be made between various classes of documents and records. If only two relevance classes exist, that is, if a given document is either relevant or nonrelevant, one may assume that two different Poisson distributions characterize the occurrences of the specialty words, the first one pertaining to the relevant items, and the other to the nonrelevant ones. [46] If π and $(1-\pi)$ represent the probabilities of a record belonging to classes 1 and 2 respectively, and λ_1 and λ_2, are the means of the respective Poisson distributions the probability that a document receives k tokens of the given specialty word is then

$$P(k) = \pi \frac{e^{-\lambda_1} \lambda_1^{k}}{k!} + (1-\pi) \frac{e^{-\lambda_2} \lambda_2^{k}}{k!} \quad . \qquad (30)$$

Under similar assumptions, the decision function of equation (22) may be rewritten as

$$g(x) = e^{-(\lambda_1-\lambda_2)} \quad \frac{\lambda_1}{\lambda_2}^{k} \quad \frac{\pi}{1-\pi} \quad . \qquad (31)$$

The term independence problem is ignored by replacing the vector $\underline{x}$ of equation (22) by a single term x assumed to exhibit k occurrences in the document or record.

Although the two-Poisson model operates imperfectly, very likely because the separation of the stored records into only two homogeneous classes may be unrealistic in practice [47], similar considerations have led to the generation of various retrieval evaluation models. [48-49] Once again two distinguished populations of objects are recognized, termed A and B, and identified in an information retrieval environment with the nonrelevant and the relevant records, respectively, with respect to some information query. A measurable characteristic Z is then chosen for the two populations, for example the query-record similarity coefficient between each member of A and B and the given query. An appropriate indication of system effectiveness may then be provided by measuring the differences between the respective probability density functions f(Z) for the two distinguished populations.

A typical distribution for the similarity coefficients pertaining to the relevant items ($f(Z|B)$) and the nonrelevant ones ($f(Z|A)$) respectively is shown in Fig. 5. If the means and variances for the two popoulations are λ_1 and λ_2, and σ_1^2 and σ_2^2 respectively, an evaluation measure can be generated based on the differences between these parameters. For example, a cut-off value $Z=C$ may be chosen as shown in Fig. 5 and the proportion of objects B for which $Z \geq C$ may then be computed as the area under the B curve to the right of cut C, and similarly for the A curve. The former measure may be interpreted as the probability of retrieval of a relevant item, previously defined as the recall; the latter is the probability of retrieval of a nonrelevant item and is known as the fallout. Several other retrieval evaluation procedures make use of differences in measurements between relevant and nonrelevant items, respectively. [50,51]

C) Term Independence

The probabilistic models described earlier involve estimates of the occurrence probabilities of query term combinations $\underline{x}$ in the relevant and nonrelevant document populations. Solutions have been provided for two special cases, namely when a single query term x is present rather than a multiplicatiy of terms, and when the various query terms occur independently of each other in the stored collection. It may be of interest to inquire to what extent the term independence assumption violates the actual occurrence characteristics of terms (words and phrases) in natural language texts.

An accepted measure of the degree of statistical dependence between random variables x and y is the correlation coefficient

$$\rho_{x,y} = \frac{E[(x-\overline{x})(y-\overline{y})]}{\sqrt{E[(x-\overline{x})^2]}\sqrt{E[(y-\overline{y})^2]}} \qquad -1\leq\rho\leq 1 \qquad (32)$$

where E is the expected value, and $\overline{x}$ and $\overline{y}$ are average values for x and y respectively. A correlation ρ close to 1 indicates linear dependence between the variables; ρ near -1 indicates perfect negative correlation, while correlation values near 0 are assumed to reflect statistical independence.

In a collection of written documents, the variables x and y of equation (32) may be interpreted as the occurrence frequencies of two terms across the documents of a collection, and $\overline{x}$ and $\overline{y}$ as the average frequencies in each case. For n documents, the correlation coefficient then becomes

$$\rho_{x,y} = \frac{\sum_{i=1}^{n} (x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i-\overline{x})^2}\sqrt{\sum_{i=1}^{n} (y_i-\overline{y})^2}} \qquad (33)$$

where x_i and y_i are the occurrence frequencies of x and y in the ith document. Experimental output containing percentages of the pair-wise correlation within certain ranges of the correlation coefficient $\rho_{x,y}$ is shown in Table 3 for 2,590 terms (or about 3.4 million term pairs) occurring in 3,469 titles of research articles in computer science, and for 2,651 terms (or about 3.5 million pairs) occurring in 424 document abstracts in aerodynamics. [52]

Column 2 of Table 3 shows that almost all correlation coefficients are near 0 for the title words in computer science; only a small fraction of one percent of the pairs exhibits any kind of positive correlation. For word pairs chosen from document abstracts, about 3 percent of the pairs are positively correlated, although the value of the correlation coefficient is generally small (below 0.2). The last column of Table 3 again shows the correlation coefficient for the 2,651 terms in aerodynamics; however in that case a clustered collection is used in which documents are grouped into classes or clusters according to similarities in their term vectors, and the term pairs chosen are those that co-occur in any of the document clusters. Since the document classes are constructed by using affinities between the items, the restricted environment within which the pair-wise correlation coefficients are computed in that case is known in advance to contain substantial term similarities. The output of Table 3 shows that in that case about 30 percent of the terms exhibit nontrivial positive correlations. In fact about ten percent of term pairs, consisting mostly of terms occurring only once in the same document of a given cluster, exhibit a perfect correlation of 1.

