Dagobert Soergel

College of Library and Information Services
University of Maryland
CollegePark, MD 20742
Tel. 301-405-2037 Fax 301-314-9145 Home 703-823-2840
ds52@umail.umd.edu

Indexing and retrieval performance: The logical evidence

Submitted to the Journal of the American Society for Information Science, publication planned for May 1994.

Abstract

This paper presents a logical analysis of the characteristics of indexing and their effects on retrieval performance. It establishes the ability to ask the questions one needs to ask as the foundation of performance evaluation, and recall and discrimination as the basic quantitative performance measures for binary non-interactive retrieval systems. It then defines the characteristics of indexing that affect retrieval — namely, indexing devices, viewpoint-based and importance-based indexing exhaustivity, indexing specificity, indexing correctness, and indexing consistency — and examines in detail their effects on retrieval.
It concludes that retrieval performance depends chiefly on the match between indexing and the requirements of the individual query and on the adaptation of the query formulation to the characteristics of the retrieval system, and that the ensuing complexity must be considered in the design and testing of retrieval systems.

Introduction

Indexing consumes the lion's share of database creation costs. Database designers need to know

- whether indexing improves retrieval performance,
- whether the improvement is worth the cost, and
- what characteristics of indexing are important to achieve an improvement.

Many experiments have been conducted to answer these questions. But indexing characteristics and their effects on retrieval are so complex that they largely defy study in artificial test situations. Most experiments fail to account for important interactions among factors as they occur in the real world, and thus give results that mislead more than they enlighten, results that have little meaning for the assessment or improvement of operational retrieval systems. For example, an experiment studying the effect of exhaustivity of indexing must consider the factor of query formulation. Good searching varies the query formulation to adapt to the level of exhaustivity in indexing; in contrast, many experiments studying the effects of exhaustivity leave the query formulation unchanged in the name of experimental control and thus study the effects of exhaustivity under conditions of bad searching, a subject of little interest.

Providing guidance to system designers, indexers, and searchers requires taking a step back and exploring the many factors that bear upon the effects of indexing on retrieval performance, resulting in a framework for the study of these effects through both logical reasoning and empirical research. This is what this article sets out to do; it discusses the assessment of retrieval performance, defines the characteristics of indexing, and examines their effects on retrieval.

Information retrieval extends well beyond retrieval of bibliographic records and of text. Accordingly, we will draw examples from other contexts as well, particularly from software reuse. (Software reuse requires the retrieval of existing software modules that can be incorporated into a new program to serve specified functions, such as sorting data or stemming English words.) We will use **entity** or **item** as a generic term to include documents, program modules, food products, etc.

1 Assessing retrieval performance

> *I can call spirits from the vasty deep.*
> *Why, so can any I, or so can any man; but will they come when you do call for them?*
> (Henry IV, Part 1, Act 3, Scene 1)

At the foundation of all performance evaluation is a simple criterion: **Can one ask the questions one needs or wants to ask and get an answer with acceptable levels of successful retrieval and of distracting noise**? The remainder of this section gives variations on this fundamental theme.

1.1 Relevance, pertinence, utility

The concepts of topical relevance, pertinence, and utility are central for good indexing and for assessing retrieval performance.

Topical relevance is a relationship between an entity and a topic, question, function, or task. A document is topically relevant for a question if it can, in principle, shed light on the question. "In principle" means that the document can do so for a person (or system) who knows the language of the document, has the background to understand the document, and is capable of processing the information transmitted by the document in relation to the question. "Shedding light on" means that the document provides information that either directly answers the question or is part of a premiss set from which the question can be answered through a chain of inferences. The degree of relevance of a document depends on a number of factors: the amount of relevant information given; the strength of the relationship between the information given and the question — for example, the length of the inference chain from the information given to the answer; the strength of the contribution to the quality and/or surety of the answer; and perhaps other factors. (On a discussion of topical relevance see Cooper, 1971 and Wilson, 1973)

For a software module, the question of relevance is a question of applicability to a function to be carried out, such as sorting. A software module is topically relevant for a function if it can, in principle, assist in creating a program carrying out the function. "In principle" means that the software module can do so for a programmer who has the equipment to run the software module, the knowledge needed to fit the module in a larger whole, and the ability to modify the module if necessary. "Assisting" means that the module can be used as is or with modification; that it can serve as a model which facilitates creating a new module (for example, the module found could be re-coded in another programming language); or that it can be incorporated into a larger program being written to carry out the function.

Pertinence is a relationship between an entity and a topic, question, function, or task with respect to a person (or system) with a given purpose. An entity is pertinent if it is topically relevant and if

it is **appropriate** for the person, that is, if the person can understand the document and apply the information gained. (On pertinence see Kemp 1974.)

Utility. An entity has utility if it is pertinent and makes a useful contribution beyond what the user knew already. Utility might be measured in monetary terms ("How much is having found this document worth to the user?") (Goffman & Newill 1967, Cooper 1973). A pertinent document may lack utility for a variety of reasons; for example, the user may already know it or may already know its content.

1.2 Performance measures

Retrieval systems should be judged by what they are designed to do — retrieving relevant items and rejecting irrelevant items in response to a query. The correctness of retrieval decisions is measured by **recall**, the fraction of all relevant items correctly retrieved. Recall can be seen as the probability of an item being retrieved, given that it is relevant. The correctness of rejection decisions is measured by **discrimination**, the fraction of all irrelevant items correctly rejected. Discrimination can be seen as the probability of an item being rejected, given that it is not relevant. The complement of discrimination is **fallout**, the fraction of all irrelevant documents incorrectly retrieved (fallout = 1 - discrimination). The definitions of recall and discrimination can be based on topical relevance, pertinence, or utility, possibly taking into account the degree of relevance (such as considering only highly relevant documents in the computation of recall or including moderately relevant documents as well).

