EVALUATION OF INFORMATION RETRIEVAL SYSTEMS:

A SIMULATION AND COST APPROACH

Michael David Cooper

School of Librarianship

University of California

Berkeley, California

94720

May 1971

Evaluation of Information Retrieval Systems:

A Simulation and Cost Approach

by

Michael David Cooper

DISSERTATION

Submitted in partial satisfaction of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Librarianship

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge

Dr. M.E. Maron, Chairman

Dr. Patrick G. Wilson

Dr. C. West Churchman

Dr. Raynard C. Swank

Dr. Robert M. Hayes

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I have benefited from the help of a number of people during my career. Dr. Richard E. Beckwith, Mr. Donald Cross, Dr. W. Lee Hansen, and Dr. L. David Heggie all provided encouragement at various times. To Dr. Ferdinand F. Leimkuhler I owe a very special thanks for his advice and aid during the past years.

I wish to thank my chairman Dr. M.E. Maron for the many valuable suggestions he has made to drafts of this dissertation; to Ruth J. Patrick for reviewing parts of this paper; and to my colleague Caryl K. McAllister for helping me with a number of methodological problems in this dissertation.

To Dr. C. West Churchman I owe a debt of gratitude for the considerable time that he has spent helping me and for the assistance that he has given me with all aspects of this project.

I have also spent countless hours with Dr. Patrick Wilson discussing not only this dissertation but many other problems of information science as well. I have profited enormously from these dialogs and from the wit and perception that he has brought to them.

Finally, I would like to thank I. Earl Cooper and Bessie Cooper for their encouragement throughout the years.

Computer time for this dissertation was provided by the Computer Center at the University of California, Berkeley and by the Campus Computing Network at the University of California, Los Angeles.

Evaluation of Information Retrieval Systems:

A Simulation and Cost Approach

Abstract

Michael David Cooper

This dissertation examines problems of how to evaluate an information retrieval system. Two specific approaches are explored. The first is a mathematical model for use in studying how to minimize the cost of operating a mechanized retrieval system. Through the use of cost analysis, the model provides a method for comparative evaluation between systems. The cost model divides the costs of a retrieval system into two components: system costs and user costs. In addition, it suggests that a trade off exists between the performance level of the system and the combination of user and system time that is expended in working with the system. With this approach it is possible to determine the allocation of user and system time that minimizes the total cost of operating the system. This allocation is done for a given performance level and for a given cost per unit of user and system time.

The second approach to the evaluation of literature searching systems is the development of a simulation model as a preliminary step toward the creation of a tool for system design and evaluation. The simulation program creates a well specified collection of documents and analyzes the effect of changes in query file characteristics on system performance. First a thesaurus of term relations is generated. Then, employing the thesaurus, routines generate pseudo-documents and pseudo-queries. These pseudo-documents and pseudo-queries are then compared

to see the effect of various query file parameter changes on the quantity of material retrieved.

Evaluation of the simulation output indicates that there are small differences between the results of the experimental runs. It is concluded that one method for generating pseudo-queries is not clearly better than another. It is believed, however, that the simulation model as an approach to the evaluation of retrieval systems provides a limited but useful framework for the evaluation of information retrieval systems.

Chapter 1

Introduction

1. Introduction.

The members of a society have many needs. They require goods such as food and clothing and services such as medical assistance. Another requirement that individuals have is for information: information about things, methods, places, events, and ideas. As the population of the world increases, there will be more goods and services produced to meet demands. At the same time, as the technological complexity of the world increases, more people will require and also generate information.

As more information is required and as more is supplied by individuals, governmental units, businesses, and educational institutions, the greater will be the requirements for efficient methods of communication. Better methods for information transfer are needed in an increasingly complex society.

One possibility for improving the information dissemination process is to use computers. The rapid growth in computing technology has resulted in the development of very fast computational devices and memory units having large information storage capacities. The capabilities of such machines are beginning to be used in the process of information storage, retrieval, and dissemination. With the growth in mechanized retrieval systems has come a variety of techniques for processing documents to identify their content and a variety of rules for retrieving the documents once they are stored in the computer.

An important problem that must be carefully examined is whether one technique for information retrieval is better or worse than another. For example, when searching through a large data base to find documents that satisfy a query, a number of different methods can be employed.

Similarly there are various methods for automatically assigning content indicators, such as index terms, to a document. Another problem has to do with the concept of relevance. What does it mean to say that a document is relevant to a user's need, and how can this be measured and predicted?

This dissertation examines some of these problems as a first step toward analyzing how best to evaluate an information retrieval system. Two specific approaches are explored. The first is a model for use in studying how to minimize the cost of operating a mechanized retrieval system. Through the use of cost analysis, the model provides a method for comparative evaluation between systems. The second is a simulation program which generates a well defined set of documents and analyzes the effect of changes in query file characteristics on system performance. The application of simulation to the analysis of query and document files of an information retrieval system has not been tried before and it is felt that this approach may prove to be a valuable evaluative tool.

In Chapter 2 of this dissertation a number of concepts and problems related to the development of information retrieval systems are presented. Distinctions are drawn between literature searching systems and question-answering systems. An overview of the functioning of an information retrieval system is also presented. Included in the chapter is an analysis of the components of an automated information retrieval system.

Chapter 2 shows that there is a large array of alternative components that can be used to construct information retrieval systems. An important question that must be addressed is how to decide among the alternatives. Is System A better than System B, and if so, how much better?

The question is very important. Unless it is known whether or not one technique for search and retrieval is, in fact, an improvement over a prior technique, it will not be possible to determine if improved systems are really being developed. In Chapter 3, then, a review and discussion of the 'standard' ways of evaluating information retrieval systems is presented. The chapter continues by noting that the evaluation problems have been largely ignored in the literature except for a few well tried methods such as the measurement of user satisfaction with material retrieved. It is suggested that new methods may be in order. An analysis of a cost approach and a simulation approach to the problem are presented as possible techniques.

The ideal form of an evaluation technique would be one that has general applicability, is easy to use and is conclusive. While this paper does not develop such a technique, it does evaluate the feasibility of a cost model for system measurement. The model divides the costs of a retrieval system into two components: system costs and user costs. In addition, it suggests that a trade off exists between the performance level of the system and the combination of user and system time that is expended in working with the system. With this approach, it is possible to determine the allocation of user and system time that minimizes the total cost of operating the system. This is done for a given performance level and for a given cost per unit of user and system time.

In addition to the cost model, a simulation model is developed as a preliminary step toward the creation of a tool for system design and evaluation. The simulation program creates a static collection of pseudo-documents and pseudo-queries. First a thesaurus of term relations is generated. Then employing the thesaurus, routines generate documents

and queries. These are compared to see the effect of various parameter changes on the quantity of material retrieved.

In the later part of the dissertation, the simulation model is evaluated for a set of cases. The simulation output indicates that there are small differences between the results of the experimental runs. It is concluded that one method for generating pseudo-queries is not clearly better than another.

The simulation model is by no means complete. A complete model would imply that there exists a theory of how information is represented in the receiver, a theory concerning the meaning of an information need and an explication of the meaning of information, to name only a few of the more difficult problems. While these problems have not yet been solved, it does appear that a simulation model can have a useful, if more limited, role in evaluation of alternative information retrieval systems. In particular, the model developed in this paper appears useful in studying the variables connected with the process of query formulation against a well defined document collection.

Thus the major problem that this dissertation examines is that of evaluating information retrieval systems. More specifically, it examines analytic models and simulation models as two techniques for limited evaluation of retrieval systems.

Chapter 2

An Analysis and Review of Information Retrieval Concepts

2. An Analysis and Review of Information Retrieval Concepts.

There exists a wide range of computerized systems that perform the function of retrieving information from a store of data. At one end of the spectrum are the so-called 'data providing retrieval systems' or question-answering systems. [7]. These systems, as the names imply, provide a specific fact or answer to a question. At the other end of the continuum are the reference retrieval or literature searching systems. These systems, in response to a question, provide lists of references to documents that may answer the question. The most important distinction between these two systems is the type of inference making capability that each system employs. [63]. When a query is posed to a question-answering system, a body of data is examined in order to extract one fact from the data file. The desired fact may not be present in the form needed by the user. In order to gather the required information, the question-answering system may have to deduce the answer from a number of related items of information.

In contrast to a question-answering system, a literature searching system has a very trivial inference making capability. When a query is compared to a document representation, the literature searching system infers that the document meets the users needs if the words in the representation match the words in the query.

The problems connected with developing question-answering systems are enormous. Those question-answering systems that have been implemented are operating in an experimental environment and use limited data bases. In addition most of these systems suffer from problems of high computation times and large memory requirements. (See [41] and

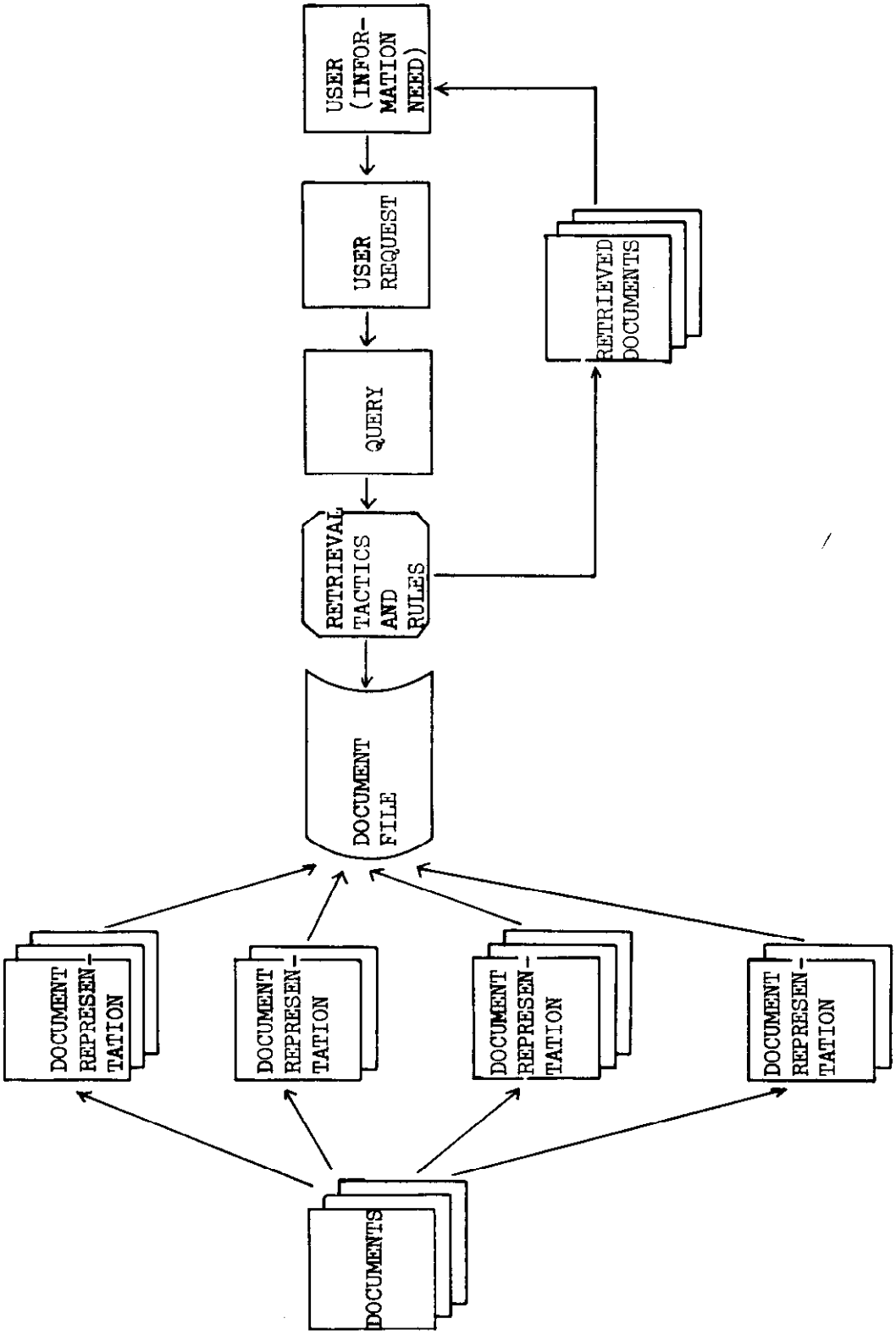
[90] for examples of such systems.) The remainder of this paper will concentrate on the analysis of literature searching or reference retrieval systems.

A model of a reference retrieval system is presented in Figure 1. The retrieval process involves searching through a file of documents to determine which, if any, documents will satisfy the user's need. Thus one end of the figure shows the initial input to the system in the form of documents in natural language. At the other side of the figure is the user with a yet unmanifested requirement for information. An automated literature searching system processes documents to determine their subject content. The content indicators are then assigned to replace the document itself and they become representations of the document. Once each document representation has been created, the representation is stored with the previously processed document surrogates in the document file.

When the user establishes his requirement for information, he expresses it in the form of a request. The next process that the user must go through is to convert that request into a form that the retrieval system can process. The converted form of the requirement is called the query. Given the query, and the document representations stored in the file, the reference retrieval system searches the file, using a search strategy or a retrieval rule to determine if there are any document representations that match the query. The bibliographic citations and/or abstracts to those matching documents are then presented to the user. Based on the results of the search, the user can decide to stop or reformulate the query and make another search.

Figure 1

The Retrieval Process



2.1 Problems in the Development of a Retrieval Theory.

There are a number of fundamental issues that need to be explored if advances are to be made in the development of improved reference retrieval systems. One problem has to do with developing in a computerized system the ability to predict what the user probably wants in the way of information. Another issue centers on designing a system capable of picking documents that meet user requirements on level of complexity (e.g. mathematical, non-technical, survey.)

What would be extremely valuable is to understand the process that the user goes through in determining his needs. If it were possible to characterize this process, the result might be to gain new insights on methods of information transfer. In order for the system to predict what a user wants, it would be necessary to have information about the user's state of knowledge, information processing capabilities and intelligence. And as more information is supplied, the system should be able to modify its representation of the state of the user's knowledge. As the system has more information and better prediction rules, it can make better inferences. The issues here are extremely complex and no simple solutions are forseen.

There is also an additional question of whether technology should be allowed to develop in such a way as to be in a position to predict what information will satisfy a user based on a previous state of intelligence. If such systems could be developed, unauthorized persons might use the systems to manipulate individual behavior by providing false data to the inquirer. Perhaps the possibility of abuse of such a theory is too great.

Rather than concentrating on the development of a theory of these underlying processes, there is a parallel approach that can be used in the development of information retrieval systems. It is possible to build systems based on tentative conjectures about retrieval rules that would lead to effective results. Then through empirical testing, the performance of such systems can be monitored.

2.2 Formal Techniques for Literature Searching.

Presently there is a lack of adequate explanations for the various phenomena involved in information acquisition and transfer between a human being and an information retrieval system. Nevertheless such systems continue to be built on the premise that experimentation will lead to the design of effective retrieval systems. These systems are developed using procedures and practices that are mechanizable or formal.

Current models of mechanized literature searching systems are composed of three principal parts. The first component performs the process of extracting content representations from documents. These content indicators are used by the systems as a means of identifying the documents. The second function involved in system operation concerns the formulation of queries. In order for document references to be supplied, a formalization of the user's needs is necessary. This takes place when the user presents a query to the system. Finally, once the content of the documents in the collection has been identified and the query formulated, methods must be employed to compare the query with the document representations. These are termed retrieval rules or a

retrieval strategy. Each of the three components will be discussed in detail.

2.2.1 Identification of Document Content.

In an information retrieval system, content analysis is used as a technique for identifying what the document is about. The estimates of the document content then become the basis for the processes of indexing, abstracting and classifying of the documents.

It is possible to distinguish at least two methods for content analysis. Syntactic methods are those that use the structure of word sequence in a sentence or phrase as a clue to whether certain words are content bearing. The statistical approach relies on the occurrences or frequency of words to select content bearing words that are good clues to document content.

2.2.1.1 Statistical Methods for Automatic Content Analysis.

An early study of the use of statistical methods in language analysis for information retrieval was conducted by H.P. Luhn. [80]. Luhn hypothesized that "... the frequency of word occurrence in an article furnishes a useful measurement of word significance." [80, p. 160]. He proposed a weighting scheme to select the sentence in the document which is the most representative of the content. He argued that the significance of words in a document was a function of the frequency with which the words occurred. This procedure of Luhn's was predicated on

his idea that there are the following three classes of words: those frequently occurring words that provide little resolving power, a group of very infrequently occurring words which also are not useful, and finally a middle frequency group that have the greatest significance for content analysis purposes.

Edmundson, Oswald and Wyllys have proposed several extensions to Luhn's work. [38]. In addition to allowing more than a single word to be used as an index term, they formulate a number of ratios of word occurrence in a document to give clues to the importance of a particular term. Two quantities are computed: The quantity 'f' is the frequency of occurrence of a word in a document, calculated by dividing the total number of occurrences of the word in the document by the total number of occurrences of all words in the document. The quantity r is the frequency of occurrence of a word in a class of documents. The authors suggest four measures that can be used to determine the significance of a particular word. [38, p. 36].

1. $s_1 = f - r$
2. $s_2 = f / r$
3. $s_3 = f / (f + r)$
4. $s_4 = \log (f / r)$.

Eight years later, Curtice and Jones reported considerable success in selecting content words from document abstracts. They hypothesize that "... words which freely occur in almost any text environment are less suited to serve as index terms than those whose environment is detectably constrained." [28, p. 152]. The authors formed the ratio

$$r_i = N_i / f_i ,$$

where N_i is the number of different words that occurred in the same

abstract with the i th word, and f_i is the frequency of occurrence of the i th word in the abstract. A regression line is then fitted to all (r_i, N_i) pairs for the vocabulary, and the distance from r_i to the line (Δr_i) is calculated. It was found through a subjective analysis that for a given term i, the sign and magnitude of Δr_i indicated whether the word was 'dispersed' or 'constrained.' Dispersed words are general terms such as 'existing,' 'purpose,' and reduced.' They all had positive Δr_i's. Constrained words are more specific terms such as 'grease,' 'boron,' and 'extrusion.' They had negative Δr_i values.

In the preceeding experiment, the statistic r_i was developed to distinguish between 'good' and 'bad' index terms. In a study conducted by Dennis [33], several such measures were developed and compared empirically. Dennis suggested that 'non-informing' words have a frequency distribution different than 'informing' words. Eight statistics were developed and tested for their effectiveness in discriminating between content and non-content words. [33, p. 65-66]. These include the absolute frequency of occurrence of a word in a text as well as the relative frequency. Subject specialists were enlisted to judge which of the distributions was best able to make the discrimination. It was found that one of the members of the Erlang family of curves induced a word ranking that coincided most closely with the judges' rankings.

Experiments of a similar nature have been conducted by Damerau [32] and Stone [127]. They both have found that the Poisson distribution is a good ordering device for discriminating between content and non-content words.

Recently Edmundson has suggested that the simple statistical approach to content analysis can be extended by using various clues in

the document. [37]. His approach attempts to isolate sentences that are most informative. Four basic methods are used:

1. The cue method of analysis relies on three dictionaries to determine if words in a sentence are relevant or not. The first dictionary contains words that would be useful content indicators, the second contains words that would definitely not be useful as content indicators, and the third contains words having neutral content value.

2. The key method uses word frequencies to identify content words.

3. The title method uses a dictionary to isolate content words from the title and headings and weights them as to their content.

4. The location method is most similar to the ideas of Baxendale in that the placement of a sentence in the text gives clues as to its importance. [8]. These methods were used in combination with one another to pick high content bearing sentences.

The statistical methods for automatic content analysis that have been presented in this section by definition all use the frequency with which words occur in text to give clues as to the extent to which the words are content bearing. In the next section an alternate approach using syntactic methods to analyze document content will be discussed.

2.2.1.2 Syntactic Methods for Automatic Content Analysis.

It is possible to observe in the information retrieval literature a peaking of interest in the use of statistical models for content analysis. This is perhaps due to a feeling that a limit has been reached with the performance of these models. Attention seems to be

focusing on the syntactic methods of language analysis. Syntactic methods take advantage of the structure of word sequence in a sentence to determine which words are content bearing.

In an excellent review article, Simmons [118] synthesizes the major tracks that have been followed in natural language research. He points out that machine translation and early information retrieval research has found itself up a blind alley because it has persisted in the use of words as the unit of meaning rather than phrases, sentences, paragraphs, etc. With one paradigm exhausted, a more global approach is being explored.

A number of methods of syntactic analysis have been examined in the past years. It is beyond the scope of this paper to delve into the methodology employed, but rather, to indicate to the reader further sources of information.

Linguistic problems are surveyed in an introductory volume by John Lyons [81]. Application of linguistic theory to the problems of information science are surveyed in a number of volumes of the Annual Review of Information Science and Technology. Attention is drawn particularly to the articles by Montgomery [91] and Salton [110] in that series.

There have been very few systems developed using syntactic methods for content analysis and indexing. This is the case despite the fact that there are now available good procedures for syntactic analysis. (For example see [66] and [71].) Among the few indexing systems employing syntax analysis are those developed by Baxendale [9], Earl [36], Montgomery [92], and Salton [111].

2.2.2 Query Formulation.

User interaction with an information retrieval system begins with the formulation of a query. The user transforms his requirements into a request. The request is then mapped into the query language of the retrieval system.

There is a wide spectrum of languages available for this communication between the user and the retrieval system. The simplest query language is one in which the user is allowed to specify a single word, and all documents that have this word as a content indicator are retrieved. At a next level of complexity are query languages that allow specification of a series of terms that must be present before the document will be retrieved. Beyond this point are query languages that permit Boolean expressions of terms to be included in a request. At present there are some languages which extend the Boolean concept to include the possibility of the user placing numerical weights on certain terms in the expression to reflect the importance the user attaches to that word versus other words in the query. [11], [88], [120]. In principle, the most complex query language is ordinary language. Designers of retrieval systems have not yet reached the point of including the facility for this form of communication between the user and the system.

2.2.3 Retrieval Rules.

Document representations are formed by applying content analysis methods to the text of a document. Retrieval rules are used to compare a query to a file of document representations. There are a number of methods that can be used by a mechanized literature searching system to perform this comparison. They are discussed in the following sections.

2.2.3.1 Matching Rules.

The simplest search rule available is a match-no-match scheme. It involves examining the terms in the query and determining if there are document representations that contain these terms. Those representations that match the query are presumed to be relevant to the request and are retrieved. Those representations that do not match are considered not to be relevant to the request and are not retrieved. The effect of this strategy is to divide the library into two parts: those documents that are relevant to the request and those that are not.

The next extension of the above rule is to try to measure the degree of the match between the query and the document representations. Then if the number of terms in common exceeds a user specified threshold value, the document reference is returned to the user.

A number of variations are possible on these basic procedures. For instance the retrieval system may do more than present to the user a list of retrieved documents. Instead the system may rank the documents to reflect the degree to which the query matched the documents. Degree

of match could be computed in a number of ways. First the ordering could take place on the basis of the number of terms matching in each retrieved document. Alternatively it is possible to compute a measure of distance between the query and document and order the retrieved documents on the basis of the value of the distance measure. Appendix 1 presents some distance measures that could be used.