The data of Table 3 indicate that if a large enough environment is chosen -- for example, a complete document collection -- the deviations from zero correlation are minor. However, when the environment is restricted and consists of items with obvious similarities, the independence assumption is not tenable.

Various retrieval system models have been proposed for which at least some dependencies between terms are taken into account. Thus, the vocabulary may be divided into thesaures classes of similar terms where perfect dependence exists within each thesaurus class, but independence is assumed between classes. [10] Alternatively, the pair-wise term similarities may be assumed to be describable by specified probability distributions. [11]

In one recent model, the extent to which two terms x_i and x_j deviate from independence is measured by the expected mutual information measure $I(x_i,x_j)$:

$$I(x_i,y_j) = \sum_{x_i,x_j} P(x_i,x_j) \log \frac{P(x_i,x_j)}{P(x_i)P(x_j)} \quad ; \qquad (34)$$

where $P(x_i,x_j)$ is the joint probability of occurrence of x_i and x_j. [39] A graph is then constructed in which the nodes represent the terms, and the edge weights represent the dependencies $I(x_i,x_j)$. Since for t terms there may be $t^2/2$ dependent pairs, it becomes important to concentrate on the most important pairs by eliminating less important edges from the dependency graph. This can be done by constructing the maximum spanning tree which spans the graph by covering each node while maximizing the sum of the edge weights $\sum_{i=1}^{n} I(x_i,x_{j(i)})$. A retrieval decision function g(x) is then produced analogous to the one shown in equation (23); the function is, however nonlinear in that case, and requires knowledge of the pair-wise occurrence probabilities. [39]

It is likely that experimental output will become available soon for retrieval models incorporating at least some term dependencies.

6. Interactive Searching

Many of the existing, successful retrieval operations are currently implemented in an interactive mode permitting the user, or a search intermediary, to submit the information requests using a console device with direct access to the stored information files. [53] In such an environment it is possible to conduct a search iteratively by modifying the initial search statements until a final, satisfactory search output is eventually obtained. Various strategies suggest themselves for the reformulation and improvement of the original search requests. [54] Vocabulary displays can be used to add new terms related to the ones already present in the search request. Alternatively, it is possible to use information obtained from documents retrieved in an earlier search iteration to improve subsequent query statements.

One particularly effective interactive retrieval methods is known as the relevance feedback process. It consists in returning to the system information concerning the relevance, or nonrelevance of previously retrieved items. [55] The query can then be automatically modified by rendering it more similar to previously retrieved items identified as relevant (for example, by increasing the weights of query terms also found in the relevant items), while simultaneously rendering the query less similar to the retrieved items identified as nonrelevant.

Since relevance assessments of records with respect to queries are thus made available for at least some query-document pairs, the question arises about the proper use of this information for retrieval purposes. Consider again the contingency listing of Table 2 which exhibits the breakdown of relevant and nonrelevant records that do, or do not, contain a given query term x_i. The information of Table 2 can be used as a basis for various kinds of term weighting functions: [56]

a) If only the presence of the terms is used ($x_i=1$) and the terms are assumed to occur independently in the set of relevant items, and independently also in the whole collection, an appropriate expression of term value is the proportion of relevant items in which the term occurs civided by the proportion of the whole collection in which the term occurs:

$$f_1 = \frac{r_i}{R} \div \frac{n_i}{N} \quad . \qquad (35)$$

b) If only term presence is used as before, but the independence assumption extends separately to the relevant and the nonrelevant items, the corresponding term weighing obtained from Table 2 is

$$f_2 = \frac{r_i}{R} \div \frac{n_i - r_i}{N-R} \quad . \tag{36}$$

c) Using the term independence assumptions of case a) but assuming that both the presence as well as the absence of query terms is important, one obtains as a function of term importance the proportion of relevant items in which term x_i either does or does not occur divided by the ratio of the whole document set in which the term does or does not occur

$$f_3 = \frac{r_i}{R-r_i} \div \frac{n_i}{N-n_i} \quad . \tag{37}$$

d) The last possibility consists in taking into account both the presence as well as the absence of query terms while assuming term independence in the set of relevant records as well as the set of nonrelevant items. This produces the formula

$$f_4 = \frac{r_i}{R-r_i} \div \frac{n_i - r_i}{N-n_i-R+r_i} \quad . \tag{38}$$

Equation (38) is identical with the term importance criteria of equation (28) obtained earlier by using the formal probabilistic model under appropriate term independence assumptions. It appears then that this last expression is formally correct, and should be utilized in interactive retrieval when information is available about the term occurrences in the relevant and nonrelevant records. [56] (When the full contingency information cannot be generated directly, the table can be filled in little by little as relevance data are furnished during interactive retrieval for more and more items. In the latter case, approximate term importance indicators are computed by using a subcollection N' of N, a subset R' of R of relevant records, a subset $N'-R'$ of the nonrelevant items.)