The choice of the measures of retrieval performance to be used in logical analysis and empirical studies is important. Recall and discrimination are the two elemental, basic measures. The more commonly used measure **precision** (the fraction of relevant items in the items retrieved) is a composite measure, the joint result of several factors: The number of relevant items in the collection and recall together determine the number of relevant items retrieved; collection size and discrimination together determine the number of irrelevant items retrieved. Precision is a sensible measure of final answer quality from a user's point of view, but discrimination (or its complement, fallout) is the better measure for studying retrieval effects. Mooers (1959) uses recall and fallout. Goffman and Newill (1967) employ an analogy with the assessment of diagnostic tests and introduce recall (called sensitivity, the fraction of people having a disease testing positive) and discrimination (called specificity, the fraction of people not having a disease testing negative). In a comprehensive review, Robertson (1969) convincingly argues the merits of fallout for measuring retrieval performance. The predominance of precision in spite of this can perhaps be explained by a bandwagon effect.

Much effort has been spent on defining a single measure of retrieval performance that could be used in comparing systems (Swets, 1963, Good, 1967). Such a single measure is meaningful only with reference to a specific request from a specific user in a specific situation. The answer to the question "which retrieval result is better" often depends on the user and her situation; different search requests have different requirements, and the quality of a search can be judged only in light

of these requirements. Thus, a retrieval performance score for an individual query should be computed as a function of several criteria, each weighted according to search requirements as proposed by Bourne et al. (1961). A real-life system can be judged by the average of individual retrieval performance scores, with each search weighted by importance. This method favors flexible systems that let the searcher emphasize the search outcomes required in each search.

In addition to recall and discrimination there are other criteria to judge retrieval effectiveness, such as timeliness and novelty; for a discussion in the framework of bibliographic retrieval systems, see, for example, Soergel 1985. Beyond that, the support a system gives to a searcher in coming up with a good query formulation in the first place is critical for performance.

The discussion so far has been limited to binary retrieval systems: items are considered either relevant or not and are considered either retrieved or not. Now relevance is clearly a matter of degree, and systems are available that make retrieval a matter of degree by assigning each item a score. Such scores can be interpreted either as the degree of relevance predicted by the system or as the system-determined probability that the item is relevant on a binary scale; either way, the scores can be used to rank the items for examination by the user. The implications for measures of retrieval performance are discussed, for example, in King & Bryant, 1971, Ch. 2, and Swets, 1963 and 1969.

One further qualification is needed. The measures discussed assume the approach of a "one-shot" search: A query encompassing the entire information need is formulated, the search run, and the results evaluated. In real life, many searches are at least somewhat interactive: The searcher uses clues from entities retrieved to detect new search paths or to change the search topic, as exemplified most vividly in hypermedia systems. In this process, the searcher may accumulate the information needed in bits and pieces gleaned from documents encountered — the "berry-picking" approach to information retrieval (Bates, 1989). Interactive searches require additional criteria and measures.

While this paper focusses on the effects of indexing on retrieval performance in a binary retrieval system as measured by recall and discrimination, the thinking presented also sheds light on evaluation of retrieval more generally.

2 Indexing characteristics

Before we can talk about the effects of indexing on retrieval performance, we need concepts and vocabulary to talk about the characteristics of indexing that affect retrieval. This section reviews well-known indexing characteristics and defines some new ones. (on Sections 2 and 3 in general see King & Bryant, 1971, Chapters 4 and 5.)

2.1 The over-all approach to indexing

The approach taken to indexing can be expected to have a major impact on indexing quality and therefore on retrieval performance. Indexing can be request-oriented or entity-oriented (see User-centered indexing by Raya Fidel in this issue, Soergel, 1985, Chapter 14, or Cooper 1978). In request-oriented indexing, the index language is built from a detailed study and logical analysis of user requirements and then serves as a communication device from user to indexer. The index language communicates to the indexer a conceptual framework to be used as a checklist in indexing. The idea is to maximize as far as possible the probability that a descriptor needed in searching is available in the index language and is used properly in indexing.

2.2 Indexing devices

There are structural and syntactical indexing devices. The most important structural device is the **hierarchy** of the index language. A well-structured hierarchy, preferably using facet arrangements where appropriate, provides a framework for the indexer and thus supports correct indexing; it is a prerequisite for request-oriented indexing. Hierarchy also supports searching. Associative relationships augment the hierarchy. Synonym relations constitute another structural device which is very important if the indexing vocabulary is not controlled.

The **degree of precombination** (precoordination) is another element of the index language structure (see Soergel, 1985, Chapters 14 and 15). A high degree of precombination makes indexing more difficult and thus may affect indexing correctness, if for no other reason than the increase in the number of descriptors in the index language; as we shall see, precombination also tends to degrade indexing specificity.

Two syntactical devices that are perhaps more important than generally assumed are links and role indicators/relators (See Körner, 1985 for a review). **Links** express relationships between descriptors that are stronger than mere co-occurrence in the indexing of the same entity. They are particularly important in systems with a low degree of precombination. For example, a document about *The effects of alcohol dependence on experimentation with cocaine* would be indexed with *drug dependence : alcohol* and with *experimentation : cocaine*; the links prevent its retrieval in a search for *cocaine dependence*, which would use the query formulation *drug dependence : cocaine*.

Role indicators specify the role of a descriptor in the context of an entity representation. For example, Medline has the role indicators (called subheadings) *therapeutic use* and *adverse effects* to distinguish between the roles a chemical substance may play in a document, for example, *diabetes* - ***drug treatment***, *insulin* - ***therapeutic use***, *triglycerin* - ***adverse effects***. Role indicators must be linked to the descriptor whose role they specify, lest confusion reign. **Relators** represent patterns of role indicator pairs (or triplets, etc.), for example, *diabetes* - ***treated by drug*** - *insulin*.