As an example of the application of distance measures to the ordering process, consider the case where documents are represented as points in n-dimensional index space and a query is also represented as a point in the same space. Then a measure of distance between a document and the query could be the angle between the vectors formed by connecting the query point and document point to the origin. [107].

2.2.3.2 Associative Searching.

If the user had the time or inclination he could probably attain the best results by formulating a query and then manually looking at each document in a store to see if it was in any way related to the query. The user would then engage in an intellectual process which would result in his examining each document in hopes that there would be some direct or indirect relation between the query and the document. The process of associative searching is a formal attempt to generalize the searching procedure. [47], [86].

When associative retrieval is employed, the user submits a query to the retrieval system and the terms in the query are augmented by terms highly related or associated with the original query terms. Then the

augmented query is used to search the document file.

In order to implement associative retrieval, it is necessary to form an association matrix which reflects the strength of the statistical relation between all pairs of terms in the vocabulary. The procedure for generating an association matrix begins by the formation of a document-term matrix. The document-term matrix has as many columns as there are terms in the vocabulary. Then a given document is represented by a row vector indicating which terms are present in the document. Given the document-term matrix, a term-term association matrix can be computed by a number of formulae. Appendix 1 surveys these measures. Augmentation of the query terms is accomplished by consulting the association matrix.

2.2.3.3 Clustering.

The methods of matching and associative retrieval assume that a physical search of a document file or index will encompass all documents assigned any of the index terms in the query. The exact method of physical searching will of course depend on the file structure employed. (See Section 2.3). However, the methodology of clustering is available to pre-group logically related documents and thus minimize the number of documents that need to be searched for a given query. Instead of determining closeness of index terms as is done in associative searching, the closeness of documents is determined.

The objective of clustering is to put objects with similar characteristics into one group, with the result that the objects in a given

group will be more similar to each other than to other objects not in that group. In effect it is desired to minimize within-group variance and maximize between-group variance.

There are numerous methods for clustering. For example see [4], [6], [30], [31], [50], [83], [85], [96], [97], [98], [123], [137], and [139]. Rather than discussing each of these methods and their relation to the retrieval process, a number of clustering principles will be examined.

Several different types of classification schemes can be constructed. [121]. The numerical taxonomist's distinction between monothetic and polythetic schemes is particularly applicable. A monothetic classification system is one in which an item (e.g. a document) must have a specific set of representations in order to belong to a given cluster or class. In contrast to a monothetic classification system, a polythetic classification scheme requires that an item have certain characteristics before it can be considered as belonging to the cluster or class. However the item does not have to have all the characteristics in order to be a member of the class when using polythetic classification methods.

Still different ways of classification are available. [122]. Objects can be grouped together on the basis of overall similarity (phenetic classification), on the basis of similarity at a given point in time (chronistic), or common lines of descent (cladistic). The reader is referred to [122] for further proposals along this line.

Once a framework for the clustering has been established, it then must be decided how to select the characteristics on which classification will depend. In the case of a document, it should be established

what the content indicators will be (e.g. index terms), how many of them there will be, and what rules will be imposed on their selection. Given the object identifiers, it must be decided how to measure the similarity between them. Many possibilities are open including measuring Euclidian distance, Hamming distance, using association measures (See Appendix 1) or correlation coefficients.

There are two principle algorithms that can be employed to perform the actual clustering once the attributes have been selected and the distance measure established. [82]. The devisive method of clustering begins with the entire set of objects as one cluster and successively divides this cluster into a number of smaller clusters. An alternate approach is to assume that each object is a clump by itself. Then the procedure is to look for clumps that can be combined because of their similar characteristics. Some clustering algorithms use a combination of these approaches. But no matter which method is used, the distance measure is used to determine the homogeneity of the clusters, and thus via a threshold control membership in the clusters.

The problem of when to terminate the clustering process is very difficult. There are a number of approaches to selecting a stopping rule, but intuition and experience are valuable adjuncts. One possibility is to continue clustering until no clusters change with respect to their membership on further computation. Clustering could also be terminated when the average within or between cluster variance reaches a certain level or continue as long as the variances continue to decline.

At this stage of development, clustering of large files of documents appears to be a very expensive method to be used in a retrieval system. Beside the fact that these methods require a large computer

with a fast instruction speed to handle moderate sized problems, some of the algorithms do not yield a unique clustering pattern for the same data and some are dependent on the order in which the records are processed by the system.

2.2.3.4 Feedback Methods for Searching.

One of the most promising approaches in the development of techniques for searching a document file is the use of feedback methods. The retrieval process involves comparing a query to a number of document representations. The methods by which this can be accomplished are numerous and have been described previously in this chapter. The feedback methodology is applicable to many of these searching techniques.

A number of methods have been proposed by members of the Smart project at Harvard University and (later) Cornell University for improving search performance through the use of a dialog between the user and the system. An early model of user feedback was developed by J.J. Rocchio. [107].

Rocchio's model assumes that a document can be represented by a vector in an n-dimensional index term space. Similarly, a query is represented by a vector in the same space. The process of retrieval involves finding the document or documents that minimize the angle between themselves and the query vector. Once an initial set of documents has been retrieved, the user determines which are relevant to his request and which are not. This information is then used by the algorithm to reformulate the query. In general the n+1 st generation query

(q_{n+1}) is a linear combination of the n th generation query (q_n) and the vector difference between the sum of the relevant (d_r) and non-relevant (d_m) documents retrieved in the n th iteration. That is

$$q_{n+1} = \alpha q_n + \beta[\sum d_m - \sum d_r]$$

where α and β are constants. This formulation attempts to form a query which will minimize the angle between the query and relevant documents and maximize the angle between the query and non-relevant documents.

In a later Smart report, Riddle, Horwitz and Dietz propose another query modification model. [106]. Let R represent an m x n matrix of n retrieved documents each with m possible index terms associated with it. Also define W as an n x 1 vector containing a numerical value reflecting the relevance of the n th retrieved document to the original query Q. Then the query modification procedure can be written as

$$Q_{n+1} = Q_n + \alpha RW.$$

Here Q_{n+1} is the modified query and α is a multiplier which controls the extent to which the original query is modified.

Eleanor Ide has extended the two previous search feedback methods by proposing an even more general model: [56].

$$Q_{n+1} = \pi Q_n + \omega Q_0 + \alpha \sum_{1}^{\min(n_a', n_r')} r_i + \mu \sum_{1}^{\min(n_b', n_s')} s_i$$

In this equation Q_0 is the initial query and Q_n the previous query. The quantities r_i and s_i represent relevant and non-relevant document vectors. There are n_r' relevant documents retrieved and n_s' non relevant documents retrieved. The 'min' function is used in the summation of the relevant

and non-relevant document vectors to allow some quantity (n_a', n_b') less than the maximum number of relevant (non relevant) documents to be used to modify the query. The constants π, ω, α, and μ are used to weight each component in deriving the new query.

It can be observed that the three methods described above are basically similar in their approach. Searching begins with an initial scan of the collection to determine which documents satisfy the query. With a retrieved set of documents on hand, the user then indicates which are relevant. This information results in the modification of the original query or in some cases the creation of two queries out of the original query. [16]. The search is repeated using the new query in hopes that more relevant documents will be retrieved. The entire process can be repeated until no new relevant documents are retrieved or until the user is satisfied with the results so far obtained.

2.2.4 The Thesaurus.

The thesaurus is a potentially useful device as an indexing aid and as a retrieval tool. In this section the role of the thesaurus in an automated information retrieval system is discussed and techniques for automatic thesaurus construction are outlined.

One definition of a thesaurus is the following. "An information retrieval thesaurus is a term-association list structured to enable indexers and subject analysts to describe the subject information of a document to a desired level of specificity at input, and to permit searchers to describe in mutually precise terms the information required

at output. A thesaurus therefore serves as an authority list and as a device to bring into coincidence the language of documents and the language of questions." [133, p. vii].

The process of indexing can be considerably assisted by the use of a thesaurus. The indexer determines the meaning of the document, translates this assessment of the content into index terms, and assigns a subset of those index terms to the document. The thesaurus aids the process in a number of ways. When used as an authority list, it indicates which terms may or may not be used as index terms. In addition, it helps refine the indexer's choice of terms so that the final set of terms is as broad or as specific as the indexer desires.

A second principal function of the thesaurus is as an aid to the retrieval of documents. As Stevens puts it: the thesaurus in this capacity is attempting to provoke the user into selecting suitable terms. [124, p. 114]. The objective of such a dialog is to transform a user query expressed in an uncontrolled vocabulary into a query incorporating terms that the system uses.

The process of using a thesaurus as a search aid can be as simple or complex as desired. For an on-line system, the user might sit at a console and enter his query. The system could then examine each word of the query to see if it was present in the thesaurus. If a word was not present, the system could give the user the option of changing or deleting it. If the word was present, the hierarchy of terms around the chosen term might be displayed. This would give the user a feel for the manner in which the system uses and recognizes the term in question.

There are a number of problems involved in the automatic construction of a thesaurus and in reality these are the problems of language analysis

in general. Algorithms for automatic thesaurus construction must be able to detect synonyms, analyze homographs, determine syntactic equivalences, recognize indirect references in language and incomplete relations between words, and finally detect word meaning changes over time. [111, p. 22].

Several models have been developed to deal with the problem of synonym detection. Rubenstein and Goodenough attempt to show the relation between the context of words and their synonymy. [109]. They suggest that if it can be established that words that are synonyms appear in similar contexts, then this information can be used to detect synonymous words. Two statistics are developed which are intended to detect when in fact word contexts are similar: One statistic uses frequency of word types and the other uses frequency of word tokens.

Another approach to the problem of automatically detecting synonyms and antonyms was proposed by Lewis, Baxendale, and Bennett. [78]. They hypothesized that if two words are synonymous they will seldom co-occur in the same sentence "... but in their seperate occurrences they tend to have similar contexts." [78, p. 21]. The authors also consider that both an alpha and a beta error can occur in such an analysis. Several quantities are used to determine synonymy:

x_a, x_b	A pair of words.
n_{ab}	The frequency of occurrence of the word pair (a,b).
D_a	The number of words which have occurred in at least one sentence with x_a.
P_{ab}	The number of words which have not occurred in any sentence with both x_a and x_b but which have occurred in some sentence with x_a and in some other sentence with x_b.
J_{ab}	The number of terms x_i which have occurred at least once in a sentence in which both terms x_a and x_b also occur.

The hypothesis being tested can then be restated more succinctly. "If a pair of words, x_a and x_b, is synonymous, then n_{ab} will be small, perhaps zero; and the pairwise context, P_{ab}, will be large relative to both D_a and D_b. The condition of n_{ab} being small also implies that P_{ab} will be large relative to J_{ab}." [78, p. 25]

Once synonomy has been established the remaining problem is to form the words into a structure. The techniques of clustering appear to have applicability to this problem. (See Section 2.2.3.3). However some methods for hierarchy formation have been developed which are directly applicable to the thesaurus construction problem.

Consider a thesaurus as being represented by a graph whose vertices correspond to terms and edges correspond to term-term semantic associations. [1], [2]. Then synonym relations are expressed by a symmetric pairwise relationship on the graph and hierarchical relations are non symmetrical. Algorithms can be constructed to decompose the graph so that mutually exclusive categories are formed. This has the effect of clustering terms based on their graph relations. If a complete subgraph can be formed, then the method has in essence detected synonyms. If partially complete subgraphs can be detected, then these indicate incomplete semantic relations that should be manually investigated. [2, p. 137].

Using the graph model, Salton has devised an algorithm (adapted from Abraham [2]) that produces a hierarchy. The method involves determining how pairs of terms are related. Relatedness is calculated by measuring the extent to which the pairs occur in the same documents. Four relations are possible: the terms are not related as measured by the similarity coefficient, term A dominates term B as measured by the

similarity coefficient, term B dominates A, or A and B have similar weights. [111, p. 60]. This information is then used to determine, for all pairs of words, where in a hierarchy the terms belong relative to one another.

Automatic construction of a thesaurus seems a long way off. Current methodology allows for computer assistance in the clerical processes of thesaurus construction such as checking for valid cross referencing and consistent usage of terms defined to be broader or narrower than a given term. The crux of the problem is that of determining relations between terms. Clustering techniques seem to have applicability here as do content analysis methods. Very crude algorithms are now available for forming a thesaurus automatically. But much work remains to be done in this area.

2.3 File Organization.

This chapter has concentrated on issues related to the development of a theory of information retrieval and subsequently on an analysis of mechanized procedures that are currently being used. In general, these problems address the question of the 'goodness' of the access provided to the user of the literature searching system. There are, however, another set of issues that must be analyzed if it is desired to fully evaluate retrieval systems. These considerations have to do with the cost and efficiency with which document representations are scanned and retrieved from a mechanized system.

Once documents have been received at an information center and

converted to machine readable form, then automatic content analysis can be performed. Automatic statistical and/or syntactic analysis will result in candidate terms being selected as possible index terms. Then perhaps with the aid of a thesaurus and various mapping rules, the index terms can be selected from the candidate list and assigned to the document. The remaining task is to store the document, abstract, index terms, bibliographic information, etc. in a file for retrieval. This section discusses methods of organization of documents in a computer storage device. A unifying model of a document retrieval system can not ignore the problems of file organization for they are intimately related to the problems of searching a collection of documents.

A file is composed of a number of records. Each record in turn is made up of one or more fields. A file structure is an ordering or arrangement of the records. The most simple file structure is a sequential organization. The properties of this structure are such that in order to find one particular record in the file it is necessary to scan each record in turn until the proper record is found.

If one considers a set of card catalog drawers in a library, it is possible to make an analogy between that set and a hierarchical or indexed sequential data structure. To locate a particular catalog card, a scan of the labels on each drawer is made. When the proper drawer is found, it is opened and a scan of the index tabs is made to locate the area in the drawer containing the desired bibliographic record. Finally a sequential scan is started from the selected index tab to the desired record.

Another commonly used file structure is known as random or direct organization. Two concepts are important for an understanding of this

access method. The first is that of a key. When a search is conducted of, for example, an author-title card catalog for a specific author, the operation involves searching for a key in the file that matches the key (author name) that is of interest. The second concept concerns the physical storage location of a record. In the card catalog, a particular card can be located by specifying the drawer number and card number within the drawer. Similarly in a disk or drum storage device attached to a computer, information can be located in terms of cylinder number and track number. A random or direct file organization technique is one that organizes and locates records on the basis of a transformation between the logical key and the physical storage location. (See [19] and [58] for examples of this type of transformation). By performing certain types of operations on the author name, that key can be converted to a physical location on a disk or drum. Then to retrieve the record, it is necessary to go to that physical location and read the record.

A file organization scheme closely related to the indexed sequential method is an inverted structure. In a simple indexed sequential structure there is one key per record which identifies the record. An index is constructed which contains pointers to records having the specified key. The records in the file are stored sequentially in key order. An inverted file structure stores its records in order by a sequence or accession number. When the file is constructed, a determination is made as to which fields of the records will be indexed. The index then contains as entries a list of unique fields in the file of records. Corresponding to each entry are the accession numbers of the records that contain that index term.

The chain or list structure is another method for file organization.

Consider the case of storing bibliographic records such that some logical relation between subjects, authors, and titles of books is preserved. For example, given a particular subject, a logical chain would be formed that would lead from that subject to every author in the file who had written on the subject. From a particular author another logical chain could be constructed to point to each book on the subject that the author has written. By linking logical records with related attributes together, a chain is formed. The linking takes place by storing in each record the address of the next record and/or preceeding record in the chain.

A number of hybrid organization schemes stemming from these four basic techniques are in use. For example, the Multilist system [77] and [102] uses both a hierarchical and list structure combined with access to the file by more than one key. Chapin [22], Senko [113], and Rettenmayer [105] review other combinations.

For each of the file organization possibilities, the system designer has a number of factors to consider in selecting one for an information retrieval system. File creation time, file maintenance time, and access time are all important. Some methods will require more space than others to store the same number of records because of the need for indexes and pointers. Problems connected with a rapidly expanding file will have to be considered in the context of record overflow and file reorganization. And the way in which the file will be searched (single or multiple key, sequentially or randomly) will influence the selection.

Analytic tools are now available to aid in the selection of a file design. With the analyst supplying the computer configuration and file requirements, formulae are available to compute memory requirements,

access time, search time, etc. (See [51], [55], [77], [89], and [116] as examples).

The emphasis in this chapter has been on exploring many of the problems connected with the development of retrieval systems. It has been noted that although there is yet no comprehensive theory useable in the development of information retrieval systems, an alternate approach of mechanizing certain retrieval processes is underway. In the next chapter various methods for evaluating the effectiveness of these systems are presented.

Chapter 3

Methods for Evaluation of Literature Searching Systems

3. Methods for Evaluation of Literature Searching Systems.

There are a number of approaches that can be used in the evaluation of retrieval systems. Three alternatives are explored in this chapter. The traditional methods using measures of retrieval effectiveness are presented first. In succeeding sections cost methods and simulation methods for evaluation are described.

3.1 Measures of Retrieval Effectiveness.

Much of the research connected with the evaluation of retrieval systems has centered on the development of measures designed to reflect the performance of a retrieval system. To a great extent these measures are all based on the contingency tables shown in Figure 2. [130, p. 245-246]. In the Figure, 'a' designates the number of documents for a given request that are both relevant and retrieved; 'b' - the number of documents retrieved but not relevant; 'c' - the number relevant but not retrieved; and 'd' - the number not retrieved and not relevant. Figure 2b shows the corresponding costs and values for each of the quantities.

The two measures used most frequently in retrieval system evaluation are the recall and precision ratios. Recall (R) is defined as the ratio of the number of documents both relevant and retrieved to the total number of relevant documents in the collection. That is,

$$R = a/(a+c).$$

Precision is then the number of documents both relevant and retrieved divided by the total number of retrieved documents,

Figure 2

Retrieval Effectiveness Contingency Tables

	Relevant	Not Relevant	
Retrieved	a	b	a + b
Not Retrieved	c	d	c + d
	a + c	b + d	a + b + c + d

Figure 2a
Number of Documents in each of Four Categories.

	Relevant	Not Relevant	
Retrieved	V_1	K_1	
Not Retrieved	K_2	V_2	

Figure 2b
Value (V) and Cost (K); per Document Falling into each of the Four Categories.

$P = a/(a+b)$.

In the framework of these two ratios, the objective of an information retrieval system is to maximize both recall and precision over a large number of queries. High recall implies that the system "... rejects very little that is relevant but may also retrieve a large proportion of irrelevant material, thus depressing precision. High precision, on the other hand, implies that very little irrelevant information is produced but much relevant information may be missed at the same time, thus depressing recall." [112, p. 213].

A number of other simple ratios have been proposed and they are presented in Table 1 using the standard notation of Figure 2.

All the ratios that are calculated from the quantities of the contingency table of Figure 2 suffer from a number of defects. By far the most serious problem is that there is no adequate theoretical basis for selecting one measure over another. A further difficulty has to do with the dichotomy forced by the contingency table. The effect is that either a document is relevant or it is not. There is no middle ground. Still another more practical problem comes into play when the measures must actually be used. For example, how can one measure in a large corpus the number of documents relevant but not retrieved for a given query?

It has been noted previously that some retrieval systems have the ability to present ranked lists of documents to the user in response to a query. Several retrieval measures have been developed to evaluate these procedures. They include the normalized recall, normalized precision, rank recall, and log precision measures formulated by Salton [111, p. 283-293] and the expected search length measure developed by

Table 1

Measures of Retrieval Effectiveness

Name / Author	Standard Notation	Authors' Notation
Generality Ratio Cleverdon [25]	$\frac{a+c}{a+b+c+d} \cdot 1000$	$\frac{1000C}{N}$
Concentration Ratio Fairthorne [40]	$\frac{a+c}{a+b+c+d}$	$\frac{C}{N}$
Non Relevant Doc. Ratio Mooers, Fels [42]	$\frac{b}{b+d}$	–
Specificity Rees [103]	$\frac{d}{b+d}$	$\frac{d}{b+d}$
Resolution Factor Perry [101]	$\frac{a+b}{a+b+c+d}$	$\frac{m}{n}$
Elimination Factor Perry [101]	$\frac{c+d}{a+b+c+d}$	$\frac{m-n}{n}$
Noise Factor Perry [101]	$\frac{b}{a+b}$	$\frac{n-w}{m}$
Omission Factor Perry [101]	$\frac{c}{a+c}$	$\frac{x-w}{x}$
Distillation Fairthorne [40]	$\frac{ad-bc}{(a+b)\ (c+d)}$	$\frac{RN-CL}{L(N-L)}$
Discrimination Fairthorne [40]	$\frac{ad-bc}{(a+c)\ (b+d)}$	$\frac{RN-CL}{C(N-C)}$

W.S. Cooper. [26].

The objective of the Salton measures is to compare the ranks of the relevant documents obtained for a particular query to the rankings of an ideal set of relevant documents. An ideal ranking of five documents is just the ranks (i=) 1, 2, 3, 4, and 5. The actual ranking produced by the system for the five documents might be (r_i=) 3, 5, 6, 11, and 16. Then, by determining the area between the two rankings for n relevant documents

$$\sum_{i=1}^{n} r_i - \sum_{i=1}^{n} i ,$$

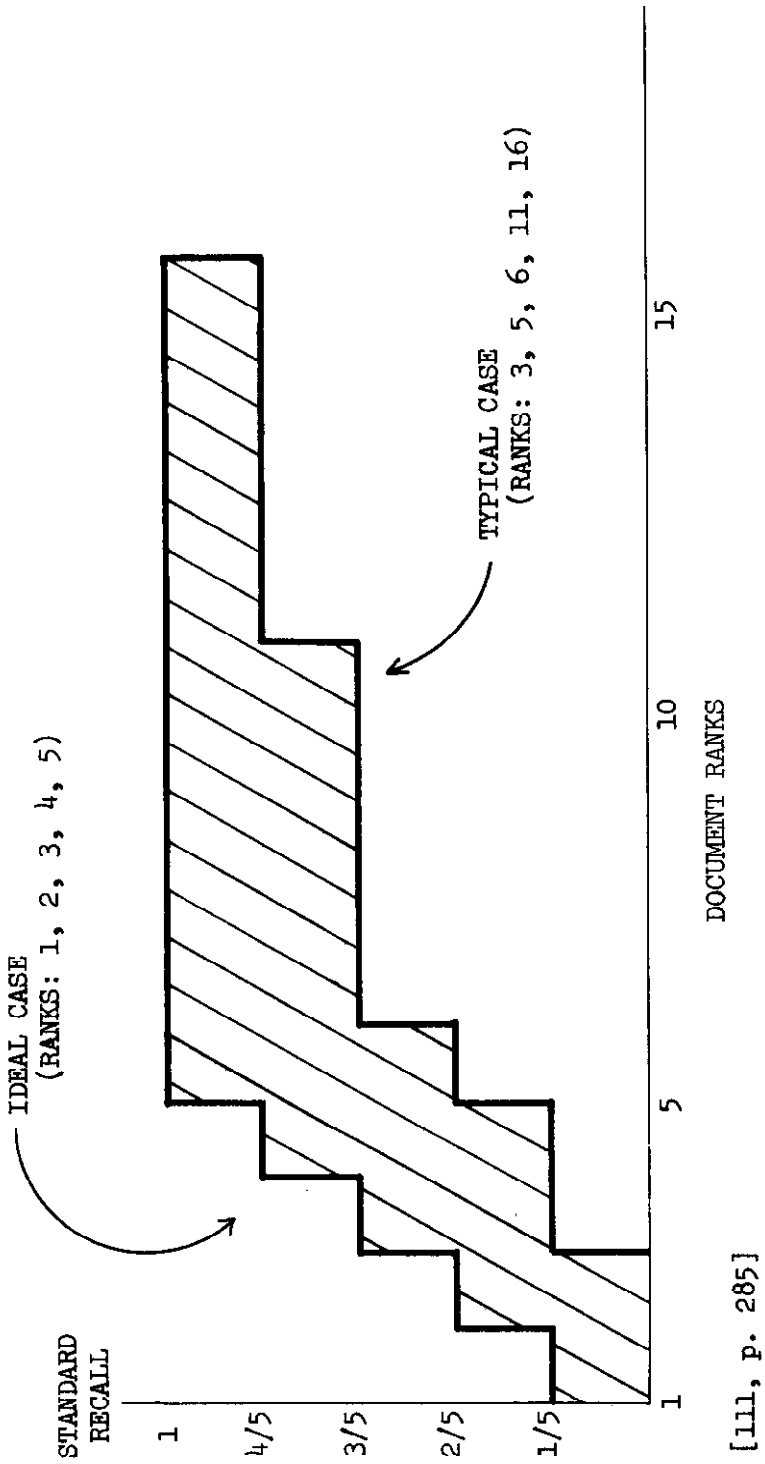
a measure of performance of the system is obtained. (See Figure 3.) The normalized and ranked measures are modifications of this basic relationship.