The term relevance factor (equation (28) and (38)) has been studied theoretically and used experimentally in various retrieval tests. Thus, it can be shown formally that when records are retrieved in decreasing order according to the number of matching terms with the query, a term weighting system derived from the relevance factor permits a stricter ranking of the items in decreasing order of presumed relevance which proves at least as effective in terms of recall and precision as the original system without relevance weights. [57] Evaluation output of this type showing the average search precision at fixed values of the recall is shows in Table 4 for a collection of 425 articles in world affairs from Time magazine, averaged over 41 different user queries. [58] In Table 4 a standard term weighting system is used where a_{ij}, the weight of term j in document i, is defined as the frequency of term j in the document. This factor is multiplied by L_j, the term relevance factor for term j, in the output shown on the right-hand side of the Table. It is seen that precision improvements varying from 10 to 20 percent are obtainable even though the theoretically required term independence assumptions are not totally verified for the experimental collection under study. [58]

When term relevance information is obtainable through the interaction with the system, many other possibilities become available for effecting improvements in retrieval. For example, if the accuracy of term i is defined as the ratio of the relevant items in which the term occurs to the total number of items containing the term (r_i/n_i), then formal proofs can be furnished of the usefulness of a thesaurus method in which high accuracy terms related to the ones originally available are added to the queries; similarly, a phrase generation method in which two or more terms each with lower than average term accuracy are replaced by a single "phrase" term with higher than average accuracy will necessarily prove effective in retrieval. Finally, a term weighting system in term accuracy order also furnishes improved retrieval output. [59]

Another approach to the use of term occurrence data in the relevant and nonrelevant items of a collection consists in computing occurrence probabilities for certain term combinations, in both relevant and nonrelevant records. This makes it possible to add to (or to subtract from) the query formulations term sets exhibiting sufficiently high (or sufficiently low) occurrence probabilities in the relevant records of a collection. Once again the effectiveness of the process can be proved formally under well-defined conditions. [60]

Consider again the contingency information of Table 2, and assume that an original query Q contains m terms. The probability of a relevant item containing a term j of Q, $1 \leq j \leq m$, is $p_j = r_j/R$ where r_j is the number of relevant records containing term j, and R the total number of relevant in the collection. Similarly, the probability of a relevant item not containing term j is $(1-p_j)$. If once again the terms are assigned independently to the relevant records, the probability that a relevant item contains exactly the terms $\{x_1,x_2,\dots,x_i\}$ but not the terms $\{x_{i+1},x_{i+2},\dots,x_m\}$ will be $\left(\prod_{k=1}^{i} p_k\right)\left(\prod_{k=i+1}^{m} (1-p_k)\right)$. Since there are exactly $\binom{m}{i}$ ways of choosing i terms out of m, the probability that a relevant record has exactly i terms in common with the query is

$$C(p_1,\dots,p_m,i) = \sum_{\binom{m}{i}} \left(\prod_{k=1}^{i} p_{g(k)}\right)\left(\prod_{k=i+1}^{m} (1-p_{g(k)})\right) \qquad (39)$$

where g is a permutation of the integers $1,2,\dots,m$ and the summation is taken over all $\binom{m}{i}$ combinations of choosing i terms out of m. Finally, the expected number of relevant items exhibiting i or more term matches with a query Q will be $\sum_{k=i}^{m} C(p_1,\dots,p_m,k)$.

Analogously, if n_j-r_j is the number of nonrelevant items containing term j, and $q_j = (n_j-r_j)/(N-R)$, $1\le j\le m$, is the probability that a nonrelevant item contains a term j of Q, and if the terms occur independently in the nonrelevant records, the expected number of nonrelevant items exhibiting i or more terms in common with query Q will be $(N-R)\sum_{k=i}^{m} C(q_1,\dots,q_m,k)$. Given these expressions, it is not difficult to prove that new, improved queries are produced by adding terms $x_{m+1},x_{m+2},\dots,x_{m+\ell}$ when the C-factors exhibit appropriate relationships. For example, ℓ new terms can be added to a query, each with weight Δ/ℓ, $0<\Delta<1$, provided for every i, $1\le i\le \ell$, [60]

$$\sum_{k=i}^{\ell} C(p_{m+1},\dots,p_{m+\ell},k) \ge \sum_{k=i}^{\ell} C(q_{m+1},\dots,q_{m+\ell},k) \quad . \qquad (40)$$

In the same way, terms with appropriate negative characteristics can usefully be subtracted from the query.

Obviously, if the term occurrence information necessary to compute the p_i and q_i probabilities is not available for a sufficient number of terms, or if the sample collections used to obtain the probabilities are not typical of the record collections that one must process in practice, none of the interactive query alteration methods examined in this section is guaranteed to lead to improved retrieval results.

7. File Organization and Record Clustering

One aspect of retrieval that has not so far been mentioned is the choice of a file organization providing effective access to the records. In some situations one can store related records in the same general vicinity within the storage medium so that the number of required storage accesses is limited. Alternatively, even when the related records themselves are scattered in storage, retrieval may be speeded up by storing in a common area the addresses of the related records. The latter strategy is used in the well-known inverted file systems where an auxiliary index stores references to all records sharing a given key word.

Since the computation of vector similarity coefficients is natural when records and information requests are identified by keyword sets, it is easy to extend similarity computations to apply to larger groups of records instead of only to vector pairs. This leads to the notion of a classified file where records whose pair-wise similarity is sufficiently large are grouped into common classes, or clusters. If one assumes that records which are jointly relevant to certain search requests are likely to be identified by similar attribute vectors, such records may then appear in common classes and hence may be retrievable together. [61] In any case, the idea of record classification is well-known and has received extensive use in retrieval. [62-66]

A typical clustered file organization is shown in Fig. 6, where each x represents a record, and the circular configurations are the clusters. The distance between two x's is once again assumed to be inversely related to the similarity between the corresponding attribute vectors. Thus, when two clusters appear close together, and in particular when there is overlap between clusters, the respective records may present considerable similarity. In the diagram of Fig. 6, each record cluster is identified by a central item known as the centroid, which is obtained from the other records in the cluster by a computation similar to that shown in equation (10). The best known cluster generation systems are expensive to use when the number of records to be classified is large. [67] However cheaper methods are known which may be applicable to information files of realistic size. [68,69]

Consider now the file search problem. In general a clustered file search may be carried out by initially comparing the query formulations with the cluster centroids only. This operation is then followed by a comparison between the query and those individual records whose corresponding query-centroid similarity was previously found to be sufficiently large. In principle, a clustered search can be quite rapid because under favorable circumstances large portions of the file are rejected at an early stage of the search, and the detailed examination of the records is restricted to areas in the record space that prove to be productive.