Weights as an indexing device are discussed in connection with exhaustivity in the next section.

2.3 Exhaustivity of indexing

An entity being indexed is relevant, in varying degrees, to a number of concepts. **Exhaustivity of indexing** is the extent to which these concepts are covered by the descriptors assigned to the entity. Exhaustivity has two components: viewpoint exhaustivity and importance exhaustivity.

A concept cannot be covered in indexing unless it is included in the index language. **Viewpoint exhaustivity** addresses the question: Are the facets or viewpoints useful for retrieval represented in the index language and thus available for retrieval? The degree to which this question can be answered with "yes" is viewpoint exhaustivity. For example, adding the three facets *level of difficulty*, *quality*, and *ideological orientation of the author* increases exhaustivity; so does adding the facet *research method used* in the indexing of research studies.

Importance exhaustivity addresses the question: What is the importance threshold for the assignment of descriptors as prescribed in the indexing rules? For the indexer considering an entity this question takes the form: Which of the concepts associated with this entity are important enough to warrant indexing? This question addresses the entity view of importance exhaustivity, as illustrated in Table 1.

High importance threshold	**Low importance threshold**
Few descriptors, low exhaustivity	Many descriptors, high exhaustivity,
Sample rules	
Use a descriptor only if the entity is **definitely** useful in a search for the descriptor.	Use a descriptor also if the entity **might** be useful in a search for the descriptor, even if the relationship is only tenuous, e.g., little space devoted to the descriptor topic.
Index only the main topics of a document.	Index also minor topics of a document.
Index only the main research method used.	Index all research methods used.
Index only the first ingredient of a food product.	Index all ingredients of a food product.
Index a program module with a function descriptor (such as *Sort* or *Reduce to stem form*) only if the module carries out the function.	Index a program module with a function also if the module could be modified to carry out the function, or if it suggests a good programming technique for writing a new module for the function.

Table 1. **Importance exhaustivity (entity view)**

The searcher considering a descriptor for a query formulation looks at the importance threshold from a different perspective: Will the descriptor find only documents that are centrally important or will it also find documents that just touch on the descriptor? This question addresses the descriptor view of importance exhaustivity. Exhaustivity — a system's indexing rules prescribing an importance threshold — may vary from subject area to subject area or even from descriptor to

descriptor. For example, an information system for food products may prescribe indexing with *wheat* if it is the first or second ingredient but indexing with *mushroom* whenever mushroom is an ingredient, no matter how far down the list (since mushrooms pose a danger to food safety). What really matters in a search is not some general importance threshold but the importance threshold used for each of the descriptors that make up the query.

The rules given to the indexers define the level of exhaustivity intended in the system. Retrieval performance depends also on the application of these rules. Thus, to study the effects of exhaustivity (both viewpoint exhaustivity and importance exhaustivity) one must also consider the completeness of indexing, to be discussed below.

To test the effects of exhaustivity of indexing or to give guidance to a searcher in using a database, one must somehow measure exhaustivity. Unfortunately, measuring indexing exhaustivity is problematic. Given two databases covering the same entity type (two databases covering documents, two databases covering software modules, etc.), we can determine which uses higher exhaustivity as described below (keeping in mind that database 1 may be more exhaustive in one subject area and database 2 in another). Thus we can arrange several databases in a rank order based on their exhaustivity of indexing.

To compare the viewpoint exhaustivity of two data bases, we can compare the index languages used with respect to the facets and individual descriptors included. To compare the importance exhaustivity, we can examine the indexing rules (such as the rules given in Table 1). We might even define an ordinal scale of exhaustivity values defined by such rules. But assigning a quantitative value to the importance threshold used in indexing is well-nigh impossible; any such value would be a fuzzy measure at best.

The average number of descriptors assigned to an entity in the database being studied is often used — somewhat naively — as a stand-in measure for exhaustivity. This would work if exhaustivity was the only determinant of the number of descriptors per document. But there are other factors: the properties of the entity being indexed, the degree of precombination, the correctness of indexing, and the indexing policy with respect to the assignment of descriptors that are broader or otherwise related to a "best-fit" descriptor. Table 2 shows these influences in detail. (See Maron, 1979 for further discussion.)

Influence on the number of descriptors per entity	
Entity properties	At the same level of exhaustivity, a simple program module needs fewer descriptors than a complete software package. A long document often needs more descriptors than a short one. However, a 20-page journal article may need 15 subject descriptors at a medium level of exhaustivity, whereas an economics textbook needs just one: *Economics, general* (i.e., all subjects of economics covered).
Degree of precombination	The single precombined descriptor *Methods of instruction for reading in elementary schools* entails three elemental concepts; in a postcombination system the same topic requires three elemental descriptors, namely *Methods of instruction*, *Reading*, and *Elementary schools*. Thus, as a rule, the higher the degree of precombination, the fewer descriptors are required at a given level of exhaustivity. Therefore, comparing the exhaustivity of indexing of a given entity in two systems requires that all descriptors be reduced to elemental concepts.
Correctness of indexing	Incorrect descriptors increase the count without contributing to exhaustivity.
Indexing rules	Some systems require that the indexer add descriptors that are broader than or otherwise related to a "best-fit" descriptor; for example, a document on *clinical depression* would be indexed by that descriptor (which makes the best fit), but also by the broader descriptors *affective psychosis*, *psychosis*, and *behavioral and mental disorder* — three additional descriptors that do not add to exhaustivity.