W.S. Cooper's expected search length measure appears to be similar in some respects to those developed by Salton. Cooper is attempting to determine the number of irrelevant documents that must be scanned in order to attain a specific set of relevant documents. In the formulae allowances are made for measuring search effort for various kinds of requests such as a specific answer to a question or the need for n relevant documents. One difference between the measure and Salton's is that the expected search length formulae allow for the possibility that there may be a number of documents having similar relevance values, and thus a linear ranking is not possible.

It seems apparent that one cannot be content with evaluation based on the simple ratios presented so far. As Swets [130] has noted, none of the ratios completely characterize the system being evaluated. They deal with one part or several parts but are not in any sense all

Figure 3

Differences in Document Ranks



[111, p. 285]

encompassing. Several broader measures of retrieval effectiveness have been proposed, such as combining recall and precision into one measure. [103], [111], [134]. But this seems an inadequate solution. Table 2 summarizes some of these proposed measures.

Table 2a

Overall Measures of Retrieval Effectiveness

Name / Author	Standard Notation	Authors' Notation
Effectiveness Rees [103]	$\frac{a}{a+c} + \frac{d}{b+d}$	$\frac{a}{a+c} + \frac{d}{b+d}$
Effectiveness NSF [134]	$\frac{a}{a+b} \cdot \frac{a}{a+c}$	-
Composite Measure Salton [111] (1)	-	$\frac{\sum_{i=1}^{n} i}{\sum_{i=1}^{n} r_i} + \frac{\sum_{i=1}^{n} \ln i}{\sum_{i=1}^{n} \ln r_i}$
(2)	-	$1 - 5(R_{norm}) + P_{norm}$
Measure of Merit Verhoff [136]	$V_1 a - K_1 b - K_2 c + V_2 d$	$\alpha\lvert AP_1\rvert - \beta\lvert A\overline{P}_1\rvert - \gamma\lvert \overline{A}P_1\rvert + \delta\lvert \overline{AP}_1\rvert$
Effectiveness Goffman [48]	$V_1 a - K_1 b$	$\alpha\mu_I(A) - \beta\mu_{I'}(A)$
Value Function Good [49]	$\alpha(\sqrt{a} - \sqrt{c}) - \beta b$	$\alpha(\sqrt{n_R} - \sqrt{n_{RD}}) - \beta n_{IS}$

Table 2b

Overall Measures of Retrieval Effectiveness

Name / Author	Standard Notation	Authors' Notation
Efficiency Swets [129], [130]	$\frac{M_{f_p}(z_i) - M_{f_{\bar{p}}}(z_i)}{\sigma_{f_p}(z_i)}$	same
Search Effectiveness Dale [29]	$\frac{\sum_{j=1}^{n} \frac{A_j}{j}}{N}$	same
Retrieval Score Swanson [128]	$R - K_1 I$ $I = (a+b) - (a+c)R$	$R - \rho I \quad (I = N - LR)$
Retrieval Score Borko [15]	$R - K_1 \frac{b}{a+b}$	$r - \rho i$ $r = S/T;\ i = M/N$

3.2 The Concept of Relevance.

In the preceeding discussion of measures of retrieval effectiveness, it was shown that in the process of evaluating retrieval systems it must be determined whether the material retrieved by the system was or was not relevant to the user's need. In this section the concept of relevance is discussed as it pertains to the evaluation problem.

What is relevance? It seems most reasonable that the relevance of a document to the user is based on the judgment that the user makes about the satisfaction of the document relative to his unmanifested information need. However, this comparison seems difficult to measure. As an alternative it is usually proposed that relevance is measured between the user need and the document. Some mechanical systems attempt to quantify the relevance relation by computing the similarity between the query representation and the representations of the documents. The view taken in this thesis is that the computed similarity between weighted terms in the query and document can not be considered as a measure of relevance. It can be considered as an extremely crude approximation.

A number of definitions of relevance have been offered in the literature. Logical relevance is explicated in terms of conditional probabilities and is used when one has an hypothesis which is to be confirmed or denied using certain evidence. Consider the probability of C given A or $P(C|A)$. Then if the probability of C given A and B, $P(C|A\&B)$, is greater than $P(C|A)$, it is said that B is logically relevant to A. [21]. This concept of logical relevance is not applicable to information retrieval systems analysis. The relation that is of

concern is the relevance of a document to a user, not the relevance of evidence to an hypothesis.

Another definition of relevance is that conveyed in the concept of satisfaction. [140]. When a document is relevant in this sense, it provides satisfaction relative to a need. Satisfaction can then be measured in an arbitrarily selected unit scale. Thus with this definition the concept of 'relevance' is replaced with that of 'satisfaction relative to a need.' (As obvious as this definition may seem, it has been argued by some authors that relevance is a property of a document alone. It has been noted that this approach is similar to asserting that knowledge exists independent of a knower, or perception independent of a perceiver. This issue is discussed in [27, Volume I, p. 23]).

Very little empirical work has been done in the area of defining more precisely what a user does when he decides whether a document is relevant to his needs. A particularly valuable study in this area was performed by Cuadra and Katter. [27]. They isolated six types of variables that they believed influence relevance judgments. These include focusing variables which tend to orient the judge of the relevance to the correct frame of reference for making a relevance decision; delimiting variables which indicate what kind of judgment is to be made (e.g. degree of relevance vs. probability of relevance); situational variables which help the judge perceive the relation of his judging activities to the environment; stimulus materials variables which have to do with style, credibility, specificity, etc.; individual difference variables such as the knowledge, experience, and susceptability to bias of the judge; and finally the scale form in which the relevance judg-

ments are made. [65].

Obviously the process of making relevance judgments is complex, and consequently a full understanding of it will take time to develop. However, Cuadra and Katter's report seems to indicate the direction in which work should be performed in order to model the process. The importance of such research should not be underestimated since the concept of relevance is one of the most important to be explored by the information scientist.

It was mentioned earlier that if it were possible to predict user satisfaction with a particular document, then the efficiency of the information transfer process could be considerably improved. Such an approach might require that a computing system know the user's state of knowledge before supplying him with a particular piece of material and then update the state afterward. This concept, and the concept of relevance as providing satisfaction relative to a need, both suggest still another definition of relevance - an 'ideal' meaning of relevance. In this scheme, relevance would be the net gain in utility to the user of the additional information. Then in order to determine whether the user was more satisfied with one system than another, a cost of each unit of utility could be assigned and the comparison made on a cost basis.

In reviewing the measures of retrieval effectiveness and concepts of relevance that have been developed in the literature, it should be clear that the evaluation framework has been relatively narrow. In succeeding sections in this chapter it will be suggested that cost and simulations methods may provide the wider perspective needed for comprehensive evaluation of retrieval systems.

3.3 Cost Methods for Retrieval System Evaluation.

There are a number of stages that a system goes through in its development. Initially a new laboratory prototype is built only with the objective of determining if it will function at all. Then once the system has proven its capability, a production model is developed. Finally the design of the production model is changed continually over time to improve its performance.

The development of methods for evaluating retrieval systems has paralleled that of the development of the systems themselves. In particular, the use of measures of retrieval effectiveness for evaluation has provided a limited but useful approach to the problem. Because of the fact that retrieval system designers are now taking a more global view of the evaluation question, the techniques to be used in the process will have to encompass a broader perspective than before.

A fundamental notion in the analysis of systems is that of a system composed of a number of components that work together toward an overall objective. [24]. One begins the analysis by defining the system and its scope. Next, the objectives of the system are determined, and the constraints on system operation are established. With the objectives, constraints and preliminary system definition in mind, the analyst is in a position to begin examining the components of the system. This is done with the hope of establishing, if possible, sub-objectives and sub-constraints for each of the subsystems. In the case of the retrieval system, its components include the content analysis procedures, thesaurus construction routines, and search strategy algorithms. Each of these should be analyzed seperately to see if it is achieving its

objective. Then an overall analysis can be conducted to see how well the system as a whole performs. It is suggested that cost methods for evaluation may be particularly valuable in this analysis. When used in conjunction with the various effectiveness measures, the labor, space, overhead, computer, etc. costs can provide a valuable insight into the cost-benefit relationship in the operation of an information center.

Cost accounting methods are not new to libraries. For some period of time there has been interest in determining the cost of activities such as the acquisition and cataloging of material, as well as the cost of answering reference questions. [74], [76], [100]. Analytic models of library processes are also beginning to be developed. [12], [60], [75], [79], [93], [138]. What is lacking so far is the development of analytic and cost models for the analysis of literature searching systems rather than libraries.

Cost methods for evaluation of retrieval systems are quite varied. For example, they may involve computing the cost per retrieved citation, the cost of indexing a document, or these costs in relation to a measure of retrieval effectiveness. [73, pp. 160-180]. Alternate approaches involve computing the expected cost of the system in terms of start-up, maintenance, query preparation, computer operation and output costs. [67]. In Chapter 4 of this paper an alternate cost model is developed.

3.4 Simulation as an Approach to Retrieval System Evaluation.

The two methods for evaluation that have already been discussed - measures of retrieval effectiveness and cost methods - both have certain desirable and undesirable features. The measures of retrieval effectiveness allow comparison of the adequacy of the searching facility in delivering relevant documents to the user. They do not allow measurement of any of the time and cost variables involved in the process, and this is where cost methods of evaluation are important. There are numerous variables in a retrieval system and there are numerous subsystems within a retrieval system. An ideal measurement technique should be able to monitor all subsystems and all variables and detect and predict significant changes in the performance of the system. It is suggested that the methodology of simulation may be useful for the purpose.

In this section the concepts of simulation are discussed and a review of the applications of simulation to retrieval system evaluation is presented.

3.4.1 Simulation Concepts.

There are a number of definitions that have been proposed to characterize the concept of simulation. For example, Naylor et. al. [95, p. 2] present two possibilities. The first is Churchman's formal definition of simulation.

> ..."x simulates y" is true if and only if (a) x and y are formal systems, (b) y is taken to be the real system, (c) x is taken to be an approximation to the real system, and (d) the rules of validity in x are non-error-free. [23, p. 12].

Here

> A formal system is a set of entities, operations, properties and relations; a set of rules for combining these; a set of rules that provide estimates of what combinations are assertions; a set of rules that provide estimates of what assertions are valid; a set of rules that provide estimates of what can be inferred from an assertion; a set of rules that provide measures of the costs and accuracy of the estimates; a set of rules that generate assertions about the "whole"; and a set of rules that provide estimates of the validity of these wholistic assertions as well as the cost and accuracy of the estimates. [23, p. 4]

Another definition is given by Shubik.

> A simulation of a system is the operation of a model or simulator which is a representation of the system or organism. The model is amenable to manipulations which would be impossible, too expensive or impractical to perform on the entity it portrays. The operation of the model can be studied and, from it, properties concerning the behavoir of the actual system or its subsystems can be inferred. [117, p. 909].

Thus in the process of simulation, properties from the actual system are identified and relationships are developed to form a model of the system. There are a number of different types of models and corresponding to each is a simulation method using that particular kind of model. For example, Ackoff suggests there are iconic, analogue and symbolic models. [3, p. 109-110]. Iconic models look like the state, object or event that they represent. In an analogue model one property is used to represent another. An example of an analogue model would be when an hydraulic system is used to represent an electrical system. Models in which the properties of the system being represented are expressed symbolically are called symbolic models. The simulation models described later in this chapter are all symbolic models.

In addition to classifying a simulation study by the type of model used, it is also possible to categorize the model using other features. Naylor et. al. [95, p. 16-19] classify simulation models as deterministic, in which the variables in the model are non-random and are not in

in the form of probability distributions; stochastic, in which at least one variable in the model is given by a probability distribution; static, in which time is not explicitly accounted for; and dynamic, in which time relationships are included. The models in Section 3.4.2 fall in a number of these categories.

An important issue that needs to be resolved when considering the use of simulation is whether there are other methods of model solution that may yield the desired result with less cost, effort, etc. In general it has been pointed out that there are several cases in which simulation is a useful tool. First is the situation in which it is desired to modify a system that can not, in practice, be modified. By constructing a simulation model, the model can be varied and the results observed. An example of such a system might be the solar system.

Another case is the one in which it may be possible to modify the system and observe the result, but the cost to do so may be prohibitive. An example of this might occur in studying the optimal design of an automobile assembly line. In both of the above cases, any type of model development, including development of a simulation model, would be helpful.

A situation in which simulation may be usefully employed occurs when the system is so complex that it can not be described in an analytic form. [95]. A related problem occurs when a complex system can be described analytically but it can not be solved analytically.

Library literature searching systems have certain characteristics that are amenable to mathematical analysis and others than can be usefully explored using simulation techniques. In the former category are the problems having to do with optimal indexing depth of a document relative

to retrieval efficiency. [17]. Another is the question of the optimal number of uses a particular term in a thesaurus should have so as to maximize retrieval effectiveness.

3.4.2 Simulation Applications in Information Science.

Simulation methodology has been applied to a wide variety of problem areas including transportation, business, education, and medicine. [57]. But very little research has been done in its application to the problems of designing and investigating information retrieval systems. The few projects that have been undertaken are reviewed here.

Simulation, as an aid to file design, has been used by Senko [114], [115] and Rettenmayer [105]. In Senko's model, detailed equations are developed of every aspect of several file organization schemes and the operation of the structures is simulated. Rettenmayer's model is designed to determine whether clustering techniques can be fruitfully applied to file organization problems. A simulation model is developed in which various file organization schemes are evaluated in conjunction with a clustering algorithm.

The analysis of user behavior in a library has been modeled by Reilly. [104]. He considers the probability that the user will avail himself of library services, and the estimated service time for a patron, as two critical variables in the simulation. Using a simple linear learning model of user behavior [20], it is shown how the variables change over time.

Aspects of retrieval systems that appear amenable to simulation

analysis include studying the efficiencies of various computer configurations and expected delay times for the response from the system. In this section a number of applications will be reviewed. Still other features of literature searching systems that could be simulated are the construction of documents and queries and the performance of various retrieval rules. In Chapter 5 a simulation model is developed which uses simulation to create documents and queries as an aid to retrieval system evaluation.

The use of simulation is not without its limitations. It has been noted previously that in the process of simulation, a model of the system is developed. The model represents an abstraction from the actual system. To the extent that the model does not accurately represent all the important characteristics of the system, the results of the simulation will be incorrect. In order to insure that the simulation model is performing properly, the researcher validates the model. The process of validation is as complex as the system itself. If the model produces incorrect or inaccurate results, the researcher must modify the simulation model to correct the deficiencies. At each stage in the feedback process it is necessary to "...balance the cost of each action against the value of increased information about the validity of an insight." [135, p. 248].

A proposed, but never completed, simulation study by Haas suggested that the library as a system can be divided into three groups of elements. These include various classes of patrons such as graduates, undergraduates, research staff, and teaching staff. The second category of elements is the library facilities. These are broadly grouped into two sets: those provided for the comfort and convenience of the patrons,

such as furniture; and library intermediaries between the patron and the materials, such as the card catalog. The third category is termed stock. This includes books, journals, maps, microfiche, etc. Given this classification, the objective of the study was to be to see the way in which the patrons use the facilities and the stock. [52].

An extensive simulation study related to the Haas work, but with a much broader base, was conducted by Nance. [94]. In his dissertation, he explored the relation between the library, the user of the library, and the funder of library services. This is done within a university environment. With the use of techniques of industrial dynamics developed by Forrester at MIT [43], a simulation model was created to investigate the effects of various policies on the library/user/funder relationship.

A very interesting simulation approach to the analysis of alternative retrieval system design configurations was performed by Blunt. [13]. His model is concerned with measuring system response time and equipment and personnel utilization. There are three components to this model: (1) an event sequence generator, (2) a sequence integrator, and (3) data analysis routines. The event sequence generator determines what events will be required in processing the generated query, determines the sequence in which the events will be performed, and calculates the time that will be used for processing each of the events. The sequence integrator processes the events to see the effect on costs and response time of parallel operation of a number of processes. The sequence integrator allocates equipment and personnel to event processing, calculates query delay times and also calculates equipment and personnel idle time. The data analysis programs synthesize the results of the simulation run.

Bourne and Ford have also used simulation to evaluate alternative

configurations, but their methodology was not as sophisticated as that of Blunt. The Bourne-Ford model [18] simulates the operation of an information center for a specified time interval to determine expected operating costs, the amount and type of equipment required, and the number and type of personnel needed. The objective in the model is to choose between alternative configurations and determine the sensitivity of the performance to various parameter changes. Input to the model is time and cost data and the interrelationship between functions.

The Performance Simulator developed by Hertz et. al. in 1962 [54] is similar in concept to the model developed by Blunt. This simulator is designed to evaluate various configurations of an information retrieval system to see the effect on cost and on response time.

The simulator begins by generating user requirements. These are statistical descriptions of user needs. Next these requirements are analyzed and the system search strategy needed to satisfy the need is established. Then the simulator creates a statistical equivalent of the document file to be searched. In the final phase, the simulated file searching is performed, and material is selected to be returned to the user.

While Blunt, Hertz, Nance, and Reilly have looked at interactions of the components of information systems on a large scale, Fried et. al. have examined the feasibility of simulating the indexing of a document collection. [44]. The model explores the effect of controlling the number of times a term can be assigned to documents in the collection before reindexing is necessary. Also considered is the effect of varying the depth of indexing on retrieval. The simulation is undertaken for a collection that is continually growing, so that reindexing can be

examined. No results or conclusions are presented in the report.

It can be observed that in all these simulation studies, little has been done to apply the methodology to the evaluation of literature searching systems. So far most of the effort has been concentrated on simulating the performance and cost of information systems in terms of alternative equipment configurations and response time to queries.

The possible applications of simulation to the analysis of literature searching systems are large. In Chapter 5 a simulation model that deals with one aspect of such systems is developed and analyzed.

Chapter 4

A Cost Model of a Literature Searching System

4. A Cost Model of a Literature Searching System.

One method whereby retrieval systems can be evaluated is through the use of costs. In order for accurate comparative analysis to be conducted, the cost methods used must be consistent and comprehensive. The model developed in this chapter has two facets. It develops equations for the total cost of operating the system and thus allows comparative evaluation between other systems. Further, once the cost equations for the system have been presented, it shows that an optimal division can be made between those functions that the user should perform and those that the system should perform in optimizing the search process.

4.1 Overview of the Model.

Most information retrieval systems that have been designed to date use a technique of comparing a query representation with stored document representations in order to retrieve documents that satisfy the request. Those documents whose representations are 'closest' to the query are retrieved. It is hypothesized that this type of comparison process is not sufficient to insure the 'best' operation of the retrieval system. Not only should the system perform matching but it should take into account:

1. The cost of the search to the system. This implies that the system provides a service to the user at a specifiable price.

2. The cost of the search to the user. This suggests that the user places a value on the time and effort he spends using the system.

This quantity is in addition to the amount that must be paid to use the system.

3. The benefit that the patron can gain by using the system.

4. The most economic division of effort between the user and the system in accomplishing the user's search objective.

In this model, the user formulates a query and the system decides, on the basis of control parameters and previous experience, the retrieval rule most likely to yield documents relevant to the user's request.

4.2 Retrieval Activities.

Acquisition of information from an automated retrieval system involves an interaction between the user and the computer. As with any man-machine interaction, the more demanding and more sophisticated the user is in his requests, the greater will have to be the system effort to achieve the desired goal. In this section the trade off between user and system effort is explored and a schema is developed for analyzing it.

It is possible to distinguish three stages in the interaction of a user with the retrieval system. The process begins with pre-search activities. For the user this involves determining what is to be asked of the retrieval system and mapping the request into the system's formal query language. Since it is unlikely that the user will enter the correct query the first time he tries (perhaps due to syntax or spelling errors) there will be some user-system dialog involved in putting the request into a form acceptable by the system.

Query negotiation may be a simple process of correcting syntax, as

noted above, or can be more elaborate. For example, the system may be in a position to aid the user in query formulation by the use of a thesaurus and/or a word frequency list. The thesaurus is employed to tell the user the generality or specificity of the words that are present in the query. This allows the user to broaden or narrow the scope of the query depending on the search objective.

Through the use of word frequency distributions, the system can tell the user the extent to which a particular term has been used as, for example, an index term in the indexing of the document collection. When this information is supplied, the user can judge the quantity of material that will be retrieved for a given request.

The second stage in the retrieval process is the search activity. It is in this stage that the comparison of the formal representation of the user's request is made to stored document representations. Consequently the system's effort in this stage of the process is greatest, and the user is resigned to waiting for the results to be displayed.

The final stage is concerned with post-search activities. The retrieval system has predicted which documents satisfy the request and now must display the output for the user. With the documents displayed in front of him, the user then evaluates the retrieved documents in terms of their relevance to his information need. The system uses this information to calculate a performance measure for the search. In addition, a feedback mechanism operates to revise the search procedure in light of the user's satisfaction. Table 3 summarizes the user and system activities.

Table 3

User and System Activities

	User Activity	System Activity
Pre-Search	Determine information need Enter Query Query negotiation	Syntax check of query Thesaurus lookup Query term frequency analysis Map query into formal language
Search	Wait	Select comparison method (retrieval rule) Search file
Post-Search	Read output Mark relevant material Use relevant material	Display output Calculate performance measure Revise strategy and/or query with feedback

4.3 Document Representation.

A number of developments suggest that useful information is being ignored when the only keys that can be used to retrieve documents from a file are author, title and index terms. Consider, for example, the MARC format for the interchange of bibliographic information on magnetic tape. [142]. It is suggested that a large number of the fields in this format are useful means of accessing bibliographic information. Kessler's work on bibliographic coupling indicates the usefulness of storing document citations in the file to allow citation indexing and searching. [68]. Another possibility is that of including non-content information about the document (i.e. context information) in the file. [87].[1]

Context information, as distinguished from content information is that material that describes characteristics of the author (his academic background, degrees, current research interests, employer, etc.), the journal in which the paper was published, the editor, the references, etc.