Various strategies are possible in order to isolate the clusters which may contain useful records. The standard approach consists in computing a similarity coefficient between a given query and each cluster centroid, and in submitting to a detailed search clusters whose query-centroid similarity exceeds a given threshold; alternatively, the top i clusters, that is the i clusters with the largest query-centroid similarity coefficient may be examined. The standard strategy is effective when the clusters are of approximately equal size, and when the useful records are concentrated in only a few clusters.

Another possibility consists in computing a probability measure which gives an estimate for each cluster of the number of records in the cluster containing at least k matching attributes with the query. [70] In fact, if p_j is taken as the probability that a random record in a particular cluster contains the j^{th} query attribute, a development analogous to the one leading to the estimate of equation (39) shows that the expected number of records in the cluster having at least k attributes in common with query Q is

$$E_r(k) = q \sum_{i=k}^{m} \sum_{\binom{m}{i}} \left(\prod_{j=1}^{i} p_{g(j)} \right) \left(\prod_{j=i+1}^{m} (1-p_{g(j)}) \right) \qquad (41)$$

where r is the index of the cluster under consideration, q is the number of records in cluster r, and g is again a permutation of the integers {1,2,...,m}.

By computing the E value for the various file clusters, and assuming that a record containing a sufficient number of matching query attributes is in fact relevant to that query, it becomes possible to devise a reasonable cluster search strategy. Let s be the total number of records to be retrieved in a given search, and let $\Delta \geq 0$ be a constant such that any cluster containing Δ or fewer expected number of desired records will not be included in the search (because the expected search payoff would be too small in such a case). An appropriate search strategy may then be the following:

a) retrieve records from clusters for which the expected number of desired records is greater than Δ for each cluster, that is

$$E_r(k) > \Delta \ (r=1,\dots,n)$$

b) since the aggregate number of records to be retrieved is s for properly chosen k, the added condition is

$$\sum_r E_r(k) \geq s \ .$$

When overlap exists among the clusters, it is necessary to subtract from the E value the expected number of records having at least k matching query term that are situated in the intersection between adjacent clusters. When the overlap is small, as it generally must be in an efficient storage organization, the calculated E value of equation (40) may be expected to hold also in the more complicated situation.

The $E_r(k)$ value of expression (40) specifically excludes term relevance information, since p_j is simply the probability that a given query term is contained in a record. Hence a cluster search based on the computation of $E_r(k)$ values will be effective only when the presence of many query terms in a record automatically implies relevance. When term relevance probabilities are available, expression (40) can be modified to include them.

Let $P(w_1,x_i)$ be the probability that a record is relevant to a given query and that it contains term x_i, and consider $P(w_1,x_i,x_j) = P[w_1,x_i)\cap(w_1,x_j)]$. If the events $[w_1,x_i]$ and $[w_1,x_j]$ are independent, $P[(w_1,x_i)\cap(w_1,x_j)] =$ $P(w_1,x_i) \cdot P(w_1,x_j) = P(w_1|x_i)\ P(x_i)\ P(w_1|x_j)\ P(x_j)$. (41)

Defining $P(x_i)$ again as p_i, $P(w_1|x_i)$ as u_i, and $P(w_1|\overline{x}_i)$ as $\overline{u}_i$, it is obvious that expression (40) for E (k) is now transformed into

$$R_r(k) = q \sum_{i=k}^{m} \sum_{\binom{\ell}{i}} \left(\prod_{j=1}^{i} p_{g_{(j)}} \; u_{g_{(j)}}\right) \left(\prod_{j=i+1}^{m} (1-p_{g_j}) \; \overline{u}_{g_j}\right) \qquad (41)$$

where $R_r(k)$ now represents the expected number of relevant records with at least k attributes in common with the query. Appropriate estimates for u_i and $\overline{u}_i$ for pratical use are

$$u_i = \frac{r_i}{n_i} \quad \text{and} \quad \overline{u}_i = \frac{R-r_i}{N-n_i} \; .$$

Experimental cluster search output is shown in Table 5 for the standard cluster search, as well as for the $E_r(k)$ and $R_r(k)$ strategies. [71] It is clear from the data of Table 5, that the $E_r(k)$ function provides improvements only for high-precision searches when it is important to find the most useful single cluster of records. When recall is important, that is, when more than one cluster must be examined the $R_r(k)$ function incorporating term relevance probabilities must be used to obtain improvements in the cluster search effectiveness. Once again, the probabilistic parameters lead to more effective retrieval results, even though the required term independence assumptions may not be met in some instances.

8. Summary

Various formal approaches to information retrieval problems are examined in this study, including purely quantitative criteria reflecting the occurrence characteristics of various entities in a retrieval environment, and more complete structural models of the whole retrieval process. Of particular interest are models based on set mapping operations, vector processing methodologies, and decision theory.