Table 2. **Determinants of the numbers of descriptors per entity**.

From a descriptor view, we can compare the exhaustivity in two systems that have the same collection and use the same index language but differ in other parameters, such as indexer, amount of information used in indexing, amount of time used, and index language displays used. The system with more entities for a given descriptor is more exhaustive with respect to that descriptor.

Terminological note: The term **indexing depth** should be avoided; it is sometimes used to mean exhaustivity, sometimes specificity, and sometimes a combination of both.

Weights. Indexing weights differentiate descriptor assignments by importance; this is very useful in systems that use a low importance threshold (high exhaustivity) in indexing. Weights are commonly used in bibliographic databases that are distributed in both print and electronic formats, such as Medline and ERIC; the print index uses only highly weighted descriptors, but the electronic format uses all descriptors.

2.4 Specificity of indexing

Specificity of indexing is the generic level at which the concepts assigned to the entity are expressed. Indexing by *clinical depression* is more specific than indexing by *psychosis*; indexing by *shell sort* is more specific than indexing by *sort*.

Comparing two systems with respect to specificity may be tricky. While one system may be consistently more specific than another, more often one system is more specific in one aspect, and the other system more specific in another aspect. For example, system A may index *diseases* specifically and *research methods* broadly, while system B indexes *diseases* broadly and *research methods* specifically.

As a practical matter, postcombination indexing is often more specific than precombination indexing. The three elemental descriptors *Methods of instruction; Reading;* ***Second grade*** provide more specificity in the grade level facet than the precombined descriptor *Methods of instruction for reading in* ***elementary schools***. It is just not feasible to include all possible precombinations. Therefore, comparing the specificity of indexing in two systems requires that all descriptors be reduced to elemental concepts.

Intended specificity is the standard defined by the descriptors in a system's index language and by indexing policies. Actual specificity may fall short of the standard set by intended specificity due to indexing errors. An indexer may be unable to differentiate between specific descriptors, or information used in indexing may be insufficient for determining the most specific descriptor. Section 3.4 gives examples.

2.5 Indexing correctness

Indexing correctness or absence of indexing errors is of overriding importance for retrieval performance. Indexing is susceptible to two kinds of errors, two kinds of deviations from the indexing rules: Errors of omission — a descriptor that should be assigned is omitted — and errors of commission — a descriptor that should not be assigned is nevertheless assigned. Omitting a correct descriptor and assigning a broader or narrower or related descriptor instead is a special kind of error that is at once an error of omission and commission. In such a case, the searcher can compensate for the error of omission by including related descriptors in the query formulation, and the error of commission is mitigated since an item retrieved by the incorrect, but related, descriptor has at least some relevance. In the measures defined below, some credit should be given for descriptors that are half-correct, but also some blame since the same descriptors are also half-wrong. To further complicate matters, highly weighted descriptors should be given more weight in computing indexing correctness.

In order to determine indexing errors, one must know for each entity which descriptors should and should not be assigned. In reality, things are not that clear-cut. There may well be differences

between good indexers, but we assume that correctness of indexing can be determined, for example, through a consensus of several good indexers and knowledgeable users.

With this caution in mind, we define the following measures of indexing correctness.

Completeness of indexing relates to the presence of the correct descriptor assignments, the absence of errors of omission. We can look at the completeness of indexing from two points of view. Considering an **entity** and the descriptors assigned to it, we can ask: Of the descriptors required for this entity by the rules of the system, how many are actually assigned? This question addresses completeness for each entity taken individually — the entity view of completeness. As a formula:

$$\text{completeness of indexing (entity view)} = \frac{\text{no. of descriptors that are assigned correctly to the entity}}{\text{no. of all descriptors that should be assigned to the entity}}$$

A searcher considering the use of a **descriptor** for a query formulation is more interested in a different question: Of all the entities that should be indexed by the descriptor, how many are actually so indexed? This question addresses completeness from the point of view of a descriptor as it appears in the whole collection — the descriptor view of completeness (King & Bryant, 1971, p. 139, "indexing accuracy"). As a formula:

$$\text{completeness of indexing (descriptor view)} = \frac{\text{no. of entities to which the descriptor is correctly assigned}}{\text{no. of all entities to which the descriptor should be assigned}}$$

The descriptor-view measure is more difficult to determine, but it is more useful for predicting retrieval performance; it is directly related to recall.

The **purity of indexing** relates to the absence of erroneous descriptor assignments, the absence of errors of commission. Purity of indexing can also be viewed from an entity taken individually or from a descriptor as it occurs (or does not occur) in the collection. The formulas are:

$$\text{purity of indexing (entity view)} = \frac{\text{no. of descriptors that were correctly rejected for the entity}}{\text{no of all descriptors that should have been rejected for the entity}}$$

$$\text{purity of indexing (descriptor view)} = \frac{\text{no. of entities for which the descriptor was correctly rejected}}{\text{no. of all entities for which the descriptor should have been rejected}}$$

Again, the descriptor-view measure is more difficult to determine but it is more useful for predicting retrieval performance; it is directly related to discrimination.

There are other measures that address indexing errors of commission. We chose purity of indexing because it is a positive measure; the higher its value, the better the indexing. Its complement, impurity of indexing, would count incorrect descriptor assignments rather than descriptor assignments correctly rejected; in some ways, that might be more intuitive, but the parallel to completeness would be lost. Impurity of indexing (descriptor view) is directly related to fallout. Other intuitive measures would be the fraction of all descriptors assigned to an entity that are correct or incorrect, respectively, but relationship of these measures to retrieval performance is less clear.