Thus one can see that there are a number of alternate document surrogates that can be stored in the file and used for retrieval. In this paper these alternate forms are called <u>representations</u> of a document. Examples of document representations include the title, author(s), abstract(s), full text, index terms, citation, cluster center descriptions, etc. of a document.

---

1. Preliminary evidence using a small corpus has not indicated the usefulness of this data for searching, but these results are based on too small a sample to be considered indicative or final. [99].

4.4 User-System Interaction.

The user has a number of alternative strategies that he can employ in his information seeking behavior. Instead of employing an information retrieval system, the user may browse through his personal library, consult a co-worker, phone a friend, or consult a reference librarian.

The user's time is an economic quantity. Given the cost of this time and the fact that there are a number of information seeking alternatives, Simon's concept of satisficing appears particularly applicable. [84], [119].[2] The user pursues a selected information seeking strategy until the cost incurred exceeds the level of satisfaction received. At that point another strategy could be adopted or the process could stop with the user satisficed.

It is suggested that a number of variables are tested by the user in deciding whether his cost of a particular strategy has exceeded the benefit. These variables include the time the user spends at the console of the information retrieval system, the time required to map the request into the retrieval system's query language, and the waiting time until the results of the search are displayed.

Another group of variables that determine user cost, more specifically relate to man-machine interaction. [64, p. 57]. The design of the console, the flexibility of the programs in allowing the user to go as slow or as fast as he wants in a dialog with the machine, all contribute to his willingness to use the machine and the value that he places in

---

2. Baker and Nance [5] have also suggested the applicability of this idea to information seeking behavior.

the retrieval system.

Finally, the user is influenced by the results that he obtains from the system. It is in this area that measures of retrieval effectiveness are valuable. (See Section 3.1). They provide the user with a measure of the degree to which he is satisfied with the system.[3]

The cost that the user assigns to the employment of the system is a combination of the factors described in this section. If the system does not satisfy the user requirements, it is not used. Thus the user cost-benefit function is a constraint that is considered by the system. This is accomplished in a number of ways. For a given query the retrieval system predicts for the user the cost of the search and the time required to perform it. The user can then broaden or narrow his request given his budget constraint.

4.5 Search Methodology.

A complex process is undertaken when a retrieval system attempts to find material relevant to a user need. The model of user interaction in Section 4.4 suggests that information seeking patterns vary according to user cost and benefit. This section elaborates the problem further by suggesting the need to pick an optimal combination of search comparison method and document representation for the search.

---

3. An evaluative measure with properties similar to the reward-cost concept discussed earlier in this section has been proposed by I.J. Good. [49]. He suggests that a linear relation between number of documents retrieved and the value of those documents to the user has not been established. The author hypothesizes that a more complex mapping function between value and number of documents is involved.

4.5.1 Alternative Comparison Methods.

Previously a number of search comparison methods were outlined. (Section 2.2.3). These included the simple matching technique, in which the query is compared with the document representations and the degree of similarity between the two is calculated. Extensions of the methodology include the elaboration of the terms in the query with related terms to effect associative searching. Alternatively it is possible that instead of looking at every document representation in the file, clustering could be employed, and only cluster center representations be compared to the query representation.[4]

Thus there are a number of different strategies that can be employed. Traditionally one or perhaps two of these have been implemented in a given operational system. In the proposed system, however, all the comparison methods will be available to the user. This is done on the theory that a specific strategy will have certain properties that make its use advantageous in specific circumstances. For example, comparison method X may be found to be extremely exhaustive in its search for documents meeting the query objective. Method Y, on the other hand, may be particularly useful when searching for one specific subject. Then the user will have the ability to decide which strategy to employ. Alternatively he may rely on the system to pick the strategy, or may be forced to pick one because of a requirement for a specific performance level or

---

4. The description of the retrieval system given in this section is structured to convey the concept of resource allocation in retrieval system design. (See Section 4.6). The alternative approach of a retrieval system design for a particular file structure is not considered. The allocation model is independent of the file structure used.

because of a given budget constraint.

4.5.2 Alternative Document Representations.

In addition to the need to pick a particular search comparison method or retrieval rule for a specific purpose, the user has the ability to select a document representation that will be used to compare the query against.

A document has associated with it a number of representations. (See Section 4.3). These surrogates include index terms, abstracts, subject headings, etc. When a search is made, the retrieval system picks a particular representation to compare the query against. For example, if it is desired to find an article written by author W, a search of the author representation is conducted.

The search for a particular author is the easiest case for the system to handle. This is so because the system knows which representation to compare against. But take the case of a request which is in the form of a set of words characterizing a chosen subject. Here the problem is more complex because there are a number of representations that could be used for the comparison, such as the document index terms or the document abstract.

The user then has the flexibility to decide which representation will be used in his search or alternatively to let the system decide. If a broad survey of literature is desired, there may be more benefit in using subject headings than author assigned index terms for the search comparison. On the other hand, if a very specific question is posed, the

query may only be able to be answered through searching the abstracts or full text of a document. Here again there is a trade off between the degree of generality or specificity required in the search and the cost and benefit of conducting it. By allowing this flexibility to exist, the system stands a better chance of satisfying a variety of users.

4.6 Retrieval Model.

Optimal performance of a retrieval system requires that both system and user resources be considered in determining an operating level. This section considers the issues involved in selecting such a level. It also sketches the manner in which the system's strategy can be modified in the light of changes in user assessment of system benefits.

4.6.1 User-System Resource Allocation.

The total cost of a retrieval system's operation for a query, C_T, is the sum of the system cost and the user cost. That is

$$C_T = t_u c_u + t_s c_s \tag{4.1}$$

where c_u is the cost per unit of user time and t_u is the amount of user time required for a given search. Similarly c_s is the cost per unit of system time and t_s is the amount of system time required for a given search.

It is presumed that the retrieval system performance can be

characterized by a measure called P. It is believed that P is a complex function that may include variables such as those used in calculating measures of retrieval effectiveness. (See Section 3.1). For this model the measure is considerably simplified so that it has the form

$$P = f(t_u, t_s). \tag{4.2}$$

That is, the performance is a function of the amount of user time and system time expended on the search. A number of more specific formulations are possible. For example, the relation could be

$$P = t_u t_s. \tag{4.3}$$

This is the form of an isoquant curve from economic theory. (See Figure 4). Each point on an isoquant curve represents the maximum output that can be produced with a given combination of inputs. Each of the curves of Figure 4 show combinations of user and system time that yield the same level of performance, P_i.

Very little information is available about the precise shape of the performance curves. It could very well be that the curves are L shaped or even straight lines. However, assume for the following discussion that the performance can be characterized by an equation such as (4.3). Then it is possible to solve equations (4.1) and (4.3) to find the optimal level of t_u and t_s that minimizes total cost for a given performance level.

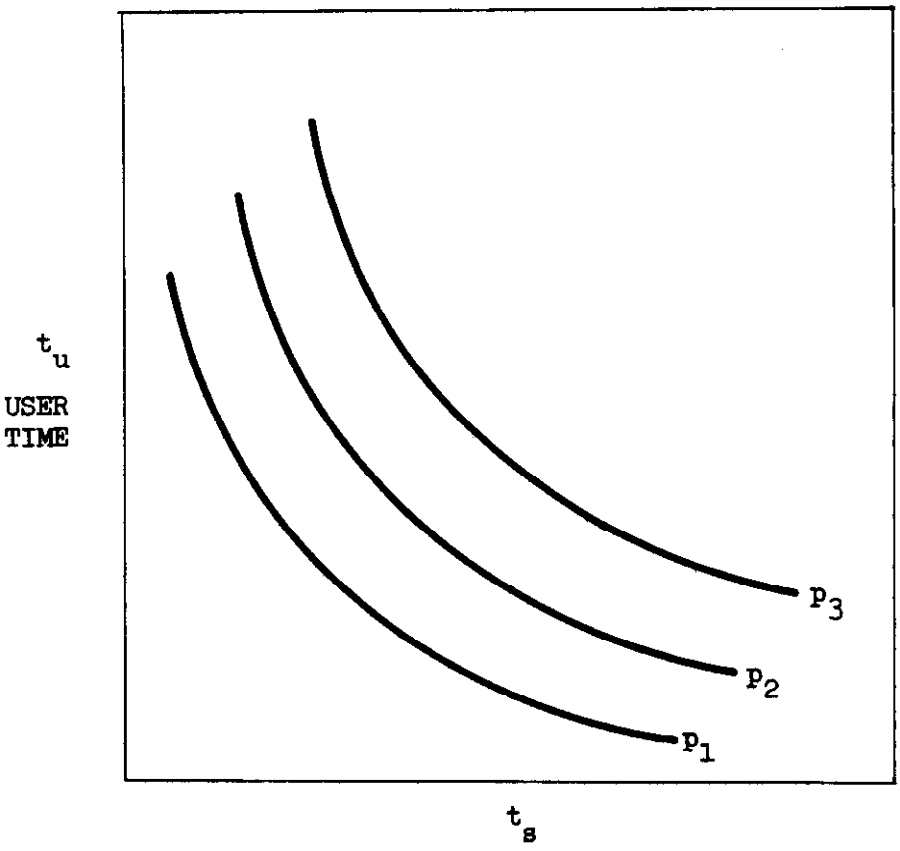
First rewrite equation (4.3) as

$$t_u t_s - P = 0. \tag{4.4}$$

Then using the Lagrange multiplier λ, form the equation

Figure 4

Isoquant Curves



$$F_\lambda = t_u c_u + t_s c_s + \lambda(t_u t_s - P). \tag{4.5}$$

The application of partial differentiation yields

$$\frac{\partial F_\lambda}{\partial t_u} = c_u + \lambda t_s = 0 \tag{4.6}$$

$$\frac{\partial F_\lambda}{\partial t_s} = c_s + \lambda t_u = 0 \tag{4.7}$$

$$\frac{\partial F_\lambda}{\partial \lambda} = t_u t_s - P = 0. \tag{4.8}$$

Then

$$t_s = -c_u/\lambda \tag{4.9}$$

and

$$t_u = -c_s/\lambda. \tag{4.10}$$

Substituting equations (4.9) and (4.10) into (4.8) yields

$$(-c_u/\lambda)\ (-c_s/\lambda) - P = 0 \tag{4.11}$$

$$\lambda = \sqrt{(c_u c_s)/P}\ . \tag{4.12}$$

Then the optimal t_u is

$$t_u^* = \sqrt{(c_s/c_u)\ P} \tag{4.13}$$

and

$$t_s^* = \sqrt{(c_u/c_s)\ P}\ . \tag{4.14}$$

Thus the optimal allocation of resources depends on the performance and the cost coefficients of each of the two resources.

4.6.2 System Resources.

System activity is divided into three areas: pre-search, search, and post-search activities. The system cost, $c_s t_s$, of equation (4.1) can be written

$$c_s t_s = c_{pre\text{-}s} t_{pre\text{-}s} + c_{search\text{-}s} t_{search\text{-}s} + c_{post\text{-}s} t_{post\text{-}s} . \quad (4.15)$$

The variables represent the costs per unit of time multiplied by the time used for each of the three activities.

The pre-search system cost per unit of time is given by

$$c_{pre\text{-}s} = \alpha_1 Ch + \beta_1 CPU + \gamma_1 Core . \quad (4.16)$$

Here Ch is the cost of the computer channels, CPU is the cost of the central processing unit and Core is the cost of the core storage per unit of time. The value of α, β, and γ represent the utilization rates of each of the components for the pre-search activity.

System search cost per unit of time is a function of the search comparison method employed and the document representation used. Thus

$$c_{search\text{-}s} = c_{comp.\ method} + c_{attribute} \quad (4.17)$$

Sections 4.6.2.1 and 4.6.2.2 explore these costs further.

Finally, the post-search cost is given in a form analogous to equation (4.16):

$$c_{post\text{-}s} = \alpha_3 Ch + \beta_3 CPU + \gamma_3 Core \ . \qquad (4.18)$$

4.6.2.1 Comparison Method Cost.

Each of the search comparison methods or retrieval rules employed in the retrieval system is presumed to have a cost associated with it. No general statements can be made about the exact formulae for the cost of a comparison method because the cost is highly dependent on the way in which a strategy is implemented in a computerized system. For example, the internal representation of the query and the documents will influence the cost. Nevertheless, it is possible to suggest the form that such an equation might take.

The comparison cost will be a function of the number of terms in the query, the number of logical operators in the query (e.g. 'and,' 'or,' 'not'), the number of document representations in the file, and the number of words in each representation. Additionally the cost will depend on the computer resources used: the central processing unit, core storage and the channels. Finally, the amortized cost of programming a particular comparison method will have to be included. For associative searching, the association files need to be constructed, and for comparison on cluster centers, the clustering will have to be performed.

4.6.2.2 Document Representation Cost.

The retrieval system calculates costs for document representations stored in the system. Total cost for a representation is made up of three components: creation cost, storage cost, and processing cost.

When documents are received at an information center a certain amount of pre-processing is performed before the document can be stored in the system's data bank. For example, the document may have to be indexed, assigned subject classifications, abstracted, etc. In addition, if it is not already in machine readable form, the conversion will have to be performed. All these functions are considered part of the information center's cost of creation of a document surrogate, c_{create}.

No uniform method exists for accurately determining document surrogate costs. Surveys by Landau [74] and Penner [100] have summarized the work that has been performed to date. Subsequently Leimkuhler and Cooper [76] proposed the use of standard cost accounting techniques as a solution to the problem. It is hoped that this approach will allow the application of generally accepted accounting principles to the problem of cost analysis of document surrogates.

The second surrogate cost component is storage cost, c_{store}. There are a number of variables that determine this cost: the rental cost of the computer storage device, the proportional cost of the control unit for the device, the capacity and utilization of the device, and the number of characters in the representation.

The final component of the surrogate cost function is the processing cost, c_p. Two types of processing are performed in the retrieval system: retrieving records from the file and creating and maintaining

the file. In addition, there are three components of the processing cost: the central processing unit cost, the channel cost, and the core storage cost. The basic form of the processing cost equation is the same as in equation (4.16):

$$c_{process} = \alpha_2 Ch + \beta_2 CPU + \gamma_2 Core \ . \tag{4.19}$$

While the costs of each of the computer components remains constant in the processing cost equation, the values of α,β, and γ vary depending on whether updating or retrieval is being performed.

To summarize, the cost of a document representation is

$$c_{representation} = c_{create} + c_{store} + c_{process} \ . \tag{4.20}$$

Preliminary analysis suggests that the cost differences between document representations in the same class (e.g. the index terms assigned to document number one and those assigned to document number two) may be so small as to minimize the need for cost computation for each representation of each document. Instead costs could be computed for each representation class. This follows the approach of standard costing suggested earlier. [76].

4.6.2.3 System Resource Allocation.

When the user begins a dialog with the system, he specifies a desired performance level and a budget constraint. Then using the solutions in equations (4.13) and (4.14) the system is able to divide the user's fixed budget between system activities and user activities.

This section explores possible approaches whereby the system can divide its time between pre-search, search, and post-search activities.

It was postulated earlier in this section that a relation exists between user time, system time, and system performance. It is also possible to establish a relation between performance, search comparison method, and document representation.

$$P = f(\text{comparison method, document representation}) \tag{4.21}$$

Using the equations reflecting representation and comparison method costs per unit time, it is possible to arrive at an optimal choice of document representation and comparison method that minimizes system search cost, $c_{search\text{-}s}$, for a given performance level. This is done in a manner similar to that used in Section 4.6.1.

Once the comparison method and document representation have been selected, the search cost is determined using equation (4.17). The search time, $t_{search\text{-}s}$, is calculated using the average number of documents to be searched or the average number of index entries to be searched. The user is charged based on the average cost figure, and variances are accumulated and at periodic intervals are used to readjust the cost coefficients. In this manner the total search cost,

$$c_{search\text{-}s} t_{search\text{-}s},$$

is determined.

The remaining problem facing the system is to allocate the remaining funds between the pre-search and post-search activities. No precise rules can be given for this. However, it is possible to delimit the values of

t_{pre-s} and t_{post-s} that are feasible. For instance the structure of the system may be such that pre-search activity requires a minimum of 'a' time units and cannot use more than 'b' time units no matter what performance is required. Then

$$a \leq t_{pre-s} \leq b \tag{4.22}$$

Similar bounds could be developed for post-search time requirements. As an additional future stage, it would be desirable to determine if a relation could be established between the system performance and pre- and post-search time allocations.

4.6.3 User Resources.

The second category of resources that are used in the retrieval process is user resources. The total cost of these factors is

$$c_u t_u$$

where c_u is the cost per unit of user time and t_u is the amount of user time expended for a given query. As before, a distinction is made between the three activities: pre-search, search and post-search. Then the total user cost is given by

$$c_u t_u = c_{pre-u} t_{pre-u} + c_{search-u} t_{search-u} + c_{post-u} t_{post-u} \; . \tag{4.23}$$

Here c_{pre-u}, $c_{search-u}$, and c_{post-u} are the per unit costs of the user's time for each activity, and t_{pre-u}, $t_{search-u}$, and t_{post-u} are the number of time units of each activity used for a given search.

It should be noted that equation (4.23) is again a simplification of the actual situation. Post-search activity time for the user is in actuality a function of the amount of time spent in pre-search activity.

In this model it is assumed that when the user is not availing himself of system services, the system can service other users or other jobs. Thus it is assumed that the system is never idle, or if it is the user does not pay for the idle system time. On the other hand, if the system is heavily loaded with other tasks, the user may have to wait for a response to his dialog with the system. This suggests that equation (4.23) should be modified as follows:

$$c_u t_u = c_{pre\text{-}u}(t_{pre\text{-}u} + \theta_{pre}) + c_{search\text{-}u}(t_{search\text{-}u} + \theta_{search}) + c_{post\text{-}u}(t_{post\text{-}u} + \theta_{post}). \quad (4.24)$$

The variable θ represents the additional time that the user must wait for the system for each of the activities. For example, search activity will require a small amount of user time perhaps to initiate the searching once the query has been accepted. Then the user will have to wait θ_{search} units of time until the system completes the search, where

$$\theta_{search} \geq 0. \quad (4.25)$$

The values of c_u and t_u in equation (4.24) are a function of the qualifications of the user, U. That is

$$c_u = f(U). \quad (4.26)$$

Many different people presumably use an information system. Each person most likely values his time at a certain price. The user cost per unit of time is related to this assessment of value by the user. For example,

a senior member of an organization will command a higher salary than a clerk. If both of them use the retrieval system, the allocation of resources between user and system will vary depending on the c_u and c_s values.

Similarly the time that a user spends at the console will depend on his experience with the system as well as his qualifications. Thus

$$t_u = f(U). \tag{4.27}$$

It is possible to conceive of a situation in which users with similar c_u values have different t_u values simply due to extensive practice with the system or a more agile mind. Equation (4.27) is intended to reflect this disparity.

In the cost model in this chapter, equations have been developed to show the total cost of operating a retrieval center. Total cost is a function of both the user and system cost of system operation. The model shows that the system cost is a function of the cost of the type of computing system employed, the type of retrieval rules that are used and the document surrogate that the query is compared against.

The model also shows that the cost of operating a literature searching system can be divided in another manner. This division has to do with the timing of activities that the system and the user engage in. It is shown that there are functions that both the user and the system can perform and that the optimal allocation of effort between the user and the system for a particular searching activity is a function of the cost of the user's time and the cost of the system's time.

Chapter 5

The Retrieval System Simulator

5. The Retrieval System Simulator.

One of the goals of this dissertation is to explore the feasibility of using simulation as a technique to evaluate information retrieval systems. In Chapter 2 it was shown that a great deal of research has centered on methods for content analysis, searching, file organization, etc. And in Chapter 3 a review of simulation studies in the area of information science showed that simulation techniques have been applied to a variety of problems related to information retrieval.

In this chapter simulation techniques are used to evaluate one aspect of the literature searching process - namely the characteristics of documents and queries and the manner in which these characteristics influence one aspect of system performance, namely the quantity of output produced by such systems.

How are retrieval systems evaluated? Historically the pattern has been as follows. First a collection of documents about a particular subject or subjects is gathered together. The bibliographic information about the documents and the document surrogates such as the abstract, index terms, etc. are then converted to machine readable form. The next step in the evaluation process is for the investigator to collect a number of queries that could be posed to the retrieval system and convert these queries to machine readable form. The queries are not confined to one subject area but rather cover a wide range of topics. Given the document collection and the file of queries, it is then possible to evaluate specific retrieval rules (Section 2.2.3) to see how the performance of the system varies. That is, it is possible to determine whether one retrieval rule (e.g. matching, searching using overlap rules, associative

searching, etc.) produces better performance than another. In addition many other components of the retrieval system can be evaluated, such as the thesaurus, and the query analysis technique. As was mentioned previously (Section 3.1), performance has traditionally been measured in terms of the recall and precision ratios. This method of evaluation requires that the user of the system make judgments about the relevance of the documents retrieved by the system to the request that was submitted.

A more comprehensive test of whether one retrieval rule or one component of a literature searching system is better than another would involve using a number of different document and query collections. Based on the performance resulting from document collection A and query collection X as against document collection B and query collection Y, etc., more meaningful inferences about the performance of a retrieval rule, etc. could be made.

The simulation model that is developed in this chapter is designed to investigate one portion of the evaluation problem. In particular a literature searching system is developed with an overlap retrieval rule. This rule measures the number of terms in common to a query and a document. The simulation program evaluates this retrieval rule by forming a set of pseudo-documents and a number of sets of pseudo-queries and comparing the queries to the documents. This procedure allows the measurement of the way in which the quantity of documents retrieved varies with changes in the rules and parameters used to create the various pseudo-query files.

The evaluation procedure used in the simulation study is analogous to the traditional procedure described above. The major difference is that instead of comparing various retrieval rules, the simulation model

is designed to evaluate the effect of changes in characteristics of the query and document files on the quantity of material retrieved. In the simulation model, the relevance of the document to the user is not simulated.

5.1 Document and Query Characteristics.

In order to understand the motivation for using simulation as an evaluative tool, it is necessary to understand the characteristics of the processes and objects being simulated. The model developed in this paper provides rules which are used to create a collection of pseudo-documents that can be used as the data base for a literature searching system. In addition, the simulation program creates a number of sets of queries that are used to interrogate the document file. These documents and queries have precisely defined characteristics and well specified rules for their generation. They are composed of sets of words picked from a created vocabulary. The individual words are the basic unit for conveying content and there is no grammatical structure to the documents or queries. The words are codes that have pre-established relationships among themselves.

The documents and queries are created using explicitly stated rules so that all characteristics of the documents and queries are well known. If a set of 'real' documents and queries were selected for use in evaluation of the system, not all the characteristics of the 'real' set could be precisely stated. Thus any conclusions about the performance of one component of a system relative to another (based on comparisons over a

number of document and query files) would have to be qualified by assuming that there was no change in performance caused by changes in the document and query files. By using pseudo-documents and pseudo-queries no such qualification is needed in stating whether one retrieval rule is better than another.

What are the variables that characterize a document collection? The list developed by Cuadra and Katter [27, Volume I] provides a good starting point.