The vector space approach leads to the notion of a low-density space in which the various records exhibit substantial separations from each other. This in turns creates the term discrimination theory which defines a useful index term as one capable of expanding the record space.

The decision theory model produces a linear discriminant function leading to the retrieval or rejection of individual records under specified conditions. An effective term weighting function incorporated into the discriminant function uses both the presence and absence of query terms in the records, and makes assumptions about term independence in both the relevant and the non-relevant records. The term independence assumption in the records of a collection is found to be nearly correct in many retrieval environments.

Two special retrieval situations are also considered including interactive retrieval using relevance information concerning previously retrieved records, and clustered file arrangements permitting rapid searches based on estimates of the cluster productivity in terms of number of included relevant records.

It appears that the large variety of formal approaches to the retrieval process may lead to real advances in retrieval capabilities in the foreseeable future.

REFERENCES

[1] R. A. Fairthorne, Towards Information Retrieval, Butterworths, London, 1961.

[2] R. M. Hayes, Mathematical Models for Information Retrieval, in Natural Languages and the Computer, P.L. Garvin, ed., McGraw Hill Book Company, New York, 1963, p. 268-309.

[3] D. J. Hillman, Study of Theories and Models of Information Storage and Retrieval, Reports 1-9, Center for the Information Sciences, Lehigh University, Bethlehem, Pennsylvania, 1962-1966.

[4] C. N. Mooers, A Mathematical Theory of Language Symbols in Retrieval, Proc. of Int. Conf. on Scientific Information, National Academy of Sciences, Washington, D.C., 1959.

[5] F. Jonker, The Descriptive Continuum - A Generalized Theory of Indexing, Proc. of Int. Conf. on Scientific Information, National Academy of Sciences, Washington, D.C., 1959.

[6] B. C. Vickery, The Structure of Information Retrieval Systems, Proc. of the Int. Conf. on Scientific Information, National Academy of Sciences, Washington, D.C., 1959.

[7] R. A. Fairthorne, Empirical Hyperbolic Distributions (Bradford-Zipf-Mandelbrot) for Bibliometric Description and Prediction, Journal of Documentation, Vol. 25, No. 4, December 1969, p. 319-343.

[8] D. deSolla Price, Theory of Bibliometric and Other Cumulative Advantage Processes, Journal of the ASIS, Vol. 27, No. 5/6, September-October 1976, p. 292-306.

[9] G. K. Zipf, Human Behavior and the Principle of Least Effort, Addison-Wesley Publishing Co., Reading, Massachussetts, 1949.

[10] M. Kochen, Principles of Information Retrieval, Melville Publishing Co., Los Angeles, California, 1974.

[11] J. Tague, Simulation of Information Retrieval Systems, unpublished manuscript.

[12] N. Houston and E. Wall, The Distribution of Term Usage in Manipulative Indexes, Am. Documentation, Vol. 15, No. 2, April 1964, p. 105-114.

[13] E. Wall, Further Implications of the Distributions of Index Term Usage, in Parameters of Information Science, Proc. ADI Annual Meeting, Vol. 1, Spartan Books, Rochelle Park, New Jersey, 1964, p. 457-466.

[14] H. L. Resnikoff, On Information Systems with Emphasis on the Mathematical Sciences, Conference Board of the Mathematical Sciences, Washington, D.C., January 1971.

[15] H. L. Resnikoff and J. L. Dolby, Access: A Study of Information Storage and Retrieval with Emphasis on Library Information Systems, Report, R and D Consultants, Los Altos, California, May 1971.

[16] L. Hodes, Selection of Descriptors According to Discrimination and Redundancy--Application to Chemical Structure Searching, Journal of Chemical Information and Computer Sciences, Vol. 16, No. 2, May 1976, p. 88-93.

[17] P. Zunde and V. Slamecka, Distribution of Indexing Terms for Maximum Efficiency of Information Transmission, Am. Documentation, Vol. 18, No. 2, April 1967, p. 104-108.

[18] G. Salton, Automatic Information Organization and Retrieval, McGraw Hill Book Company, New York, 1968.

[19] W. Marek and Z. Pawlak, Information Storage and Retrieval Systems--Mathematical Foundations, C C PAS Report, No. 149, Computation Center, Polish Academy of Sciences, Warsaw, Poland, 1974.

[20] W. Lipski Jr. and W. Marek, On Information Storage and Retrieval Systems, Banach Center Publications, Vol. 2, Polish Academcy of Sciences, Warsaw, Poland, 1976.

[21] W. Lipski, Jr., Information Storage and Retrieval--Mathematical Foundations II--Computational Problems, Theoretical Computer Science, Vol. 3, North Holland Publishing Co., 1976, p. 183-211.

[22] S. P. Ghosh, File Organizations--The Consecutive Retrieval Property, Communications of the ACM, Vol. 15, 1972, p. 802-808.

[23] S. P. Ghosh, On the Theory of Consecutive Storage of Relevant Records, Information Sciences, Vol. 6, 1973, p. 1-9.

[24] H. Borko, Toward a Thoery of Indexing, in Indexing Concepts and Methods, to be published by Academic Press, New York, 1978.

[25] F. Jonker, Indexing Theory, Indexing Methods and Search Devices, Scarecrow Press, New York, 1964.

[26] B. C. Landry and J. E. Rush, Toward a Theory of Indexing, Journal of the ASIS, Vol. 21, No. 5, Sept-Oct 1970, p. 358-367.

[27] G. Salton, A. Wong, and C. S. Yang, A Vector Space Model for Automatic Indexing, ACM Communications, Vol. 18, No. 11, November 1975, p. 613-620.