Indexing correctness must be measured in terms of the rules of the system. A document may be indexed with three elemental descriptors in system A and be 100% complete (system A uses very low exhaustivity and requires only 3 descriptors) while being indexed with 20 elemental descriptors in system B and still being only 50% complete (system B uses high exhaustivity and requires 40 descriptors). A descriptor correctly assigned in a high-exhaustivity system may be erroneous in a low-exhaustivity system.

2.6 Indexing consistency

Indexing consistency is not important in itself; it has been used as an indicator of indexing correctness, but that is problematic. Indexing can be consistently wrong; all indexers might miss an important implication of a document and thus omit an important descriptor. On the other hand, high indexing correctness results in high consistency (two indexers achieving indexing completeness and purity of 1 are also entirely consistent); thus, high consistency is a necessary, but not sufficient, condition for high correctness. The situation is parallel to measurement: high reliability is a necessary, but not sufficient, condition for high validity.

Indexing correctness and indexing consistency are both measures of agreement. Indexing correctness is measured as the asymmetric agreement of the descriptors assigned by the indexer with the descriptors that should be assigned. Indexing consistency can be measured by any of the many measures for symmetric or asymmetric agreement between two indexers (inter-indexer consistency) or between two indexing sessions by the same indexer (intra-indexer consistency). The considerations on descriptor relationships and descriptor weights from Section 2.5 apply here as well. The following is an example of a symmetric measure:

$$\text{indexing consistency (entity view)} = \frac{\text{no. of descriptors assigned to the entity by both A and B}}{\text{no of descriptors assigned to the entity by A or B}}$$

indexing consistency (descriptor view) =

$$\frac{\text{no. of entities to which the descriptor was assigned by both A and B}}{\text{no. of entities to which the descriptor was assigned by A or B}}$$

(Descriptor-view measure defined in King & Bryant, 1971, p. 138).

3 Effects of indexing characteristics on retrieval performance

There are a few simple truths that make the relationship between indexing characteristics and retrieval performance extraordinarily complex and very hard to test through experiments.

- Important determinants of retrieval performance are not universal but idiosyncratic to the query at hand: Does the index language include the descriptors needed to express the query topic? Do the indexers' judgments in applying these descriptors match the requirements of the query? Does the index language include the hierarchical relationships useful for processing the query? In short, indexing performance depends on the match between indexing characteristics and the requirements of the individual query.

- The quality of searching plays a major role both in exploiting the strengths of indexing to the fullest and in compensating for its weaknesses. It is good search practice to adapt the query formulation to the indexing environment, for example, to exhaustivity of indexing; this is the only way to ensure the best retrieval possible under the constraints of the given indexing characteristics. This principle creates a quandary for experiments that try to compare several retrieval systems; if such an experiment uses the same query formulation with all of them (to hold this factor constant), it measures retrieval performance under conditions of bad searching.

- Indexing characteristics are not the only determinant of retrieval performance; the retrieval mechanism also plays an important role. The searcher can do a lot more with a powerful computer search system then with a printed index or card catalog.

These truths are illustrated again and again in the analysis that follows.

3.1 Effects of the approach to indexing

The approach to indexing has a large effect on the availability of just the right descriptors needed for searching and on the correctness of indexing; thus it can be expected to have a major impact on retrieval performance. Request-oriented indexing is designed to increase dramatically the ability to ask the questions one needs or wants to ask. Unfortunately, there is only one rather small study investigating this important variable, with encouraging but preliminary results (Pejtersen, 1980, 1986, Pejtersen & Austin 1983, 1984).

3.2 Effects of indexing devices

3.2.1 Effects of hierarchy

Hierarchy provides a framework for the indexer and thus has positive effects on indexing correctness, which in turn improves retrieval. Hierarchy also has a direct effect on searching. It

provides a framework for the searcher in formulating the query and thus supports choosing the most appropriate descriptors. The structure may even assist the user in thinking about her problem and discovering ramifications and new aspects.

In the search itself, hierarchy is the basis for **inclusive searching** (Medline EXPLODE, Predicasts CASCADE). In inclusive searching, a query descriptor retrieves entities indexed by any narrower descriptor as well; for example, an inclusive search for *psychosis* finds as well documents indexed by any specific *psychosis*, such as *schizophrenia* (including *paranoid schizophrenia*, *schizoaffective disorder*, etc.) and *affective psychosis* (including *manic disorder*, *depression*, etc.). Inclusive searching applies the knowledge encapsulated in the index language hierarchy to provide a very powerful search tool that boosts recall. Inclusive searching is particularly important in systems that use high specificity of indexing or a high degree of precombination.

3.2.2 Effects of precombination

Precombination comes in very handy when there is a precombined descriptor that matches the query topic. In a printed index, where searching for a combination of descriptors is impractical, a precombined descriptor matching the query topic makes searching feasible. In a computer search system, a precombined descriptor helps avoid spurious combinations; for example, combining the elemental descriptors *drug dependence AND cocaine* without links would retrieve, through spurious combination, a document on *The effect of alcohol dependence on experimentation with cocaine*; using the precombined descriptor *cocaine dependence* would prevent the erroneous retrieval. Spurious combinations can also be prevented through the use of links.

Precombination can also create difficulties in retrieval. If *alcohol dependence*, *cocaine dependence*, *heroin dependence*, *marijuana dependence*, and *nicotine dependence* are all precombined descriptors, they must all be included in a search for *drug dependence*. This does not present a problem **if** the search system supports inclusive searching **and if** the hierarchy includes the relationships *drug dependence* NT *alcohol dependence*, etc.; unless **both** conditions are fulfilled, the searcher must compensate for the deficiency by including the precombined descriptors in the query formulation, lest recall suffer. For example, even though Medline supports inclusive searching, a search for *eye, inclusive* does not retrieve all documents on *eye neoplasms* since *eye neoplasms* is not a narrower term of *eye* and the database contains many documents indexed by *eye neoplasms* but not by *eye*.