> Subject matter (the field or fields of activity from which the document comes).
> Diversity of content within the document.
> Difficulty level of the subject matter in the document....
> Scientific "hardness" of the document. (Note: One often speaks of "hard" or "soft" sciences. The hardness of a particular document is indicated by the precision of the language and the relationship among the stated aims of the document, the conclusions, the methodology of inquiry, and the supporting data. If any of these, or the relationship between them, is ill-defined, nonexistent, unclear, questionable, or otherwise precarious, the document would be considred less "hard."
> Amount of "information" in the document....
> Level of condensation (or, conversely, of detail). (Note: This variable applies primarily to document representations.)
> Textual attributes (such as length, type-token ratio, etc.).
> Special qualitative attributes (such as interestingness, accuracy, credibility, workmanship, significance, etc.). [27, Volume I, p. 34-35].

Using the concepts in this list a procedure was developed which causes words to be selected to form a representation of a document. The generating procedure does not model all the variables listed above because of the magnitude of the problem. Instead certain characteristics are selected and effort is focused on modeling those chosen. A complete description of the model and the parameters will be found in the next section.

In viewing the above list of document characteristics, it seems apparent that mathematical solutions are possible in the investigation

of some of the variables. For example, if it were desired to determine the relation between document textual attributes, query textual attributes, and numbers of documents retrieved, a mathematical solution seems possible. For example, Bourne has shown that it is possible to develop a relation between number of documents retrieved and depth of indexing. [17, p. 58-69]. On the other hand, this characteristic appears to be the only one that is amenable to mathematical investigation because it is the only variable that has so far been quantified.

Once a well defined document collection is available as a data base, the next step is to create pseudo-queries with which to search the data base.

Characteristics of queries include[5]

> Subject matter (the field of interest or content to which the requirement statement refers).
> Diversity of content suggested by the statement. (Note: If two different but partially related information requirement statements were combined into a single statement in such a fashion as to preserve all features of both, the composite statement would be considered as more diverse than either of its components.)
> Difficulty level (Note: This variable has to do with the relative ease with which, in a given setting or facility, an information requirement statement may be understood and processed.
> Specificity or Amount of "Information." (Note: Subject matter may be explicitly stated, reliably implied, or only loosely implied; so may other document characteristics, such as emphasis on factual information, specifications, theoretical discussions, general descriptions, etc.)
> Functional ambiguity (the occurrence of words or phrases that are capable of different interpretations in different use contexts, and that are not clarified within the context of the statement).
> Textual attributes, such as length, number of nonsynonymic or nonredundant content words and phrases in statement, number of syntactic connections between such content words and phrases, etc. [27, Volume I, p. 35].

---

5. Cuadra and Katter use the term 'information requirement statement' to describe the request that a user makes of an information retrieval system. While not precisely equivalent, the simulator uses the word 'query' to represent the same concept.

Here again some of the above characteristics were used to develop rules used by the simulator to create pseudo-queries. The process is described in detail later.

Given a statistically controllable set of documents and queries, it is then possible to test a variety of search methods, association measures, feedback principles, etc. to evaluate which methods produce the best performance of the system. The effect of creating pseudo-documents and pseudo-queries is to eliminate the differences in variability between various document files and query files so that the true differences between searching techniques, etc. can be observed.

5.2 An Overview of the Simulator.

The retrieval system simulator is composed of five parts:

1. The thesaurus generator.
2. The document generator.
3. The query generator.
4. Search routines.
5. Evaluation routines.

The first step in the simulation involves the creation of a thesaurus. This procedure establishes the relationships between all the words in the simulated vocabulary. The methodology involves the use of mathematical distributions that characterize the frequency of occurrence of words in a set of documents. Then a number of words are 'created' with the frequency specified by the mathematical distribution, and using an assignment rule, these words are distributed to simulated word

classes. This procedure is described in detail in Section 5.3.1. The final result of the generation process is an association matrix which reflects the strength of the relations between the words in the vocabulary. The thesaurus generation routine allows a number of parameters to be varied including

1. The size of the vocabulary.
2. The form of the word frequency distribution.
3. The number of word classes formed.
4. The rule for assigning words to classes.
5. The measure used to calculate the similarity between two words.

The document generating routines create a specified number of documents. Each document is composed of a maximum number of compressed representations such as abstracts, index terms, etc., and these are generated at the same time. For a given representation the simulator calculates the number of words in the representation and then randomly selects a set of starting terms to be included in the representation. Using the thesaurus, an additional number of words related to the starting words are selected for inclusion in the representation. Succeeding representations of the same document are derived using the base representation and transition probabilities. Only content representations are generated in this manner. The simulation routines do not generate context representations. (See Section 4.3). The parameters used in the document generator are

1. The number of documents to be generated.
2. The maximum number of representations per document to be generated.

3. The probability of a given representation occurring as part of the total document description.

4. The mean and standard deviation of the length of each alternate representation.

5. The proportion of terms that are to be included in the representation that are designated as starting terms in the base representation.

6. The threshold probability that a word that is associated with a starting term will be included in the document representation.

7. The probability for a given representation that the words in the base representation will appear in a succeeding representation of the same document. For example, this rule determines the probability that a word appearing in the abstract of a document will appear in the title.

Query generation is accomplished in a manner somewhat similar to document generation. Queries are grouped into query subsets. Each subset represents requests about a particular subject. The first query that is generated for a subset serves as a base for generating the second query. As more queries in a subset are generated, a rule selects which of the preceeding queries will be the base for generating the next one. The parameters involved in this process are

1. The number of queries to generate. (This overrides number 2 below).

2. The mean and standard deviation of the number of queries per subset to generate.

3. The mean and standard deviation of the length of a query.

4. The proportion of terms that are starting terms in the first query of the subset.

5. The probability that a term that is associated with a starting term will be included in the query.

6. A rule for selecting the base query.

7. The transition probability for a given query subset that terms in the base query will appear in subsequent queries in the subset.

The search and evaluation routines of the simulator are not simulation routines. The search programs take the pseudo-document file and pseudo-query file and compare the queries to the documents to see the extent to which the queries match the documents. The evaluation routines apply various threshold tests to determine how many documents would have been retrieved for a given threshold and for a given surrogate of a document.

5.3 Thesaurus Construction.

The simulation program creates pseudo-documents and pseudo-queries using statistical properties of word usage. In the process of generating the documents and queries, the simulator relies on a thesaurus to indicate the relationships between the words that will be formed into a document representation and into a query.

There are a number of ways in which relationships between words can be expressed. For example, syntactic relations allow specification of whether terms are synonyms of one another, antonyms, whether one word implies another, etc. In addition there are logical relationships

between words and also statistical relationships. Statistical relationships are implied by the co-occurrence of words in text. In the simulation model, the relationships are expressed in a statistical manner.

The relationships between words in the system's vocabulary are stored in a symmetric matrix. Consider the example of a thesaurus representing a fifteen word vocabulary in Figure 5. The thesaurus indicates that term number one is related to term number three with a strength of 0.25, and that term number one has no relation to term number two. The relation between the pairs (i,i) is undefined for purposes of the simulation. An element in the thesaurus (association matrix A), a_{ij} can take on values in the range 0.0 to 1.0, where 0.0 indicates no relation between terms i and j, and 1.0 indicates synonomy or perfect co-occurrence between the two.

In the simulation model, not every word that is used in English is represented in the thesaurus. Instead the thesaurus is intended to model word relations in a small subset of English. For example this subset could be all the words used in the technical literature in the field of information retrieval.

There is another restriction on the words that are entered in the thesaurus. Given the set of terms in the hypothetical field of knowledge, the thesaurus will only contain entries for content bearing, normalized, word types in that vocabulary. These qualifications are discussed below.

A text is composed of a number of words. Were one to count each individual word in the text, the result would be the number of word tokens in the text. The next step in the counting procedure would be to looks at the list of word tokens and determine how many unique (that

Figure 5

Hypothetical Association Matrix

TERM NUMBER	TERM NUMBER 2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2		0.00	0.25	0.00	0.00	0.20	0.00	0.16	0.00	0.33	0.00	0.50	0.00	1.00
3			0.16	0.25	0.20	0.27	0.50	0.28	0.00	0.00	0.00	0.00	0.20	0.00
4				0.00	0.25	0.30	0.25	0.14	0.00	0.25	0.00	0.00	0.00	0.25
5					0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6						0.20	0.33	0.16	0.00	0.33	0.00	0.00	0.33	0.00
7							0.20	0.36	1.00	0.20	0.10	0.10	0.20	0.20
8								0.16	0.00	0.00	0.00	0.00	0.33	0.00
9									0.20	0.00	0.00	0.00	0.40	0.16
10										0.00	0.00	0.00	0.00	0.00
11											0.00	0.50	0.00	0.33
12												0.00	0.00	0.00
13													0.00	0.50
14														0.00

is, different) tokens there are in the list. For example, there may be seven occurrences of the word 'system.' These seven tokens represent one word type, and the thesaurus only records word types. The two other qualifications further limit the thesaurus subset. By the previous definitions, the words 'system' and 'systems' are two word types. Normalization involves reducing a term to its stem, and the result of this procedure would be one word type - 'system.'

The final prerequisite necessary for inclusion of a word is that it be 'content bearing.' No precise definition of this concept is given in this dissertation. Operationally a content bearing word could be distinguished from a non-content bearing word by the use of word frequency statistics. The most frequently occurring words in a vocabulary would include 'the,' 'a,' 'and,' etc. and would be defined to be non-content bearing. Examples of such lists may be found in [69].

In order to establish the relationships between words in the vocabulary, a series of steps are performed. First, a mathematical distribution is used to characterize the frequency of occurrence of word types in the body of knowledge for which the thesaurus is being constructed. The information from this distribution is then used to provide a word description of a number of sub-classes that are found within the general subject field being modeled. Given the distribution of words in these classes, an association matrix is then computed.

5.3.1 Creation of Term Classes.

The objective of the thesaurus generating component of the simulator is to create a matrix reflecting the strength of the relation between pairs of terms in the vocabulary. In order to calculate these relations, it is necessary to know the extent to which a particular word, w_i, co-occurs with other words of the vocabulary. Thus the model being used to develop the thesaurus states that if two terms occur together in a document, a statistically significantly large number of times, then the terms are related to each other. The strength of the relation between the terms is a function of the number of times that they co-occur.

The simulator uses a rather simple strategy to form word co-occurence patterns. A number of 'term classes' corresponding to sub-fields within the field of knowledge of the vocabulary are created. For example, assume the simulation were modeling the vocabulary of the information retrieval literature. Then examples of sub-fields might be clustering, relevance, content analysis, etc.

A term class is described by the word types assigned to it. Using an analogy from the clustering literature, a term class could be considered as a description of a cluster containing a number of documents more related to each other than to documents outside the cluster. If documents were clustered on the basis of the index terms assigned to them, it would then be possible to form a term class or cluster using the collection of terms that have been assigned to the documents that form the cluster. In the case of the simulator, this is exactly what is presumed: a term class is characterized by the word types that describe the class.

The term classes are created in order to simulate the way in which

terms in the vocabulary co-occur. In the following section a word distribution is presented to characterize the frequency of occurrence of terms across all term classes in the vocabulary. The thesaurus generation routine begins with the first word in the vocabulary and, depending on the frequency of the word's occurrence, the routine specifies that the word will be present in a number of term classes. (See Section 5.3.1.3). Once the pattern of term occurrences over all term classes is known, then this co-occurrence information is used to compute the association between the terms in the vocabulary. (Section 5.3.2).

5.3.1.1 The Word Frequency Distribution.

The procedure employed in developing the thesaurus uses a mathematical distribution to characterize the frequency of word occurrences in term classes. The form of word distribution that is required for this application is one which gives the frequency of occurrence of word types across term classes. There are no known distributions that give the required relationship, but it is believed that there are distributions that can approximate the one required.

For example, consider the Waring series expansion [59] for

$$1/(x-a). \tag{5.1}$$

Gustav Herdan has shown that this distribution has the characteristic of a long tail - something frequently found in word distributions. [53].

$$1/(x-a) = \frac{1}{x} + \frac{a}{x(x+1)} + \frac{a(a+1)}{x(x+1)\ (x+2)} + \ldots \tag{5.2}$$

By multiplying both sides of equation (5.2) by (x-a) the frequency distribution is obtained.

$$1 = (x-a)\left\{\frac{1}{x} + \frac{a}{x(x+1)} + \frac{a(a+1)}{x(x+1)(x+2)} + \ldots\right\} \tag{5.3}$$

Each term, p_n, on the right hand side of the equation is then interpreted by Herdan as the fraction of the vocabulary that will occur n times.

The Herdan-Waring distribution results in the following computing forumula:

$$p_n = \frac{(x-a)a(a+1)(a+2) \quad \ldots \quad (a+n-2)}{x(x+1)(x+2) \quad \ldots \quad (x+n-1)} \tag{5.4}$$

where

$$a = \frac{1}{\frac{1}{(1-p_1)} - \frac{1}{\bar{x}} - 1} \tag{5.5}$$

and

$$x = \frac{a}{1-p_1} \; . \tag{5.6}$$

Thus the distribution requires two parameters. The arithmetic mean of the values, $\bar{x}$, is given by

$$x = \frac{1}{N} \sum_{i=1}^{n} f_i x_i \tag{5.7}$$

where N is the number of word tokens in the text being analyzed, f_i is the frequency of the i th word type, and x_i is the rank of the i th word type.

The second parameter of the distribution is p_1. This is the proportion of the vocabulary occurring only once in the corpus.

The distribution that is required for assigning terms to classes must give the frequency with which a given word type occurs over all

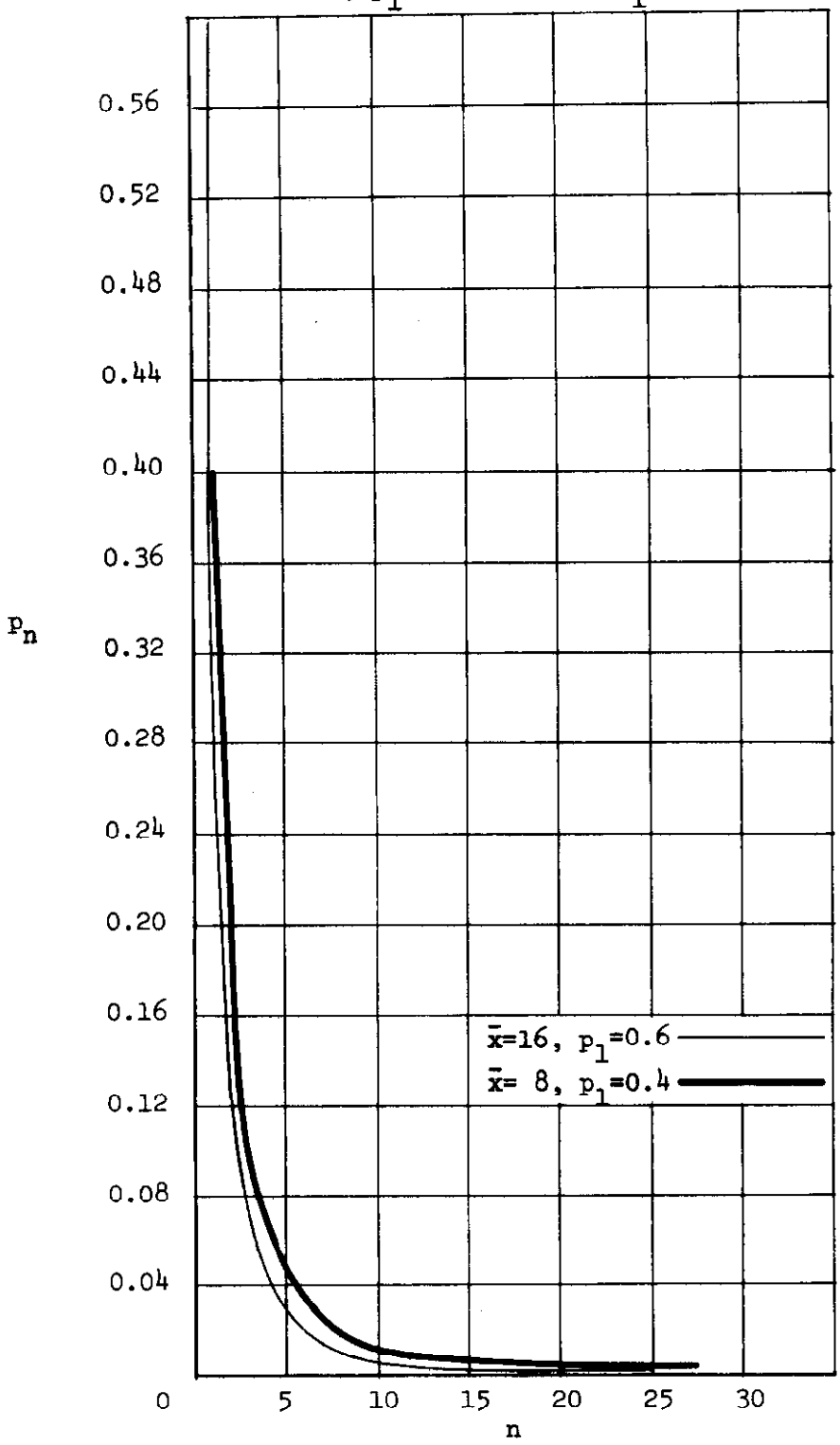
term classes. If one considers a large number of classes then it is most likely that when the frequency of each word type is tabulated, there will be a large number of types that occur only once in the classes and a small number of word types that occur more frequently. The shape of the curve is generally that of a decay function. Herdan has shown the applicability of the Waring distribution to word token occurrences. Jones, Giuliano, and Curtice have found the distribution applicable in characterizing occurrences of terms in a large technical document collection. [62, p. 63 and 71]. It was decided that the Waring expansion would be used to characterize the word type distribution required in the simulation. Any decay function could have been used, but since there was no evidence to suggest one distribution should be used rather than the other, the Waring expansion was selected.

5.3.1.2 Absolute Frequencies of Term Occurrence.

As a result of applying the two parameters $\bar{x}$ and p_1 to the Waring expansion, an infinite number of pairs of the form (n, p_n) result. In adapting the distribution to the purpose of the simulator, it is assumed that a relation of the following form is produced: 'there are p_n word types in the vocabulary that occur n times.' While theoretically n can take on values from one to infinity, in practice the values of p_n as calculated from the mathematical distribution are almost always zero by the time n reaches 50. Figure 6 plots the general form of the Waring expansion for various values of $\bar{x}$ and p_1.

In order for these (n, p_n) tuples to be of use to the simulator,

Figure 6
Plot of Waring Series Expansion for
$\bar{x}=16$, $p_1=0.6$ and $\bar{x}=8$, $p_1=0.4$



they must be converted to a form which will allow statements to be made about the absolute frequency of occurrence of an individual word across all term classes. The conversion procedure is as follows. Every word type in the vocabulary is given a number from one to N, where N is the size of the vocabulary for the particular simulation run. Then

$$a_n = p_n N \tag{5.8}$$

where a is the absolute number of terms that occur n times in the term classes. Now define w as the absolute frequency of occurrence of term i in all term classes, where i = 1(1)N. Then

$$w_1 = 1$$

$$w_2 = 1$$

$$\vdots$$

$$w_{a_1} = 1$$

$$w_{(a_1+1)} = 2$$

$$\vdots$$

$$w_{(a_1+a_2)} = 2$$

and in general the values of

$$w_1 \text{ to } w_{a_1} = 1,$$

$$w_{(a_1+1)} \text{ to } w_{(a_1+a_2)} = 2,$$

and

$$w_{a_{n-1}} \text{ to } w_{(a_{n-1}+a_n)} = n \qquad 1 \leq n \leq N. \qquad (5.9)$$

The simulator operates on a small finite corpus, and for this reason it cannot accept all p_n values generated by the Waring-Herdan distribution. This requires that

$$p_n \geq 1/N \qquad (5.10)$$

to insure that when absolute frequencies are calculated, the sum of the frequencies are at least equal to one. This restriction is not unreasonable conceptually. It says that if a corpus is small, then there is less likelihood that large values of n will be present.

The set w_i is ordered by frequency of occurrence after it is generated. The w_i's represent a list of words in the vocabulary. For purposes of preserving the independence properties of the simulation procedure, the order of the words is randomized. The randomized set of w_i's is then considered a list of the absolute frequency of occurrences of word types in the vocabulary.

5.3.1.3 Assignment of Terms to Classes.

The number of term classes, m, created must be greater than or equal to n, the largest frequency that is associated with w_i. The simulator allows m to be varied in order to evaluate the effect of the variation on the values in the association matrix.

Once the number of classes, m, has been established, an N by m

matrix, called a class matrix, is formed (N is the size of the vocabulary). An example of a class matrix is given in Figure 7. The figure shows that term number one occurs in term classes number one and four, while term number four occurs only in term class number five.

The class matrix, C, is formed a row at a time. Beginning with the first word in the vocabulary, the frequency of the w_i value is examined. Then an operation is performed such that if w_i had the value of two, the procedure would mark the presence of term number one in two of the five document classes of Figure 7. The simulator allows the rule whereby the terms are assigned to classes to be varied. A number of rules are possible, but currently the routine uses a uniform random number generator to assign the terms to the classes. The process is repeated until all words have been examined and occurrences assigned to classes.

It is also possible to add procedures to the simulator to replace binary values with probabilities in the class matrix.

Figure 7

Hypothetical Class Matrix

TERM NUMBER	TERM CLASS 1	2	3	4	5
1	1	0	0	1	0
2	1	1	0	1	1
3	0	0	1	1	0
4	0	0	0	0	1
5					
.			.		
.			.		
.			.		
N					

5.3.2 Generation of Association Matrix.

The final step in the thesaurus generation process is the creation of the symmetric term-term association matrix. This matrix, A, is the mathematical representation of the relationship between pairs of terms. The method first computes the frequency with which pairs of terms co-occur in the term classes and then calculates the association between the terms using the co-occurrence frequencies and the absolute frequencies of the terms as they occur in the term classes as given by w_i.

Define each row in matrix C (Figure 7) as a 'class vector' C_i for each of the N terms. The frequency with which terms i and j occur in the same class is

$$f_{ij} = C_i \cap C_j \qquad \text{for } i = 1(1)N-1 \text{ and } j = i+1(1)N. \qquad (5.11)$$

There are a number of formulae available to compute the association or similarity between the pairs of terms (i,j). See, for example, [70] and [122]. Appendix 1 contains a discussion of various criterion for classifying these measures along with a listing of some of those more commonly used. Initially the association measure used in the simulation is that employed by Rogers and Tanimoto [108], and Doyle [35]. (See also [132]). It is given by

$$a_{ij} = \frac{f_{ij}}{w_i + w_j - f_{ij}} \qquad \text{for } i = 1(1)N-1 \text{ and } j = i+1(1)N. \qquad (5.12)$$

5.4 Document Generation.

A document, D_j, is composed of a number of representations such as full text, abstract, index terms, etc. Each surrogate of the document, x_i, is made up of a number of terms, a_t. The complete document is composed of all the representations and the terms associated with each representation. For example, consider hypothetical document number 34 composed of three representations. Representation number one has three terms in it, representation four has two terms, and representation seven has four terms. Then the document could be described in set notation as

$$D_{34} = \{ x_1 \{ a_4, a_9, a_{71} \} , x_4 \{ a_{13}, a_{62} \} , x \{ a_9, a_{12}, a_{19}, a_{43} \} \}$$

and in general

$$D_j = \{ x_i \{ a_t \} \} . \tag{5.13}$$

Here the a_t's are terms assigned to the surrogate. Since the simulator deals with word types, a given a_t can not appear more than once in a given x_i. However, the same a_t can appear in more than one representation. Thus term a_9 appears in surrogates x_1 and x_7.