[28] G. Salton, A Theory of Indexing, Regional Conference Series No. 18, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1975.

[29] G. Salton, Dynamic Information and Library Processing, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1975.

[30] G. Salton and C. S. Yang, On the Specification of Term Values in Automatic Indexing, Journal of Documentation, Vol. 29, No. 4, December 1973, p. 351-372.

[31] W. M. Sachs, An Approach to Associative Retrieval Through the Theory of Fuzzy Sets, Journal of the ASIS, Vol. 27, No. 2, March-April 1976, p. 85-87.

[32] V. Tahani, A Fuzzy Model of Document Retrieval Systems, Information Processing and Management, Vol. 12, No. 3, 1976, p. 177-188.

[33] T. Radecki, Mathematical Model of Information Retrieval Systems Based on the Concept of Fuzzy Thesaurus, Information Processing and Management, Vol. 12, No. 5, 1976, p. 313-318.

[34] T. Radecki, Mathematical Model and Time Effective Information Retrieval Systems Based on the Theory of Fuzzy Sets, Information Processing and Management, Vol. 13, No. 2, 1977, p. 109-116.

[35] M. E. Maron, Automatic Indexing--An Experimental Inquiry, Journal of the ACM, Vol. 8, No. 3, July 1961, p. 404-417.

[36] M. E. Maron and J. L. Kuhns, On Relevance, Probabilistic Indexing and Information Retrieval, Journal of the ACM, Vol. 7, No. 3, July 1960, p. 216-243.

[37] W. Goffman, A Searching Procedure for Information Retrieval, Information Storage and Retrieval, Vol. 2, 1964, p. 73-78.

[38] A. Bookstein and D. R. Swanson, A Decision Theoretic Foundation for Indexing, Journal of the ASIS, Vol. 26, No. 1, January-February 1975, p. 45-50.

[39] C. J. Van Rijsbergen, A Theoretical Basis for the Use of Cooccurrence Data in Information Retrieval, Journal of Documentation, Vol. 33, No. 2, June 1977, p. 106-119.

[40] A. Bookstein and D. R. Swanson, Probabilistic Models for Automatic Indexing, Journal of the ASIS, Vol. 25, No. 5, September-October 1974, p. 312-318.

[41] D. C. Stone and M. Rubinoff, Statistical Generation of a Technical Vocabulary, American Documentation, Vol. 19, No. 4, October 1968, p. 411-412.

[42] F. J. Damerau, An Experiment in Automatic Indexing, American Documentation, Vol. 16, No. 4, October 1965, p. 283-289.

[43] S. F. Dennis, The Design and Testing of a Fully Automatic Indexing--Searching System for Documents Consisting of Expository Text, in Information Retrieval: A Critical Review, G. Schecter, editor, Thompson Book Company, Washington, D.C., 1967, p. 67-94.

[44] H. P. Edmundson and R. E. Wyllys, Automatic Abstracting and Indexing--Survey and Recommendations, Communications of the ACM, Vol. 4, No. 5, May 1961, p. 226-234.

[45] F. Mosteller and D. L. Wallace, Inference in an Authorship Problem, Journal of the American Statistical Association, Vol. 58, No. 302, June 1963, p. 275-309.

[46] S. P. Harter, A Probabilistic Approach to Keyword Indexing, Journal of the ASIS, Vol. 26, No. 4, July-August 1975, p. 197-206.

[47] S. P. Harter, Probabilistic Approach to Automatic Keyword Indexing, Ph.D. Dissertation, University of Chicago, 1974.

[48] J. A. Swets, Information Retrieval Systems, Science, Vol. 141, July 1963, p. 245-250.

[49] J. A. Swets, Effectiveness of Information Retrieval Systems, American Documentation, Vol. 20, No. 1, January 1969, p. 72-89.

[50] W. J. Cooper, Expected Search Length: A Single Measure of Retrieval Effectiveness Based on the Weak Ordering Action of Retrieval Systems, American Documentation, Vol. 19, 1968, p. 30-41.

[51] C. W. Clevendan, J. Mills, and M. Keen, Factors Determining the Performance of Indexing Systems, Vol. 1-Design; Vol. 2-Test Results, Aslib-Cranfield Research Project, Cranfield England, 1966.

[52] A. Wong, Studies on Clustered Files, Doctoral Thesis, Computer Science Department, Cornell University, 1978.

[53] F. W. Lancaster and S. Fayen, Information Retrieval On-Line, John Wiley and Sons, New York, 1973.

[54] M. E. Lesk and G. Salton, Interactive Search and Retrieval Methods Using Automatic Information Displays, in The SMART Retrieval System, G. Salton, editor, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1971. Chapter 25.

[55] G. Salton, Relevance Feedback and the Optimization of Retrieval Effectiveness, in the SMART Retrieval System, G. Salton, editor, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1971, Chapter 15.

[56] S. E. Robertson and K. Sparck Jones, Relevance Weighting of Search Terms, Journal of the ASIS, Vol. 23, No. 1, May-June 1976, p. 129-146.

[57] C. T. Yu and G. Salton, Precision Weighting-An Effective Automatic Indexing Method, Journal of the ACM, Vol. 23, No. 1, January 1976, p. 76-88.

[58] G. Salton and R. H. Waldstein, Term Relevance Weights in On-Line Information Retrieval, to be published in Information Processing and Management.

[59] C. T. Yu and G. Salton, Effective Information Retrieval Using Term Accuracy, ACM Communications, Vol. 20, No. 3, March 1971, p. 135-142.