3.2.3 Effects of links and role indicators

Links and role indicators can be used to formulate a more discriminating query, but their use makes for another opportunity for mismatch between indexers and searchers, and thus recall may drop. Many links are very obvious and easy to use consistently, such as the link between *drug dependence* and the specific drug. Role indicators are more complex. Rules can be overly restrictive, such as allowing only one role indicator for a descriptor, while in reality a concept often

plays several roles in the context of a document; for example, in Medline indexing a prescription drug is often shown in the two roles *therapeutic use* and *adverse effects*. The semantic relationships between role indicators must be considered in searching to counteract a mismatch of interpretation between indexer and searcher; for example, the roles *acted upon* and *result* have a certain semantic affinity that may lead to the indexer using one while the searcher would search under the other. Thus, it is prudent to search for *acted upon* OR *result*. Early experiments gave role indicators a bad name (e.g., Sinnett, 1964). However, the performance degradations reported reflect more on poorly designed systems operated by poorly trained people than on the merits of the concept. Montague (1965) gives a balanced account. Role indicators may be particularly useful for queries that combine very broad concepts **in specified roles**.

3.3 Effects of indexing exhaustivity

Common wisdom has it that indexing exhaustivity increases recall at the cost of discrimination and that specificity increases discrimination at the cost of recall. Common wisdom has it wrong half of the time.

Higher **viewpoint exhaustivity** extends the kinds of questions that can be asked of a system. It also affords the searcher an opportunity to achieve higher discrimination, possibly but not necessarily at the cost of decreased recall. Assume a system adds *research method used* to the viewpoints indexed. In a search for *longitudinal studies of attitudes toward drunk driving using the panel method* the searcher can now add a research methods descriptor, *panel study*, to increase discrimination. The amended query formulation may also fail to retrieve some relevant documents, depending on the importance exhaustivity and the indexing completeness associated with the descriptor *panel study*. (Indexing completeness might be low if the indexers are not sufficiently trained in research methods to recognize the use of the panel method.) When viewpoints such as *reading level* or *program complexity* are indexed, discrimination with respect to pertinence can be increased. Adding viewpoints to a query formulation is strictly an option; if the searcher expects that this option would reduce recall and recall is at a premium, she may elect not to use that option.

The effects of **importance exhaustivity** are more complex. When the query formulation is held constant, higher exhaustivity delivers the same or higher recall with the same or lower discrimination, depending on the situation. Consider a search for *depression* by two different users, (1) a physician who wants to brush up on the latest developments in *depression* and (2) a medical researcher who wants to do a thorough review of all aspects of *depression*. Both users run the same search and get the same results, but their **differing requirements lead to a different interpretation of these results** as illustrated in Table 3.

User	Low exhaustivity	High exhaustivity
A physician who wants to brush up on the latest developments in *depression*.	Retrieves most of the documents the physician needs without too much noise. The noise consists of borderline documents that the indexer judged quite important for *depression* but that the user finds irrelevant. The missed documents are borderline documents that the indexer did not judge important enough but that the user would find relevant.	Retrieves the same documents, plus many more that just touch on *depression* and thus are not of interest **to the physician**. A few of the added documents may be relevant — borderline documents that are judged differently by the indexer and the user.
	Good recall, good discrimination. **Good importance threshold match:** The indexer uses high, the user requires high.	Quite good recall, poor discrimination. Importance threshold mismatch: The indexer uses low, the user requires high.
A medical researcher who wants to do a thorough review of all aspects of *depression*.	Misses many of the documents the researcher needs and rejects almost all irrelevant documents.	Retrieves many more documents, most of them relevant **for the researcher**.
	Poor recall, quite good discrimination. Importance threshold mismatch: The indexer uses high, the user requires low.	Good recall, good discrimination. **Good importance threshold match**: The indexer uses high, the user requires high.

Table 3. **Effects of importance exhaustivity for different user requirements**.

High importance exhaustivity can sometimes be exploited to achieve higher discrimination, since it affords the searcher an opportunity to reformulate the query by adding an additional restrictive criterion. Consider a search on *alcohol dependence and depression*; a document is judged relevant if it concentrates on *alcohol dependence* and at least mentions *depression*. With low exhaustivity the query should be just *alcohol dependence*; qualifying with *depression* would reject so many relevant documents as to be unacceptable. The documents found must be screened for mention of *depression*. With high exhaustivity and weights, the query could be narrowed to *alcohol dependence* (highly weighted) AND *depression* (any weight). This would boost discrimination and detract little from recall; the only documents missed would be borderline documents where the indexer thought the mention of *depression* too insignificant while the user judged the document relevant. In each case, the query formulation was adapted to achieve the best retrieval possible, given the level of importance exhaustivity. In bibliographic and other text-based systems, free-text searching may be used to compensate for low indexing exhaustivity, provided the search concept is represented explicitly by a word or phrase in most relevant documents.

Indexing weights allow the searcher to choose for each query descriptor the level of importance exhaustivity best suited to the user's requirements, as illustrated in the preceding example. In a

high-exhaustivity system without weights, the query formulation would be *alcohol dependence* AND *depression*, retrieving documents in which both disorders are mentioned, including documents where *alcohol dependence* is just mentioned, not dealt with intensively as required by the user. Thus, this query would find both more and less, and the effect on recall and discrimination depends on what is in the collection.

3.4 Effects of indexing specificity

The effects of indexing specificity depend on the specificity of the search. A specific search can take advantage of specific descriptors to increase discrimination, but for a broad search specific indexing does not help and may even be harmful.