In the simulation model, representations of a document have a certain probability of occurrence. That is, every time a document is generated, the simulator generates a random variable, compares the value to an input parameter threshold for the representation, and determines whether the representation will be generated as part of the particular document.

5.4.1 Generation of Document Representations.

Once it has been determined that the representation will be generated, the simulation program determines the number of words in the representation. This is done via a normal random number generator which uses the input parameters of mean and standard deviation of the length of the representation to calculate actual length.

The procedure used to generate a set of representations for a document begins with the creation of the surrogate with the greatest number of terms. Then transition probabilities are used to form the remaining representations from the so-called base representation. The base representation can be thought of as the full text of the document. The model then uses the full text as a basis from which to derive index terms, title, etc. In actuality because of time and cost constraints, the model assumes that the base representation is the abstract of the document. It then uses the abstract to generate other representations.

5.4.2 Generation of Base Representation.

Two steps are involved in creating a base representation: selection of starting terms and selection of derivative terms. A starting term is one picked at random from the vocabulary. A derivative terms is one that is related to the starting term. The relationship is determined by consulting the thesaurus.

Assume that the number of words in the base representation has been generated and its value is n_b. As a prerequisite to generating the

terms, the number of starting terms is calculated.

$$n_s = n_b p_s \qquad 0 \leq p_s \leq 1 \tag{5.14}$$

where p_s is the fraction of terms in the base surrogate that are to be starting terms.

The number of derivative terms, n_d, is then

$$n_d = n_b - n_s \ . \tag{5.15}$$

The set of starting terms for the base representation, i.e. the a_i's, is selected from all terms in the vocabulary, i.e. the w_i's.

$$\{ a_t \} \subset \{ w_i \} \qquad t = 1(1)n_s, \ i = 1(1)N. \tag{5.16}$$

The simulator uses random numbers in the range

$$1 \leq i \leq N$$

to select n_s words to form the starting set. The only rule limiting inclusion of members in $\{ a_t \}$ is that a given word can only appear once in the set for the given surrogate.

Next, the derivative words are selected. Beginning with term a_1 the program searches the thesaurus to find the word most closely related to a_1. Then a random variable is generated. If the value of the random variable is less than the threshold probability p_{t_1}, the word is included in the base surrogate. If the word is not included in the base representation, then a randomly selected word is chosen. In this manner terms a_1 to a_{n_s} are used to select terms $a_{(n_s+1)}$ to a_{2n_s}. The process continues until a total of n_b terms have been generated.

After each n_s derivative words have been generated, two changes

occur. First, the threshold probability, p_{t_n} is changed. In addition, the previously generated derivative words are used as a base for selecting the second generation derivative words.

For example, suppose it is desired to generate an abstract having a length $n_b = 8$ words. Given that the fraction of starting terms $p_s = 0.4$, then $n_s = 3$ and $n_d = 5$. Assume that terms number 3, 11, and 7 are picked at random from a hypothetical 15 word vocabulary. Then the set of starting terms is

$$\{ a_3, a_{11}, a_7 \} .$$

The threshold probabilities for this example are

$$p_{t_1} = 0.8$$

$$p_{t_2} = 0.7 ,$$

and further, assume that the following list of 'random' numbers will be consulted for the example:

number	value	number	value
1	54	9	57
2	24	10	34
3	02	11	40
4	31	12	68
5	36	13	70
6	76	14	67
7	42	15	08
8	74	16	76

To begin, the first term, a_3, is examined. A random number is generated, whose value is 0.54, and is compared to p_{t_1}. Since the random variable is less than p_{t_1}, the word most highly associated with a_3 is found from the hypothetical association matrix in Figure 5. The term most highly associated with it is term number eight, and this is added

to the starting set. The set now contains

$$\{ a_3, a_{11}, a_7, a_9 \} .$$

Next, a_{11} is selected. Since 0.24 is less than p_{t_1}, a_{13}, the word most highly associated with a_{11}, is included in the set. The process continues by examining a_7 and selecting a_{10}.

At this point the set of starting terms has been exhausted, and a first generation of derivative words has been created. However, only six terms have been generated and eight are required. The next step, then, is to use a_9 and a new threshold, $p_{t_2} = 0.7$, to select the seventh term for the abstract, term a_{14}. Finally, a_{10} is used to pick the eighth and final term. When the random variable 0.76 is compared to the threshold p_{t_2}, the threshold is exceeded. The procedure then picks a word at random to be included in the set. In this case the random variable 42 is made modulo 15 and word a_{12} is picked. The final set of terms for the abstract is then

$$\{ a_3, a_{11}, a_7, a_9, a_{13}, a_{10}, a_{14}, a_{12} \}$$

The first three terms (a_3, a_{11}, and a_7) are starting terms; the next three are first generation derivative words; and the last two terms are second generation derivative words.

5.4.3 Generation of Derivative Representation.

The simulation program uses transition probabilities to generate the derivative document representations from the base representation. The number of words in the derivative representation is calculated in a manner similar to the way in which the length of the base representation was computed. The mean and standard deviation of the length of each surrogate is supplied as input to the routine, and the actual length is calculated using a normal random number generator. The only rule regulating the generated length is that it can not exceed the length of the base representation, x_1.

For a given surrogate, x_i, there is a probability p_{m_i} that the words in x_1 (the base representation) will appear in x_i. A random variable is generated; and if its value is less than p_{m_i}, then the first word in x_1 is transferred to x_i. The process is repeated for all words in x_i. If the process results in all words from the base representation being transferred to the derivative representation, then the process is completed. However, if not all words were transferred and the derivative representation is short of its required number of words, then words are selected randomly to fill the vacant positions. Another rule that the simulator has available is the ability to select words highly associated with the existing set to be included in the blank spots in the derivative representation.

5.4.4 Document Generation Parameters.

A number of parameters and rules have been introduced to characterize in a simple manner the construction of document representations. To review, it seems important to point out the way in which these parameters will be used to generate a document collection.

By varying p_s, the fraction of terms in the base representation that will be starting terms, control is exercised over the subject span of the document surrogate. For a large value of p_s there will be more starting terms which are picked at random from the vocabulary. A smaller p_s will create more derivative terms and thus more terms that are related to one another.

As each new generation of derivative words is created, the threshold probability p_{t_i} is changed according to the generation of derivative words being created. When p_{t_i} is allowed to vary in this manner, changes in the strength of the linkages between associated words are made. This variation prevents the development of a generating pattern in which, after a_i is selected, a_j will, with a very high probability, be selected. In addition, by varying p_t recognition is made of the fact that the further along a word association chain one proceeds, the weaker will be the chance of the chain continuing without being broken.

When the transition probabilities, p_m, used to select words for inclusion in derivative representations, are varied, control is exercised over the similarity between surrogates of the same document. The lower value of p_m, the less similarity there will be between surrogates. By varying this parameter, the extent to which words not in the base representation are introduced into derivative representations is regulated.

5.5 Query Generation.

The process of query generation parallels that of document generation with a few differences. A query, Q_k, is composed of a set of terms, y_r, contained in the vocabulary set w_i,

$$Q_k = \{ y_r \} \subset \{ w_i \} \qquad r = 1(1)d \ . \tag{5.17}$$

Here d is the number of terms in the query. At this stage in the development of the simulator, the terms that form the query are considered to form a logical disjunction. Further refinement will lead to a more elaborate generation method.

The retrieval system simulator forms groups of queries into query subsets. Input parameters to this part of the routine include the number of queries to be generated and the mean and standard deviation of the number of queries per subset to be generated. A query subset is intended to represent the dialog of an individual user with the retrieval system with regard to a specific subject. Thus each subset contains queries that are related to each other. A query file is composed of a number of query subsets.

5.5.1 Base Query Generation.

As was the case for documents, the simulator begins by generating the base query in each query subset. This procedure is analogous to that used to create the base representation for a document. The total number of words in the base query is determined from a normal random number

generator and then the fraction of terms that will be starting terms is calculated. These terms are then used in conjunction with the thesaurus and the threshold probabilities to generate the derivative words.

5.5.2 Derivative Query Generation.

The remaining queries in a particular query subset are generated using transition probabilities as described in Section 5.4.3. There is one exception to the procedure. As a user continues in a dialog with the retrieval system, his perspective is liable to change with regard to what terms to use to interrogate the file. To reflect this fact, the simulator allows the base query to change in the course of generating the queries in the subset.

For a particular subset, query number one in that subset is assumed to be the base query. It is generated as described in Section 5.4.2. The second query is generated using the methodology of Section 5.4.3. When the third query is about to be generated, the simulator consults a probability distribution to determine which of the previously generated queries will be used as a base query. A number of types of distributions are possible. A rule can be supplied to pick one of the queries at random to be the base. Alternatively it is possible to specify that there will be a greater probability that a query generated later in the sequence will be the base query rather than a query generated early in the subset.

5.6 Search and Evaluation Procedures.

Once the document and query files have been generated they are used to evaluate retrieval rules such as those described in Section 2.2.3, or other parameters of the system. The current version of the simulator has only the most simple retrieval rule implemented. With this overlap rule the number of terms common to the document and the query is recorded.

The evaluation procedure consists of summarizing the results of a comparison between a query file and a document file. There is no evaluation on the basis of the relevance of a document to a user. Because the relevance factor is omitted, the simulator simply accumulates information on the number of searches that resulted in a match between a document representation and a query for all queries of a given file and all document representations of a document file. This evaluation procedure is described in detail in Chapter 6.

Chapter 6

Evaluation of the Simulation Model

6. Evaluation of the Simulation Model.

A number of experiments were conducted using the retrieval system simulator. The purpose of these experiments was to evaluate the simulation model as a technique for studying information retrieval systems.

The simulation model allows a number of variables to be analyzed. However, due to constraints of time and cost, the only component of the system that was analyzed was the effect of changes in query characteristics on the quantity of material retrieved.

This chapter begins with a discussion of the experimental design used in the simulation runs. In succeeding sections, an analysis of the generated thesaurus, document, and query files is presented, and the experimental results are analyzed.

6.1 Experimental Methodology.

A complete experiment using the retrieval system simulator involves five steps. First, a thesaurus is generated. Then the thesaurus is used in the creation of a file of pseudo-documents. Next, the thesaurus is again used in the creation of a file of pseudo-queries. Finally the query representations are compared to the document representations and the results of the comparisons are tabulated.

An ideal experimental design which could be used to test the effect of changes in the model parameters would involve a factorial experiment. This would mean creating a number of thesaurus files, a number of document files and a number of query files. Each file would be generated

for a given value of a given parameter and succeeding files would have the parameters of each of the files systematically varied to determine the effect of various value changes on retrieval results.

As mentioned earlier, budget constraints prohibited performing a factorial experiment on all components of the system. Instead, one thesaurus file was generated and one document file was also generated. Systematic variations were made in all the parameters of the query generation programs, and twenty-two query files were generated. Each query file was compared to the document file and the results were tabulated.

A more detailed discussion of the experimental design is postponed until Section 6.4.1.

6.2 The Thesaurus.

One thesaurus was generated for the simulation experiments. It is identified in later discussions as T01. The values of the parameters used to generate this thesaurus are listed in Table 4. A plot of the Waring series expansion for the Herdan parameters of $\bar{x} = 15.0$ and $p_1 = 0.40$ is given in Figure 8. The figure represents the form of the initial word distribution used in the run.

The generated thesaurus is represented in the form of a symmetric 200 x 200 matrix. Table 5 displays a frequency distribution of the number of elements in the matrix falling in a specified value range. For all elements in the matrix, the mean value is 0.086 and the standard deviation is 0.161. Thus while the range of possible values that

Table 4

Parameters of Thesaurus T01

Parameter	Value	For explanation see Section
Word frequency distribution used	Waring series expansion	5.3.1.1
Parameters of word frequency distribution	$\bar{x} = 15.0$ $p_1 = 0.40$ $n = 50$	5.3.1.1
Vocabulary size	$N = 200$	5.3.1.2
Rule for assigning words to classes	Random assignment (uniform probability distribution)	5.3.1.3
Association measure used	Rogers and Tanimoto/ Doyle	5.3.2 and Appendix 1

any element in the matrix can take on is between 0.0 and 1.0 inclusive, the mean is very low indicating that a majority of the terms are not statistically related to each other using the Rogers and Tanimoto/Doyle association measure. Nearly 29% of all entries in the matrix are zero. At the other extreme, only 2.2% of the entries in the matrix have a value greater than 0.90.

Figure 8

Plot of Waring Series Expansion for $\bar{x}=15.0$ and $p_1=0.40$.

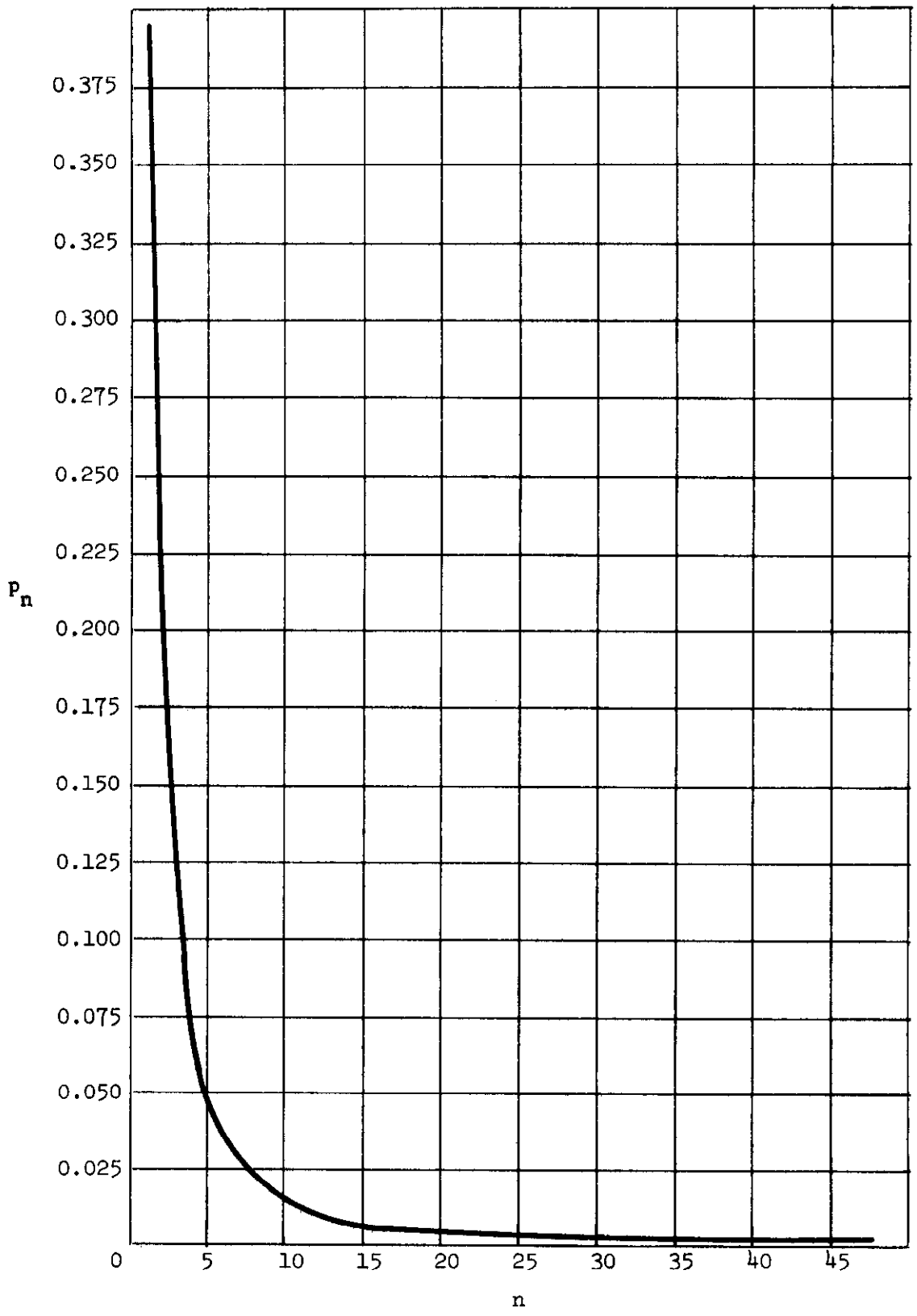


Table 5

Frequency Distribution of Values

in Thesaurus T01

Interval	Number of Values in Interval	Percent of Values in Interval
0.00	5666	28.60
0.01-0.10	2552	12.89
0.11-0.20	5758	29.15
0.21-0.30	2266	11.42
0.31-0.40	1758	8.86
0.41-0.50	1112	5.62
0.51-0.60	106	0.53
0.61-0.70	88	0.44
0.71-0.80	32	0.16
0.81-0.90	12	0.06
0.91-1.00	450	2.27

6.3 The Document File.

In the simulation model, a document file is composed of a number of documents. The single document file that was generated for the experiments described in this chapter contained 150 documents. Associated with each document in the file are a number of document representations such as abstracts, index term sets, words in the title of a document, subject headings, etc. In the document file that was generated (file D01) a maximum of five representations were generated by the simulation program.

The parameters and values of the one generated document file are listed in Table 6. Table 7 presents an analysis of the file after it was generated. Each of the 150 documents generated had an average of 4.15 representations associated with it out of a possible 5. The total number of surrogates in the file was 622. Row 1 in Table 7 gives the total number of terms for each of the generated surrogates. The 142 surrogate number 1's had a total of 2974 terms associated with them. Similarly there were 87 surrogate number 5's generated in the 150 document collection and the aggregate number of terms for these representations was 178. The total number of terms in the document file as a whole is 5935 for an average of 9.54 terms per document.

In order to further characterize the properties of the document file, an analysis was made of the relative proportion of high and low frequency terms in each of the representations in the document file. For purposes of analysis, all the terms in representation number 1 of document number 1 are grouped with the terms of representation number 1 of document number 2 and representation 1 of document 3, etc. The grouping is done for

Table 6

Parameters of Document File D01

Parameter		Value	For explanation see Section
Number of documents generated		150	5.4
Maximum number of representations generated per document		5	5.4
Probability of representation being generated	1	0.90	5.4
	2	0.95	
	3	0.80	
	4	0.99	
	5	0.70	
Mean number of words in representation number	1	20	5.4.1
	2	12	
	3	6	
	4	4	
	5	2	
Standard deviation of number of words in representation number	1	10	5.4.1
	2	2	
	3	2	
	4	3	
	5	1	
Starting fraction of terms		$p_s = 0.50$	5.4.2
Threshold probabilities for picking most highly associated derivative word		$p_{t_1} = 0.60$	5.4.2
		$p_{t_2} = 0.50$	
		$p_{t_3} = 0.40$	
		$p_{t_4} = 0.30$	
Transition probabilities		$p_{m_2} = 0.60$	5.4.3
		$p_{m_3} = 0.40$	
		$p_{m_4} = 0.70$	
		$p_{m_5} = 0.70$	

Table 7

Analysis of Document File D01

	Document Representation Number				
	1	2	3	4	5
Total number of terms in representation	2974	1536	748	499	178
Total number of times representation present in document	142	142	128	123	87
Fraction of time representation present in document	0.95	0.95	0.86	0.82	0.58
Mean number of terms per representation	20.94	10.81	5.84	4.08	2.05
Standard deviation of number of terms per representation	8.91	2.25	2.04	2.22	0.80

each representation and in addition for the collection as a whole. Then the total set of terms for each surrogate is sorted by frequency of occurrence, and each individual frequency is normalized by dividing it by the total number of occurrences for all terms. The graphs in Figures 9 through 14 display the relative frequency of terms with a given rank. Since it is possible to have a number of terms with the same relative frequency, the curve fitted to the points bisects the horizontal set of points for a given relative frequency if there is more than one point at the relative frequency level. In addition it should be recognized that the curves are continuous approximations to a discrete process.

In general, the curves in Figures 9 through 14 are very similar. The skewness of all the curves demonstrates that the document generation process selects words for inclusion in a document representation such that a rank frequency pattern occurs which is similar to a 'real' document rank frequency pattern. (See [62] as an example). Figure 14, which plots all 5935 terms for all representations, can be considered the limit of the term distribution for document file D01. Figure 9, which displays the distribution of terms in the first representation and which has the next largest number of terms, comes the closest to approximating Figure 14.