[60] C. T. Yu, G. Salton, and M. K. Siu, Effective Automatic Indexing Using Term Addition and Deletion, to appear in Journal of the ACM.

[61] C. J. Van Rijsbergen, Information Retrieval, Butterworths, London and Boston, 1975, p. 37-38.

[62] H. Borko and M. Bernick, Automatic Document Classification, Journal of the ACM, Vol. 10, No. 2, April 1963, p. 151-162, and Vol. 11, No. 2, April 1964, p. 138-151.

[63] S. Schiminovich, Automatic Classification and Retrieval of Documents by Means of a Bibliographic Pattern Discovery Algorithm, Information Storage and Retrieval, Vol. 6, No. 6, May 1971, p. 417-435.

[64] C. J. Van Rijsbergen, A Fast Hierarchic Clustering Algorithm, Computer Journal, Vol. 13, No. 3, August 1970, p. 324-326.

[65] J. G. Augustson and J. Minker, An Analysis of Some Graph Theoretical Clustering Techniques, Journal of the ACM, Vol. 17, No. 4, October 1970, p. 571-588.

[66] N. Sparck Jones, Automatic Keyword Classifications, Butterworths, London, 1971.

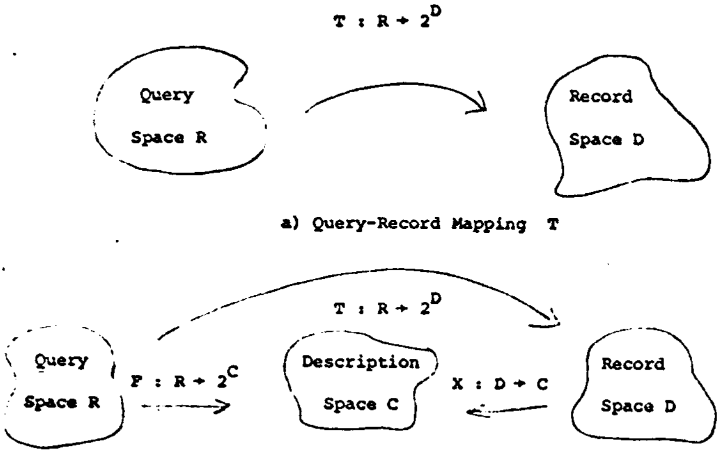
[67] J. Hartigan, Clustering Algorithms, J. Wiley and Sons, New York, 1975.

[68] W. B. Croft, Clustering Large Files of Documents Using the Single Link Method, Computer Laboratory Report, Cambridge University, 1977.

[69] G. Salton and D. Bergmark, Clustered File Generation and its Application to Computer Science Taxonomies, Information Processing 77, B. Gilchrist, editor, North Holland Publishing Company, Amsterdam, Holland, 1977, p. 441-445.

[70] C. T. Yu, W. S. Luk, and M. K. Siu, On the Estimation of the Number of Desired Records with Respect to a Given Query, Technical Report, Computing Science Department, University of Alberta, Edmonton, 1976.

[71] G. Salton and A. Wong, Generation and Search of Clustered Files, Technical Report, Department of Computer Science, Cornell University, Ithaca, New York, 1977.



a) Query-Record Mapping T

b) Retrieval Mapping Based on Indexing Function X

Query-Record Mapping Systems

Fig. 1

$$C = \begin{array}{c} \\ D_1 \\ D_2 \\ \vdots \\ \\ D_n \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & & A_t \end{array} \\ \left[\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1t} \\ a_{21} & a_{22} & \cdots & a_{2t} \\ & & & \\ & & & \\ a_{n1} & a_{n2} & \cdots & a_{nt} \end{array}\right] \end{array}$$

a) Attribute-Record Matrix C

$$T = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_t \end{array} \begin{array}{c} \begin{array}{cccc} A_1 & A_2 & & A_t \end{array} \\ \left[\begin{array}{cccc} t_{11} & t_{12} & \cdots & t_{1t} \\ t_{21} & t_{22} & \cdots & t_{2t} \\ \vdots & & & \\ t_{t1} & t_{t2} & \cdots & t_{tt} \end{array}\right] \end{array}$$

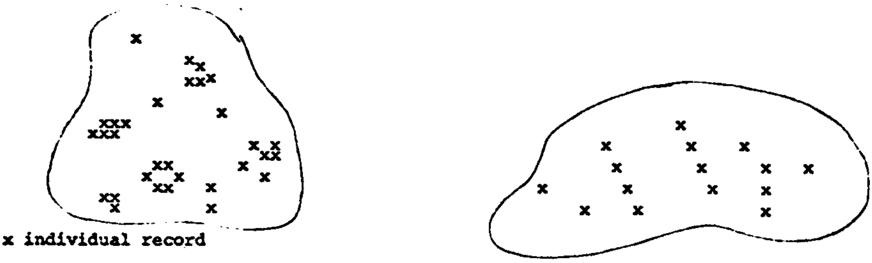
b) Attribute Similarity Matrix T

$$V = \begin{array}{c} \\ D_1 \\ D_3 \\ \vdots \\ D_n \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & & D_n \end{array} \\ \left[\begin{array}{cccc} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & & & \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{array}\right] \end{array}$$