The effect of specific indexing on a specific search depend on the correctness of indexing; specific descriptors make the search more vulnerable to indexing errors. If the indexing is correct, using a specific descriptor hurts recall very little, if at all. But if the indexers are often unable to determine the proper specific descriptor and resort to assigning a broader descriptor instead, specific descriptors become less reliable. The following examples, from retrieval systems dealing with different kinds of entities, illustrate the point. A document relevant for *x-ray film tomography* may have been indexed, erroneously, under the broader descriptor *film radiography* or by the neighbor descriptor *scanography*. A job seeker qualified for an open job as *furnace operator* may have been indexed by the broader descriptor *foundry worker*. A data set relevant to the study of attitudes toward *power* may have been indexed by the neighbor term *authority*. A patient suffering from *depression* may have been diagnosed under the broader descriptor *psychosis*. In all these cases, the broader descriptor retrieves the relevant entity, but the specific descriptor misses it due to the indexing error. With inclusive searching, the searcher can always search under the broader descriptor inclusively to increase recall at the cost of discrimination.

The effect of specific indexing on a broad search depends on the capabilities of the search system. Assume a broad search on *psychosis*; relevant documents may be indexed by *psychosis* or any of its narrower descriptors, such as *affective psychosis*, *depression*, etc. As long as the index language has a well-developed hierarchy and the search system supports inclusive searching, this is not a problem and neither search effort nor retrieval performance are affected. But when inclusive searching is not provided, the searcher looking for a broad descriptor must remember to include in the query formulation all the narrower descriptors seen from the hierarchy — quite laborious and a drawback of specificity in the environment of poor retrieval software or of card catalogs and printed indexes. When the index language does not have a good hierarchy, things are even worse: The searcher must think of all the narrower descriptors, possibly consulting reference tools. Since the searcher cannot be expected to come up with a complete list of narrower descriptors, recall suffers.

3.5 Effects of indexing correctness

Indexing correctness is a major determinant of retrieval performance. Assume a one-concept query that can be expressed well by a descriptor in the index language; assume furthermore that the importance threshold used by the indexers matches the user's requirements. In this situation recall is equal to the completeness of indexing (descriptor view), and discrimination to the purity of indexing (descriptor view). With correct indexing, the searcher can rely on the descriptor assignments and feel confident about retrieval results. This is particularly important in the search for a concept that would be hard to search using a free-text approach.

The following examples, taken from the *Alcohol and Other Drugs Thesaurus*, illustrate this point. The thesaurus includes the descriptor *acculturation* because of the interest in this topic. A document on the *introduction of alcohol into the community of the Hare Indians and the integration of drinking behavior into their culture* must be indexed by *acculturation*, or it will be lost to retrieval under this topic or the broader topic *socio-cultural change* — an indexing error of omission leads to low recall. (The abstract of the document does not mention *acculturation* or *change*, thus a free-text search would not help. On the other hand, a free-text search for *Hare Indians* might be quite successful.) The thesaurus also includes the descriptor *dual diagnosis* for the condition of simultaneous *drug dependence* and another *behavioral or mental disorder* because this is a "hot topic". Indexers must be vigilant to spot the co-occurrence of any kind of *drug dependence* with any other *behavioral or mental disorder* **in the same person** and then use *dual diagnosis* (co-occurrence of appropriate terms in the same abstract is **not** enough). As a last example, consider the document on *The effects of alcohol dependence on experimentation with cocaine*; it must be found under the descriptor *gateway drugs*. Again, if the indexers' vigilance fails, retrieval losses will occur. Conversely, "over-indexing" with a descriptor due to lack of proper understanding hurts discrimination. These examples illustrate that correct indexing requires indexers who are well-trained and familiar with the subject matter and the needs of the users as reflected in the index language.

In practice it is difficult and laborious to measure the correctness of indexing; experiments to study its effects are, therefore, at the borderline of feasibility. Their focus would need to be not so much on the positive effects of indexing correctness, which is self-evident, as on the question to what extent lack in indexing correctness can be overcome in searching.

3.6 Effects of indexing consistency

Indexing consistency affects retrieval performance only indirectly, through its possible effect on indexing correctness, which was discussed in Section 2.6.

3.7 Indexing characteristics, recall, and discrimination

Indexing characteristics are often divided into those that support recall and those that support discrimination. However, as the foregoing analysis shows, many indexing characteristics support both, depending on the situation.

A recall device can often be exploited for a query reformulation that boosts precision. The following example illustrates this for the recall device **inclusive searching**. Assume a search on *depression in elderly cancer patients*. There are three search concepts, *depression*, *elderly*, and *cancer*. *Cancer* has many, many narrower descriptors; without inclusive searching, they would all have to be ORed to express the third search concept — not a feasible option. Free-text searching does not help, since the text of many relevant documents contains a term for a specific kind of cancer but not the word *cancer* itself. So the searcher goes with *depression* AND *elderly* for a low discrimination answer and then screens the documents found for mention of any form of cancer. With inclusive searching, the third search component can be expressed as *cancer, inclusive*, resulting in a search with high discrimination.

Conversely, a discrimination device, such as role indicators, often makes it feasible to ask a question that otherwise would return so many entities as to be unrealistic. In that case one can say that recall was increased from 0 (the value if the question is not even asked) to whatever can be achieved with a reasonably discriminating query.

Conclusion

The discussion of the characteristics of indexing and of the mechanisms through which they affect retrieval performance has demonstrated the complexity of the interactions not only among the indexing characteristics themselves but also with the characteristics of the search system used. To make matters worse, many of the variables defy quantitative measurement. This situation is not made for classical experiments where all variables except the variable under study are held constant or manipulated to study their effects in a carefully controlled way. It may be unwise to accept the results of an experimental study in this area before the study design has been scrutinized to ensure that all the effects and interconnections discussed here have been considered and that all the pitfalls have been avoided. This area may be more suitable for qualitative research methods, methods that take a holistic approach and are sensitive to the special circumstances and total context of individual cases. Qualitative studies may not deliver the clear-cut and "safe" quantitative results some look for, but they may provide insights into the functioning of retrieval systems — perhaps tentative, perhaps uncertain, perhaps qualified, but insights close enough to reality to be useful for design.