Figure 9

File D01 Representation 1 Rank Frequency Distribution

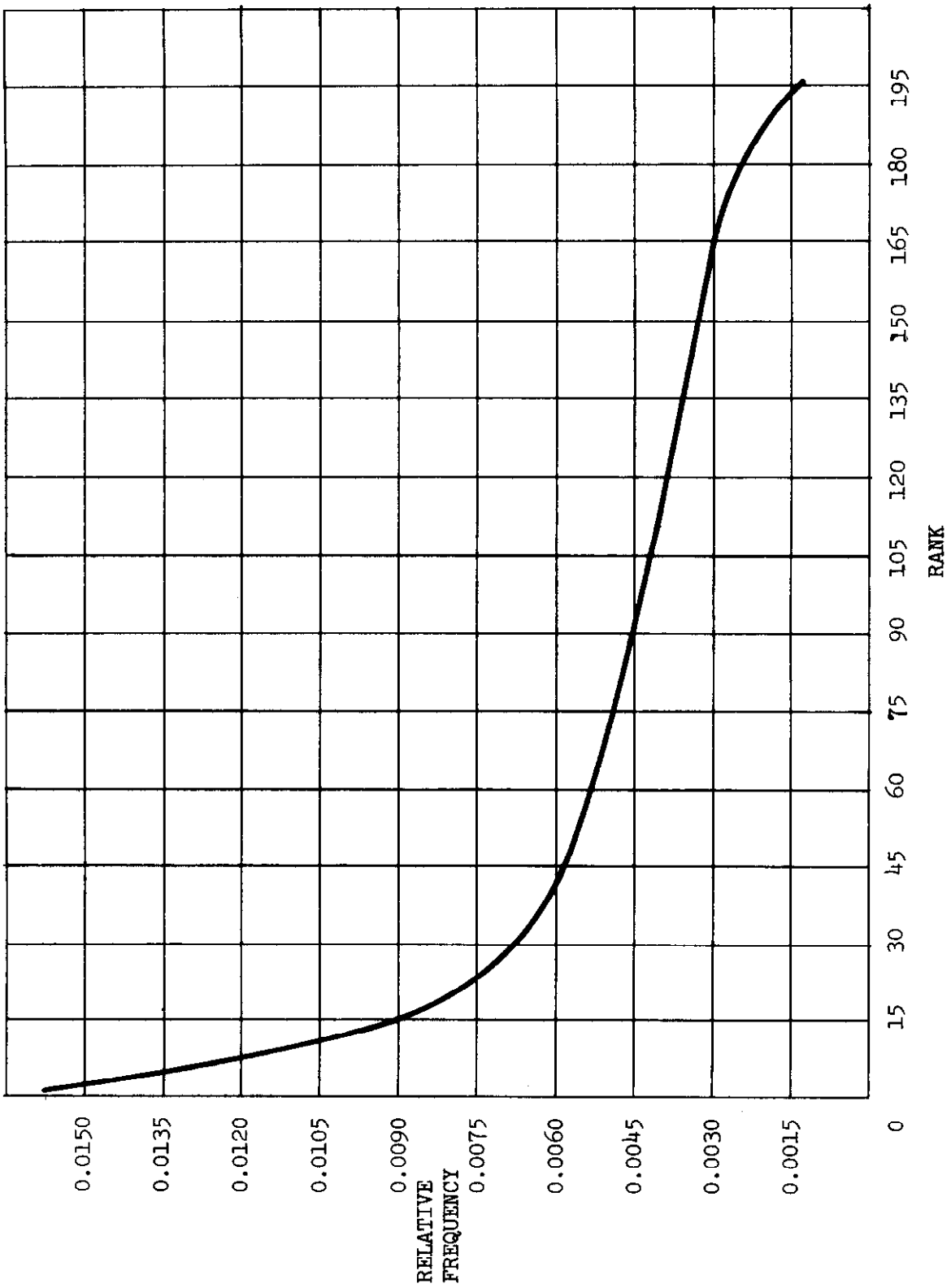


Figure 10

File D01 Representation 2 Rank Frequency Distribution

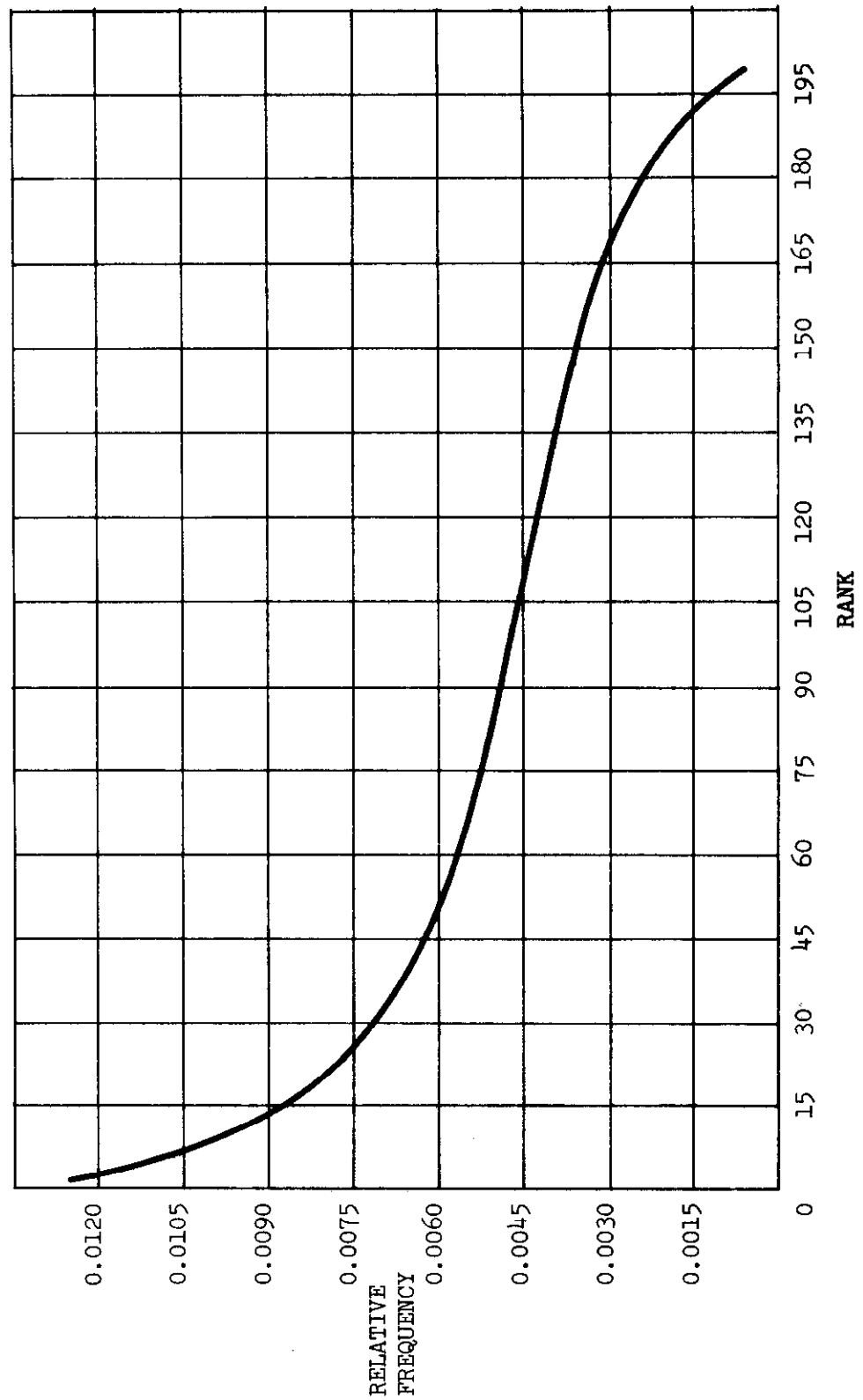


Figure 11

File D01 Representation 3 Rank Frequency Distribution

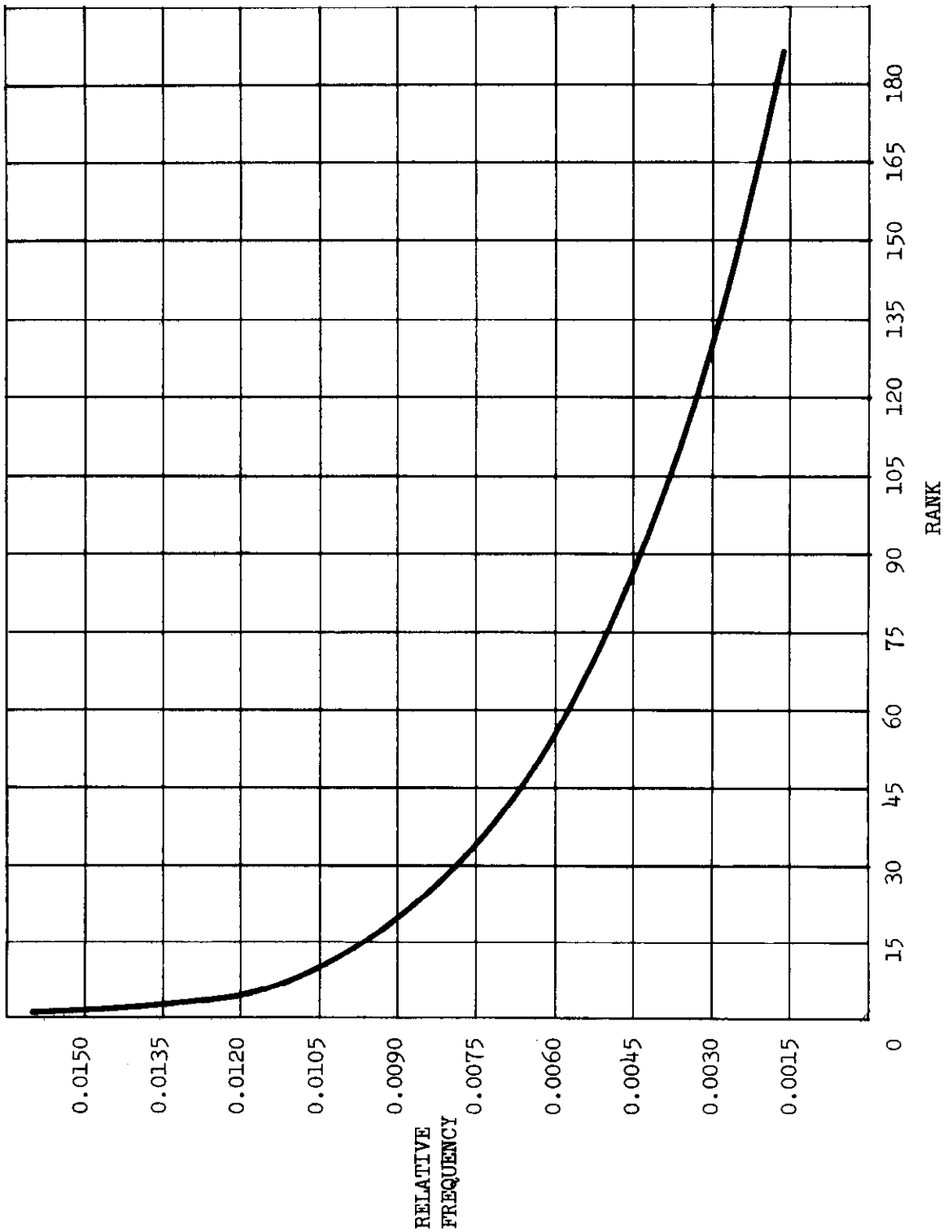


Figure 12

File D01 Representation 4 Rank Frequency Distribution

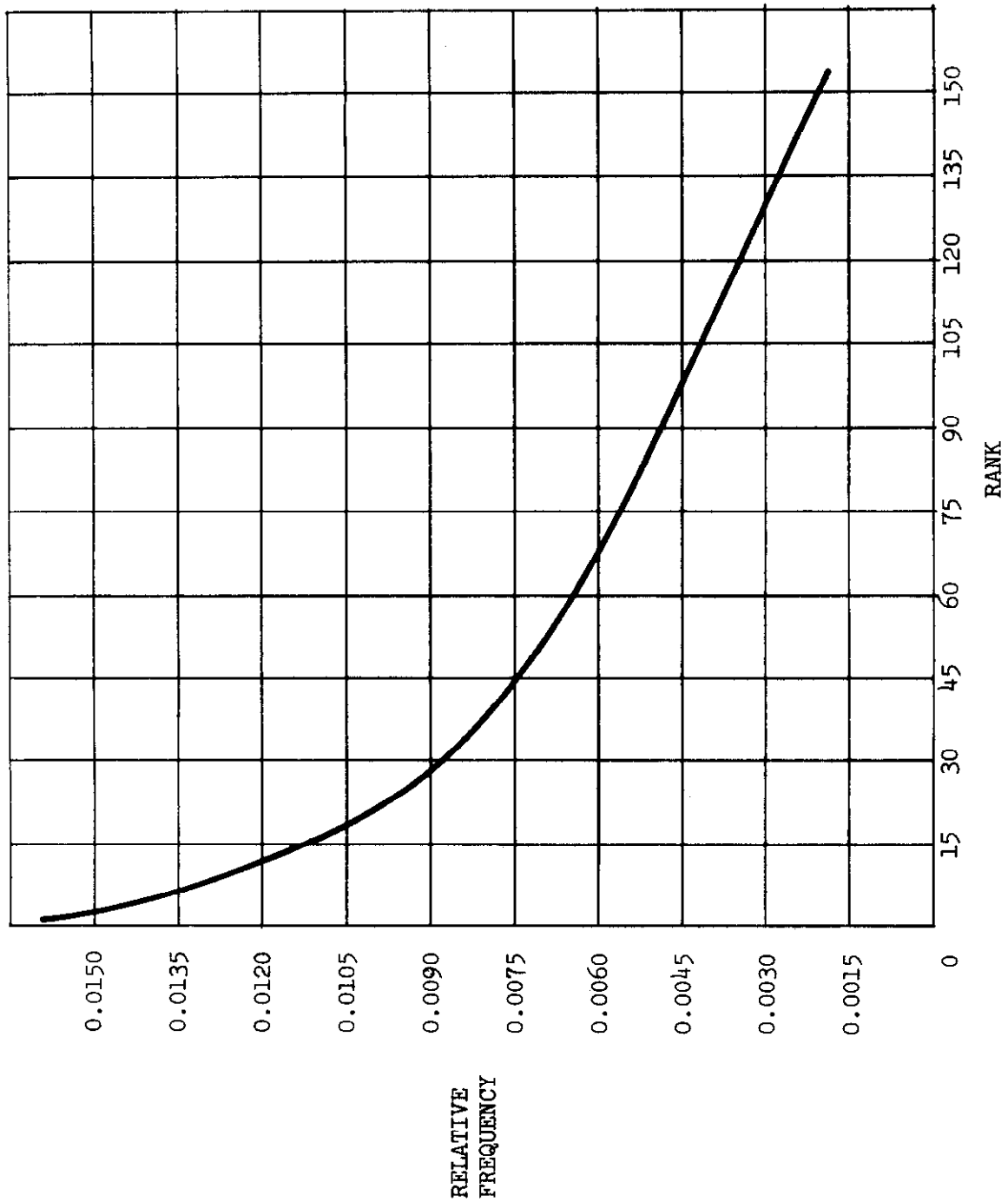


Figure 13

File D01 Representation 5 Rank Frequency Distribution

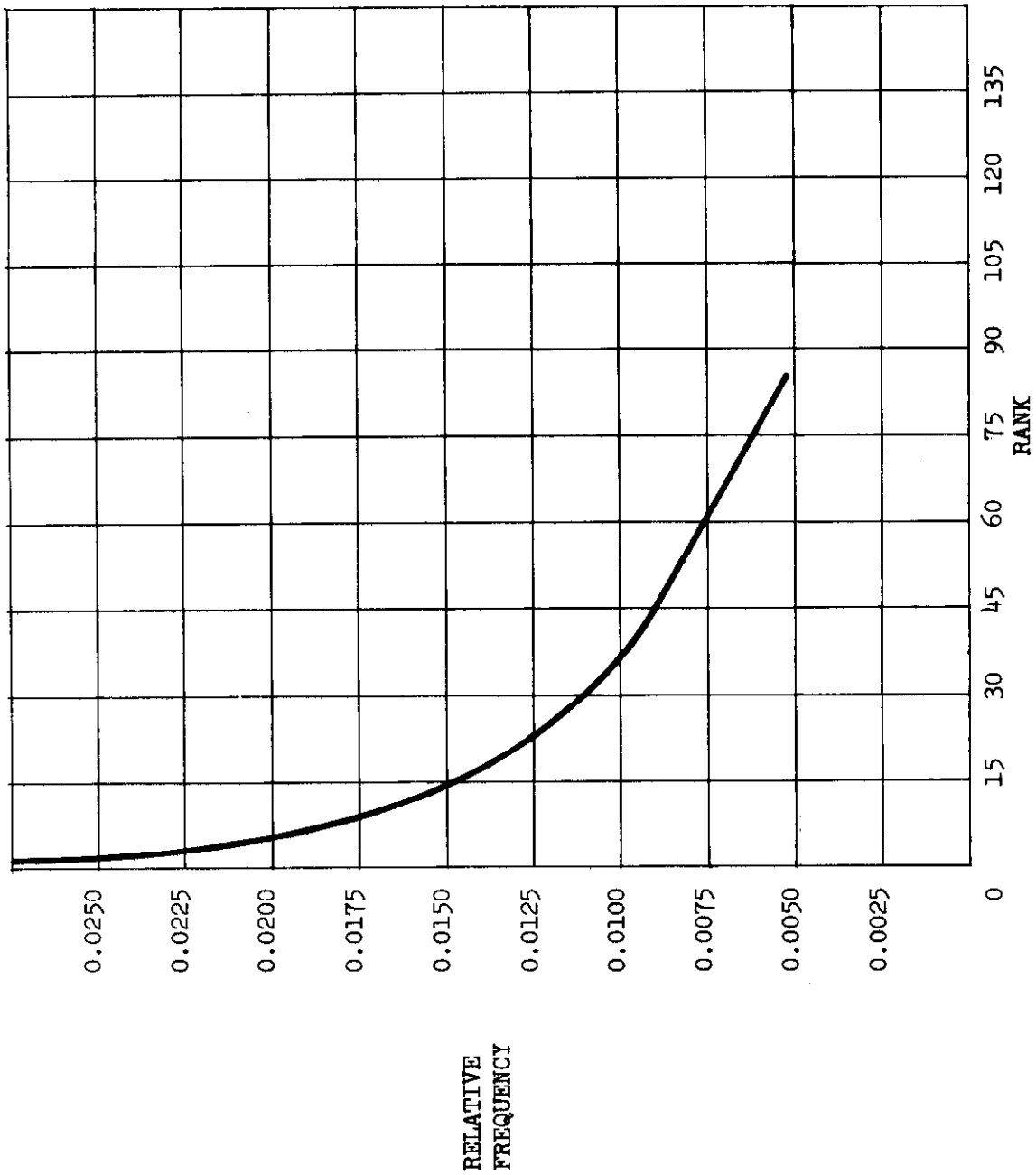
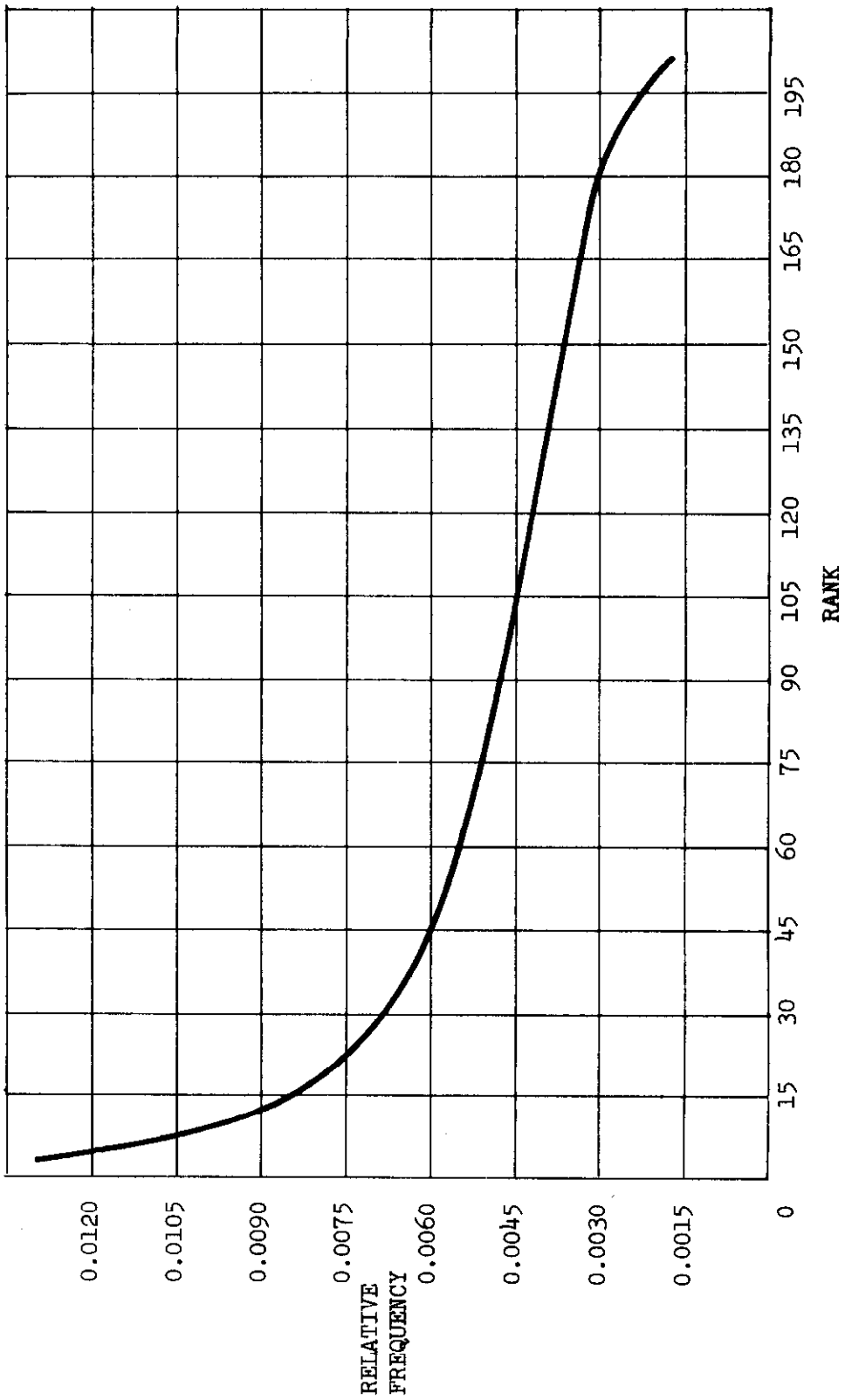


Figure 14

File D01 Overall Rank Frequency Distribution



6.4 The Query Files.

Twenty-two query files were generated for the simulation experiments. In the following sections the experimental design employed in the experiments is discussed. Then an analysis of the generated query files and the term distributions in those files is presented.

6.4.1 Experimental Design.

In Table 8 the parameters and values for each of the generated query files is displayed. The twenty-two runs can be grouped into three categories.

Query file Q01 is established as the normative run for the experiments. The values of the parameters used to generate Q01 are the closest to those used to generate the document file D01. In addition, these values constitute mid-range values for each of the parameters.