c) Record Similarity Matrix V

Record and Attribute Relations

Fig. 2

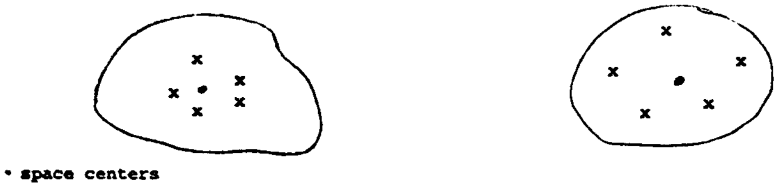


a)Clustered Record Space

b) Separated Record Space

Record Space Configurations

Fig. 3

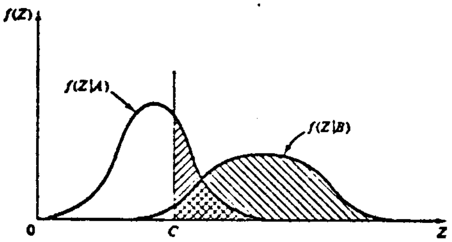


a) Records before assignment of term t

b) Records after assignment of term t

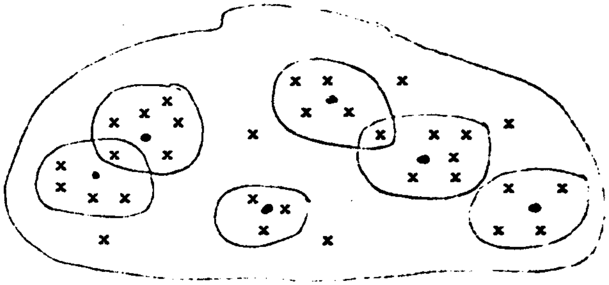
Assignment of Good Discriminating Term

Fig. 4



Probability Density Functions for Relevant Records (B) and Nonrelevant Records (A)

Fig. 5



x individual record

• cluster centroid

Clustered File Organization

Fig. 6

Recall	CRAN 424			MED 450			TIME 425		
	a_{ij}	$a_{ij} \cdot DV_j$		a_{ij}	$a_{ij} \cdot DV_j$		a_{ij}	$a_{ij} \cdot DV_j$	
0.1	0.68	0.68	+0	0.79	0.80	+1%	0.75	0.84	+12%
0.3	0.47	0.54	+15%	0.55	0.59	+7%	0.67	0.72	+7%
0.5	0.31	0.36	+16%	0.44	0.45	+2%	0.64	0.67	+5%
0.7	0.20	0.25	+25%	0.34	0.37	+9%	0.54	0.57	+6%
0.9	0.13	0.13	+0	0.18	0.20	+11%	0.39	0.42	+8%

Search Precision at Fixed Recall Values With and
Without Discrimination Value Weights
(24 user queries for collection)

Table 1

	w_1	w_2	
$x_i=1$	r_i	n_i-r_i	n_i
$x_i=0$	$R-r_i$	$N-n_i-R+r_i$	$N-n_i$
	R	N-R	N

w_1 relevant

w_2 nonrelevant

Occurrence Table for Term x_i and Query Q

Table 2

$\rho_{x,y}$	Computer Science Titles (3.4 M pairs)	Aerodynamics Abstracts (3.5 M pairs)	Clustered Aerodynamics Collection (59 M pairs)
$-1 \leq \rho < -0.1$	0	0.03%	0.10%
$-0.1 \leq \rho < -0.02$	0.06%	8.57%	22.37%
$-0.02 \leq \rho < 0.02$	99.26%	84.71%	47.52%
$0.02 \leq \rho < 0.1$	0.39%	3.55%	0.77%
$0.1 \leq \rho < 0.2$	0.12%	2.70%	0.78%
$0.2 \leq \rho \leq 0.99$	0.15%	0.38%	19.26%
$\rho = 1.0$	0.02%	0.06%	9.20%

Statistical Correlation Coefficient for Term Pairs
in Document Collections

Table 3

Recall	Term Frequency Weights a_{ij}	Term Frequency with Term Relevance $a_{ij} \cdot L_j$
0.1	0.42	0.46 +10%
0.3	0.41	0.46 +11%
0.5	0.39	0.45 +13%
0.7	0.33	0.39 +17%
0.9	0.30	0.36 +19%

Search Precision at Fixed Recall Values
With and Without Term Relevance Weights
(Time Collection 425 documents, 41 user queries)

Table 4

Number of Expanded Clusters	Standard Cluster Search	Search with $E_r(k)$ Estimate	Search with $R_r(k)$ Estimate
1	0.21	0.25 +19%	0.30 +43%
3	0.42	0.33 -21%	0.49 +17%
5	0.52	0.42 -19%	0.61 +17%
7	0.56	0.48 -14%	0.66 +18%
9	0.62	0.59 -9%	0.73 +18%
14	0.75	0.68 -9%	0.80 +7%
20	0.82	0.75 -9%	0.87 +6%

a) Average Recall Results for Various Cluster Search Strategies

Number of Expanded Clusters	Standard Cluster Search	Search with $E_r(k)$ Estimate	Search with $R_r(k)$ Estimate
1	0.24	0.29 +21%	0.36 +50%
3	0.17	0.14 -18%	0.21 +24%
5	0.13	0.11 -15%	0.16 +23%
7	0.10	0.05 -10%	0.12 +20%
9	0.08	0.08 0	0.11 +38%
14	0.07	0.06 -14%	0.09 +29%
20	0.05	0.05 0	0.07 +40%

b) Average Precision Results for Various Cluster Search Strategies

Recall and Precision for Cluster Search Methods
(424 document abstracts in aerodynamics, 24 user queries)

Table 5