Some design conclusions can be drawn from the logical analysis provided here. Best overall retrieval results can be achieved with systems that are flexible, systems that let the searcher emphasize the search outcomes in accordance with the requirements of each search. Such flexibility can be achieved through the following features:

- An index language that covers all viewpoints needed in retrieval and arranges its descriptors in a well-structured hierarchy.

- Links and role indicators.

- Exhaustive indexing (low importance threshold) with weights.

- Specific indexing and inclusive searching.

- Well-trained indexers who are sufficiently familiar with the subject matter and with user needs as reflected in the index language to make the judgments necessary for correct indexing.

The discussion of retrieval effects revolved around an important principle: Retrieval performance is a function of the agreement of the judgments of two parties, the indexer and the user, with respect to the viewpoints used in indexing, the general importance threshold used, and the specific judgments regarding the importance of a specific descriptor for a specific entity. The goal of indexing must be to maximize that agreement.

This principle can be stated more generally. Information retrieval is about meaning. While we can in many cases get at meaning through statistical and syntactic/semantic processing, in many other cases — perhaps the more important ones — we cannot, and human judgment — no matter how often it is maligned as subjective — must step in.

References

The alcohol and other drugs thesaurus. A guide to concepts and terminology in substance abuse and addiction. (1993) Rockville, MD: National Institute on Alcohol Abuse and Alcoholism and Center for Substance Abuse Prevention.

Bates, Marcia J. (1989). The design of browsing and berrypicking techniques for the online search interface. Online Review, 13(5), 407-424.

Bourne, C.P., Peterson, G.D., Lefkowitz, B., & Ford, D. (1961). Requirements, criteria, and measures of performance for information storage and retrieval systems. Stanford Res. Inst. Project Report No. 3741. (AD 270942)

Cooper, William S. (1971). A definition of relevance for information retrieval. Information Storage and Retrieval, 7(1), 19-37.

Cooper, William S. (1973). On selecting a measure of retrieval effectiveness. Part I. Philosophy. Journal of the American Society of Information Science, 24(2), 87-100.

Cooper, William S. (1973). On selecting a measure of retrieval effectiveness. Part II. Implementation of the philosophy. Journal of the American Society of Information Science, 24(6), 413-424.

Cooper, William S. (1978). Indexing documents by Gedanken experimentation. Journal of the American Society for Information Science, 29(3), 107-119.

Goffman, William, & Newill, Vaun A. (1967). A Methodology for test and evaluation of information retrieval systems. Information Storage Retrieval, 3, 19- 25.

Good, I.J. (1967). The decision-theory approach to the evaluation of information-retrieval systems. Information Storage Retrieval, 3(2), 31-34.

Kemp, B.D. (1974). Relevance, pertinence and information system development. Information Storage and Retrieval, 10(2), 37-47.

King, D.W., & Bryant, E.C. (1971). The Evaluation of Information Services and Products. Washington, D.C.: Information Resources Press.

Körner, Horst G. (1985). Syntax und Gewichtung in Informationssprachen: Ein Fortschrittsbericht ueber präzisere Indexierung und Computer-Suche. [Syntax and weighting in information languages: A state-of-the-art report on more precise indexing and computer retrieval]. Nachrichten für Dokumentation, 36(2), 82-100.

Maron, M. E. (1979). Depth of indexing. Journal of the American Society of Information Science, 30(4), 224-228.

Montague, Barbara A. (1965). Testing, comparison, and evaluation of recall, relevance, and cost of coordinate indexing with links and roles. American Documentation, 16(3), 201-208.

Mooers, C.N. (1959). The intensive sample test for the objective evaluation of the performance of information retrieval systems. Cambridge, MA: Zator Co.

Pejtersen, A.M. (1980). Design of a classification scheme for fiction based on an analysis of actual user-librarian communication, and use of the scheme for control of librarians' search strategies. In O. Harbo & L. Kajber (Ed.), Theory and application of information research (pp. 146-159). London: Mansell.

Pejtersen, A.M. (1986). Design and text of a database for fiction based on an analysis of children's search behavior. In P. Ingwerson L. Kajber A. M. Pejtersen (Eds.), Information technology and information use: Toward a unified view of information technology (pp. 125-145). London: Taylor Graham.

Pejtersen, A.M., & Austin, J. (1983). Fiction retrieval: experimental design and evaluation of a search system based on users' value criteria. Part 1. J. Doc., 39(4), 230-246.

Pejtersen, A.M., & Austin, J. (1984). Fiction retrieval: experimental design and evaluation of a search system based on users' value criteria. Part 2. J. Doc., 40(1), 25-35.

Robertson, S.E. (1969). The parametric description of retrieval tests. The Journal of Documentation, 25(1), 1-27.

Sinnett, J.D. (1964). An evaluation of links and roles used in information retrieval. Technical Report.

Soergel, Dagobert. (1985). Organizing information. Principles of data base and retrieval systems.. Orlando, FL: Academic Press.

Swets, John A. (1963). Information retrieval systems. Science, 141(3577), 245-250.

Swets, John A. (1969). Effectiveness of information retrieval methods. Journal of the American Society of Information Science, 20(1), 72-89.

Wilson, Patrick (1973). Situational relevance. Information Storage and Retrieval, 9(8), 457-471.