Files Q02 through Q17 are designed to test the effect of changing each of eight parameters involved in the query generation process. (The number of queries generated in each query file was held constant for all twenty-two query files.) For example, in the generation of files Q02 and Q03, the parameter that is varied is the starting fraction of terms. (See Tables 8(a) and 8(b) Parameter 9.) In file Q02 the starting fraction is 0.30. By comparing the results from files Q02 and Q03 it will be possible to make inferences about the effect of a change in the starting fraction on the quantity of material retrieved. Similar changes are made in files Q04 and Q05 where the only change is the

Table 8(a)

Query File Parameters and Experimental Design

		For explanation see Section		
	Query file number		Q01	Q02
	Parameter number changed for this run		Norm	9
1.	Number of queries generated		75	75
2.	Mean number of queries/subsets	5.5	4	4
3.	Standard deviation of number of queries/subsets	5.5	2	2
4.	Mean query length	5.5.1	5	5
5.	Standard deviation of query length	5.5.1	2	2
6.	Threshold probability for picking most highly associated derivative word	5.5.2		
	p_{t_1}		0.60	0.60
	p_{t_2}		0.50	0.50
	p_{t_3}		0.40	0.40
	p_{t_4}		0.30	0.30
	p_{t_5}		-	-
7.	Transition probability base query to next query	5.5.2	0.50 for all subsets	0.50 for all subsets
8.	Rule for picking base query	5.5.2	Random selection	Random selection
9.	Starting fraction of terms (p_s)	5.5.1	0.50	0.80

Table 8(b)

Query File Parameters and Experimental Design

	Query file number	Q03	Q04	Q05	Q06
	Parameter number changed for this run	9	6	6	7
1.	Number of queries generated	75	75	75	75
2.	Mean number of queries/subset	4	4	4	4
3.	Standard deviation of number of queries/subset	2	2	2	2
4.	Mean query length	5	5	5	5
5.	Standard deviation of query length	2	2	2	2
6.	Threshold probability for picking most highly associated derivative word				
	p_{t_1}	0.60	0.40	0.90	0.60
	p_{t_2}	0.50	0.30	0.80	0.50
	p_{t_3}	0.40	0.20	0.70	0.40
	p_{t_4}	0.30	0.10	0.60	0.30
	p_{t_5}	-	-	-	-
7.	Transition probability base query to next query	0.50 for all subsets	0.50 for all subsets	0.50 for all subsets	Prob. dist. range .9-.6
8.	Rule for picking base query	Random selection	Random selection	Random selection	Random selection
9.	Starting fraction of terms (p_s)	0.30	0.50	0.50	0.50

Table 8(c)

Query File Parameters and Experimental Design

	Query file number	Q07	Q08	Q09	Q10
	Parameter number changed for this run	7	8	8	5
1.	Number of queries generated	75	75	75	75
2.	Mean number of queries/subset	4	4	4	4
3.	Standard deviation of number of queries/subset	2	2	2	2
4.	Mean query length	5	5	5	5
5.	Standard deviation of query length	2	2	2	1
6.	Threshold probability for picking most highly associated derivative word				
	p_{t_1}	0.60	0.60	0.60	0.60
	p_{t_2}	0.50	0.50	0.50	0.50
	p_{t_3}	0.40	0.40	0.40	0.40
	p_{t_4}	0.30	0.30	0.30	0.30
	p_{t_5}	-	-	-	-
7.	Transition probability base query to next query	Prob. dist. range .4-.1	0.50 for all subsets	0.50 for all subsets	0.50 for all subsets
8.	Rule for picking base query	Random selection	Prob. Dist. newest	Prob. Dist. oldest	Random selection
9.	Starting fraction of terms (p_s)	0.50	0.50	0.50	0.50

Table 8(d)

Query File Parameters and Experimental Design

	Query file number	Q11	Q12	Q13	Q14
	Parameter number changed for this run	5	4	4	3
1.	Number of queries generated	75	75	75	75
2.	Mean number of queries/subset	4	4	4	4
3.	Standard deviation of number of queries/subset	2	2	2	1
4.	Mean query length	5	8	3	5
5.	Standard deviation of query length	4	2	2	2
6.	Threshold probability for picking most highly associated derivative word				
	p_{t_1}	0.60	0.60	0.60	0.60
	p_{t_2}	0.50	0.50	0.50	0.50
	p_{t_3}	0.40	0.40	0.40	0.40
	p_{t_4}	0.30	0.30	0.30	0.30
	p_{t_5}	-	-	-	-
7.	Transition probability base query to next query	0.50 for all subsets	0.50 for all subsets	0.50 for all subsets	0.50 for all subsets
8.	Rule for picking base query	Random selection	Random selection	Random selection	Random selection
9.	Starting fraction of terms (p_s)	0.50	0.50	0.50	0.50

Table 8(e)

Query File Parameters and Experimental Design

	Query file number	Q15	Q16	Q17	Q18
	Parameter number changed for this run	3	2	2	6,9
1.	Number of queries generated	75	75	75	75
2.	Mean number of queries/subset	4	2	6	4
3.	Standard deviation of number of queries/subset	3	2	2	2
4.	Mean query length	5	5	5	5
5.	Standard deviation of query length	2	2	2	2
6.	Threshold probability for picking most highly associated derivative word				
	p_{t_1}	0.60	0.60	0.60	0.40
	p_{t_2}	0.50	0.50	0.50	0.30
	p_{t_3}	0.40	0.40	0.40	0.20
	p_{t_4}	0.30	0.30	0.30	0.10
	p_{t_5}	-	-	-	0.05
7.	Transition probability base query to next query	0.50 for all subsets	0.50 for all subsets	0.50 for all subsets	0.50 for all subsets
8.	Rule for picking base query	Random selection	Random selection	Random selection	Random selection
9.	Starting fraction of terms (p_s)	0.50	0.50	0.50	0.20

Table 8(f)

Query File Parameters and Experimental Design

	Query file number	Q19	Q20	Q21	Q22
	Parameter number changed for this run	6,7	7,9	6,7,9	6,7,8
1.	Number of queries generated	75	75	75	75
2.	Mean number of queries/subset	4	4	4	4
3.	Standard deviation of number of queries/subset	2	2	2	2
4.	Mean query length	5	5	5	5
5.	Standard deviation of query length	2	2	2	2
6.	Threshold probability for picking most highly associated derivative word				
	p_{t_1}	0.40	0.60	0.40	0.40
	p_{t_2}	0.30	0.50	0.30	0.30
	p_{t_3}	0.20	0.40	0.20	0.20
	p_{t_4}	0.10	0.30	0.10	0.10
	p_{t_5}	0.05	-	0.05	0.05
7.	Transition probability base query to next query	Prob. dist. range .9-.6	Prob. dist. range .9-.6	Prob. dist. range .9-.6	Prob. dist. range .9-.6
8.	Rule for picking base query	Random selection	Random selection	Random selection	Prob. dist. newest
9.	Starting fraction of terms (p_s)	0.50	0.20	0.20	0.50

threshold probability for picking the most highly associated derivative word; Q06 and Q07 where the transition probabilities are varied: Q08 and Q09 where the rule for picking the base query is varied; etc. Each of the pairs of runs test the effect of one parameter change on retrieval results.

The third category of runs is the follow-on experiments. Files Q18 through Q22 were generated to test the way in which changes in two variables (Q18, Q19, and Q20) and finally three variables (Q21 and Q22) affect the quantity of material retrieved.

The experiments can also be grouped in another way. It is possible to divide the query parameters into two classes. First, there are those parameters which have to do with the length of a query or the number of queries in a subset (i.e. parameters 1, 2, 3, 4, and 5 in Table 8). On the other hand there are the parameters that influence query structure. (Parameters 6, 7, 8, and 9 in Table 8). The initial experiments (Q01 to Q17) are concerned with both classes of parameters. The follow-on experiments of Q18 to Q22 are primarily concerned with exploring relationships about query structure.

Before proceeding with the analysis of the query files, it may be useful to elaborate on the abbreviated descriptions of the values of some of the parameters in Table 8. Parameter number 7 is the probability distribution that determines the extent to which words in the base query will be in the query currently being generated. This is explained in detail in Section 5.5.2. The notation '0.50 for all subsets' means that the transition probability will be 0.50 for all derivative queries generated in all subsets of the file. In the case of file Q06, for example, the notation 'prob. dist. range 0.90-0.60' means that the

transition probability for a given subset is specified and that for the run these probabilities for all subsets are in the range specified.

The method of picking a base query for use in generating derivative queries (parameter 8) is discussed in Section 5.5.2. In the experiments described in Table 8, two different rules are used for the selection of a base query. The 'random' selection rule uses a uniform probability distribution to pick the new base query from the previously generated queries. In file Q08, Q09, and Q22, however, a probability distribution is supplied which gives various weightings to the likelihood that a specific query will be selected. In Q08 the probability distribution is such that in a sequence of queries, the query generated chronologically last is more likely to be selected as the base query than the first query generated. In file Q09 the reverse is true.

6.4.2 Some Remarks on the Generated Query Files.

Each of the twenty-two query files that was generated for the simulation experiments is composed of seventy-five queries. Within a query file are a number of subsets of queries. A query subset is made up of a set of queries about a specific subject. A particular query in a query file belongs to only one subset. The number of query subsets in a query file varies from a low of 14 subsets in query file Q17 to a high of 34 subsets in file Q16. The mean number of subsets for all files is 21.11. The mean number of queries per subset varies from 2.21 to 5.35, and the mean number of queries per subset for all files is 3.56. A summary of some of the other properties of the generated

query files is given in Table 9.

Table 10 presents statistical information about the number of word tokens and word types in the query files along with the mean and standard deviation of the number of word tokens per query.

Figures 15 through 25 plot the relative frequency of terms with a given rank for each of the query files. The remainder of this section is devoted to an analysis of the variations in these figures. The emphasis in the analysis will be on determining the effect of query file parameter changes on the rank frequency distributions.

In Figure 15 the rank frequency distribution for words in the normative run, Q01, is presented. Figure 16 shows the word frequency pattern in files Q02 and Q03. In files Q02 and Q03 the parameter that is changed is the starting fraction of terms selected at random from the vocabulary to be included in the query. In file Q02 the starting fraction is 0.80 and in Q03 the starting fraction is 0.30. The effect of a change of 0.50 in the starting fraction of terms causes little effect on the resulting frequency patterns. This is not the case in Figure 17 where a threshold probability change in files Q04 and Q05 causes quite a different frequency distribution. The effect of a higher threshold in Q05 causes a relatively large number of high frequency terms to be generated.

The difference between files Q06 and Q07 is that Q06 has high transition probabilities (Parameter 7, Table 8) while those in Q07 are relatively low. Transition probabilities are used to select words for inclusion in a derivative query in a subset from a base query. The high transition probabilities have the effect of reducing the number of unique terms in the file. This is so because the higher the

Table 9

Query File Analysis

Query file no.	No. of queries generated	No. of subsets generated	Mean no. of queries/ subset	Standard dev. of no. of queries/subset
Q01	75	17	4.41	1.75
Q02	75	21	3.57	1.82
Q03	75	22	3.41	1.68
Q04	75	22	3.41	1.85
Q05	75	23	3.26	2.00
Q06	75	19	3.95	2.01
Q07	75	20	3.75	1.73
Q08	75	20	3.75	1.63
Q09	75	24	3.12	1.28
Q10	75	19	3.95	2.08
Q11	75	22	3.41	1.18
Q12	75	21	3.57	2.04
Q13	75	21	3.57	1.60
Q14	75	21	3.57	1.16
Q15	75	21	3.57	2.64
Q16	75	34	2.21	0.27
Q17	75	14	5.35	1.87
Q18	75	21	3.57	1.65
Q19	75	22	3.41	1.88
Q20	75	20	3.75	1.69
Q21	75	18	4.16	1.58
Q22	75	22	3.41	1.05

Table 10

Query File Term Analysis

Query file no.	No. of word tokens in file	No. of word types in file	Mean no. of tokens per query	Standard dev. of no. of tokens/query
Q01	299	132	3.987	4.233
Q02	286	129	3.813	4.298
Q03	273	122	3.640	3.952
Q04	298	130	3.973	4.170
Q05	244	119	3.253	3.570
Q06	282	95	3.760	4.089
Q07	288	140	3.840	4.209
Q08	297	127	3.960	4.087
Q09	298	125	3.973	4.323
Q10	315	122	4.200	4.214
Q11	341	134	4.547	5.191
Q12	492	162	6.560	6.773
Q13	151	81	2.013	2.187
Q14	277	113	3.693	3.959
Q15	269	113	3.587	3.935
Q16	286	126	3.813	4.190
Q17	281	123	3.747	4.007
Q18	279	116	3.720	3.969
Q19	282	90	3.760	4.025
Q20	267	110	3.560	3.973
Q21	270	87	3.600	3.754
Q22	296	114	3.947	4.174

Figure 15

File Q01 Rank Frequency Distribution

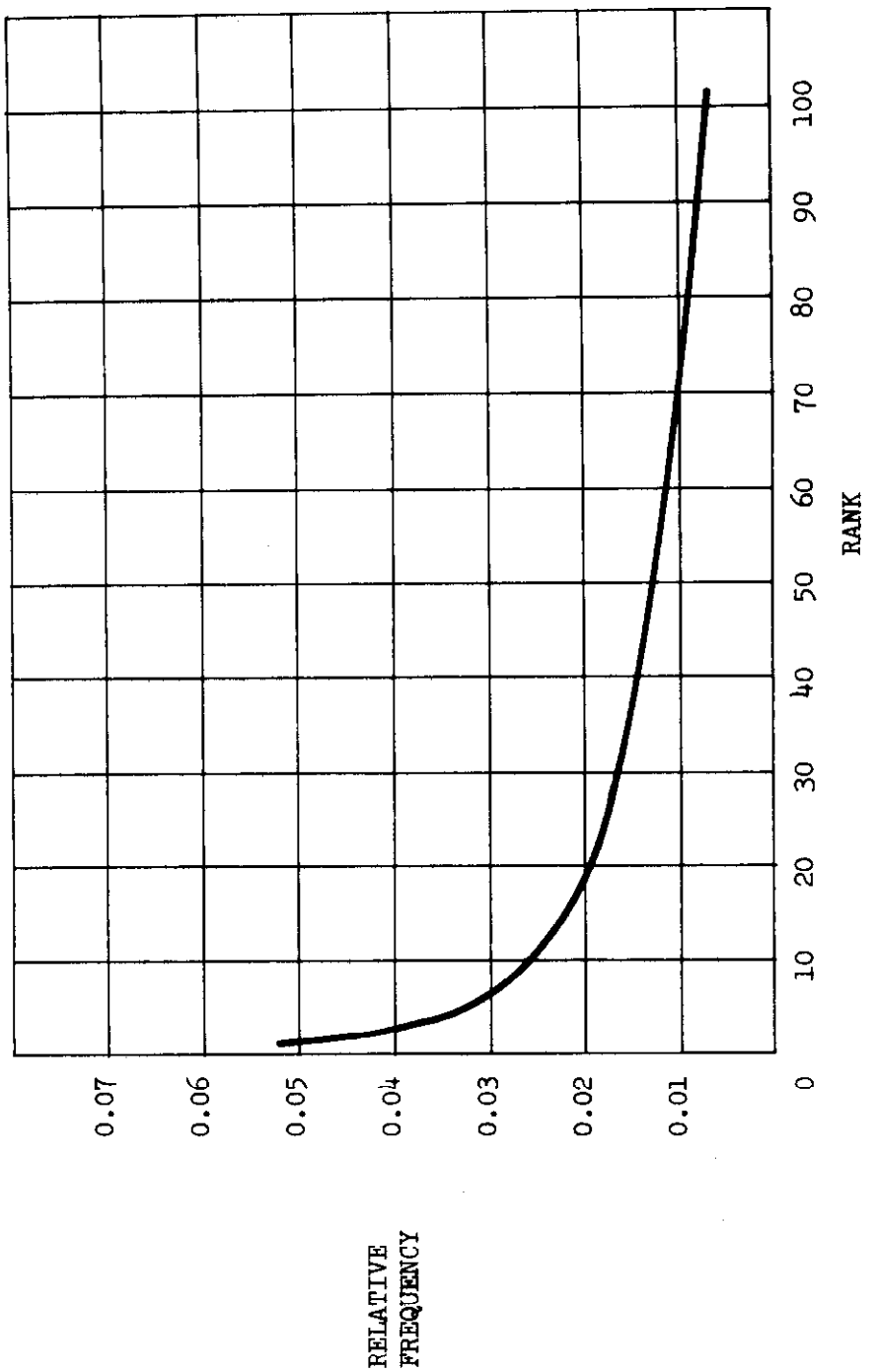


Figure 16

Files Q02-Q03 Rank Frequency Distribution

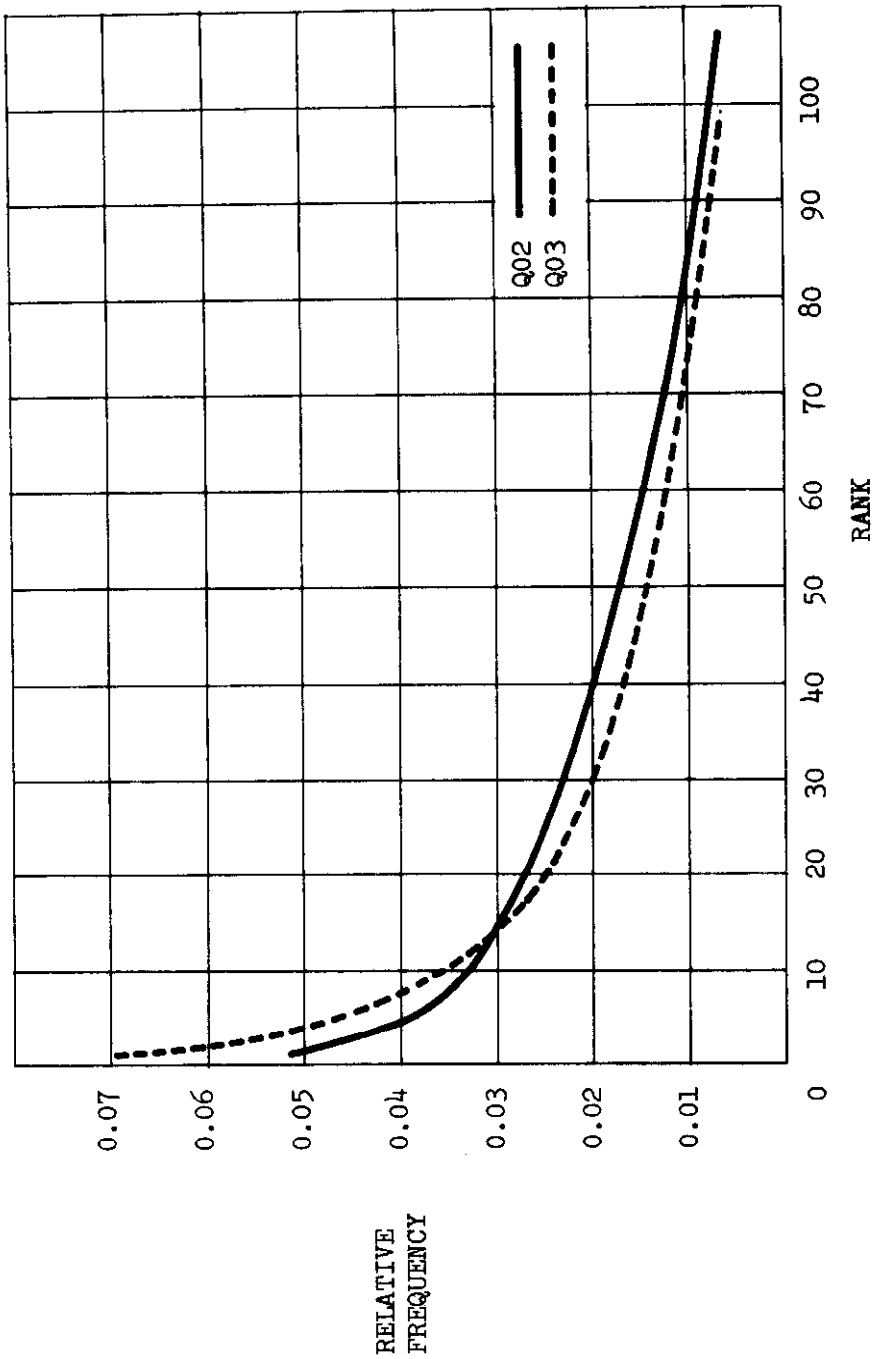
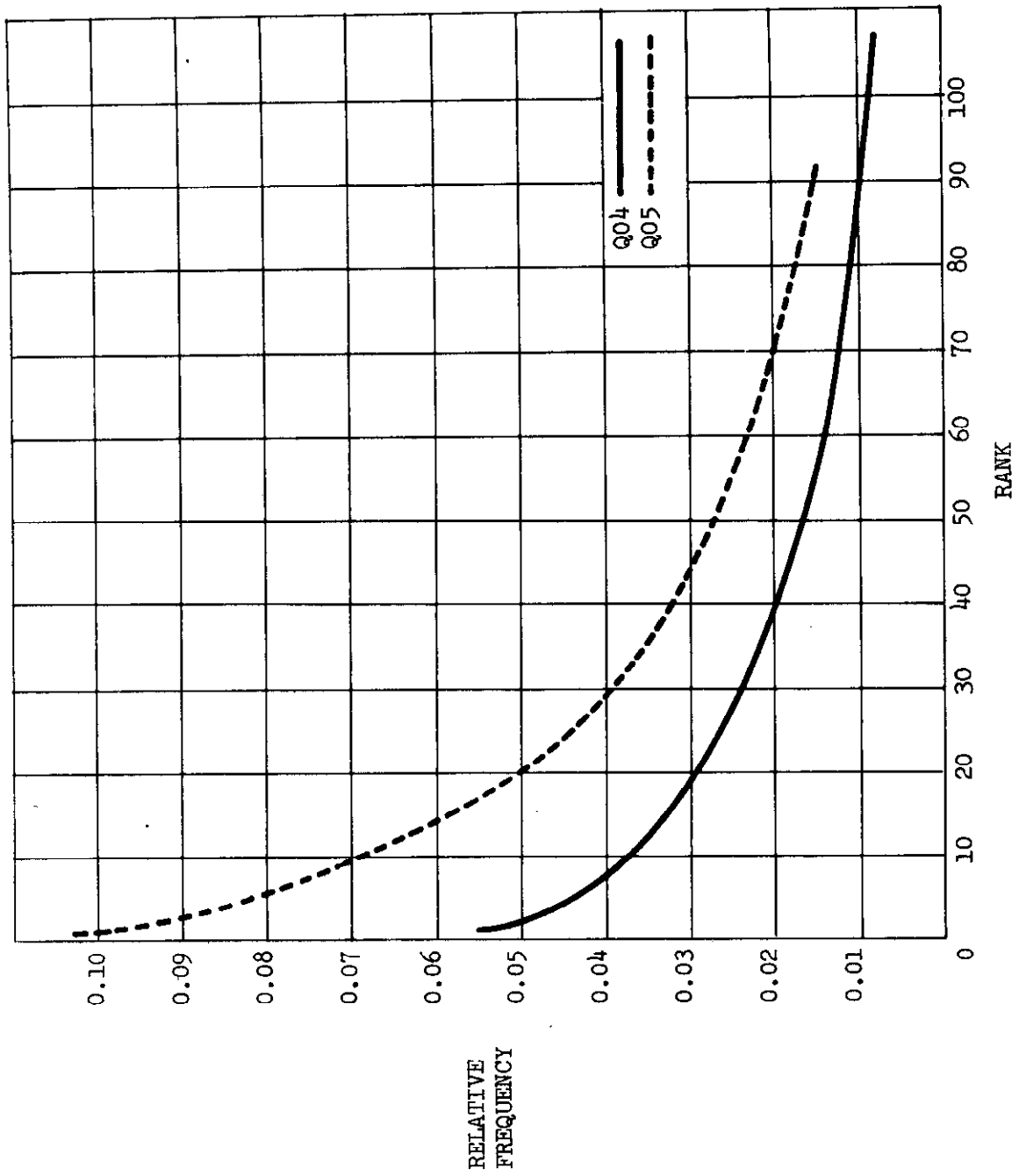


Figure 17

Files Q04-Q05 Rank Frequency Distribution



transition probabilities, the greater the likelihood that words will be transferred from one query to the next in a subset. (Transition probabilities are only used within a subset - not between subsets. A new base query is generated at the start of a new subset.) See Figure 18 and Table 10. File Q07 has low transition probabilities and these low probabilities cause generation of a high number of unique terms.

Generation of a new query subset begins with the generation of a base query. The base query is used in conjunction with transition probabilities to develop derivative queries. When the second query in a subset is generated, the first query is the base query. When the third query is generated, either the first or second query can be used as a base, etc. In file Q08 and Q09 two different rules are used to select a base query. In Q08 there is a greater probability that a query generated chronologically later in a subset will be a base query. In file Q09 the probability is greater that a query generated early in the subset will be a base query. The changes in term frequency patterns caused by the use of a newer or older base query is shown in Figure 19. In file Q09 there are more terms used more frequently than in Q08. If the x-axis of Figure 19 is divided into thirds, then the middle and low rank terms, as shown in the figure, exhibit little difference in relative frequency for files Q08 and Q09.

It may be that in this <u>particular</u> run of Q09, the fact that there are more subsets (24) in Q09 than in Q08 (20) may cause the variation. The number of subsets generated for a particular query file is a function of the mean and standard deviation of the number of queries per subset for each subset in the file. As query generation proceeds for a file, the actual number of queries per subset for the first subset

Figure 18

Files Q06-Q07 Rank Frequency Distribution

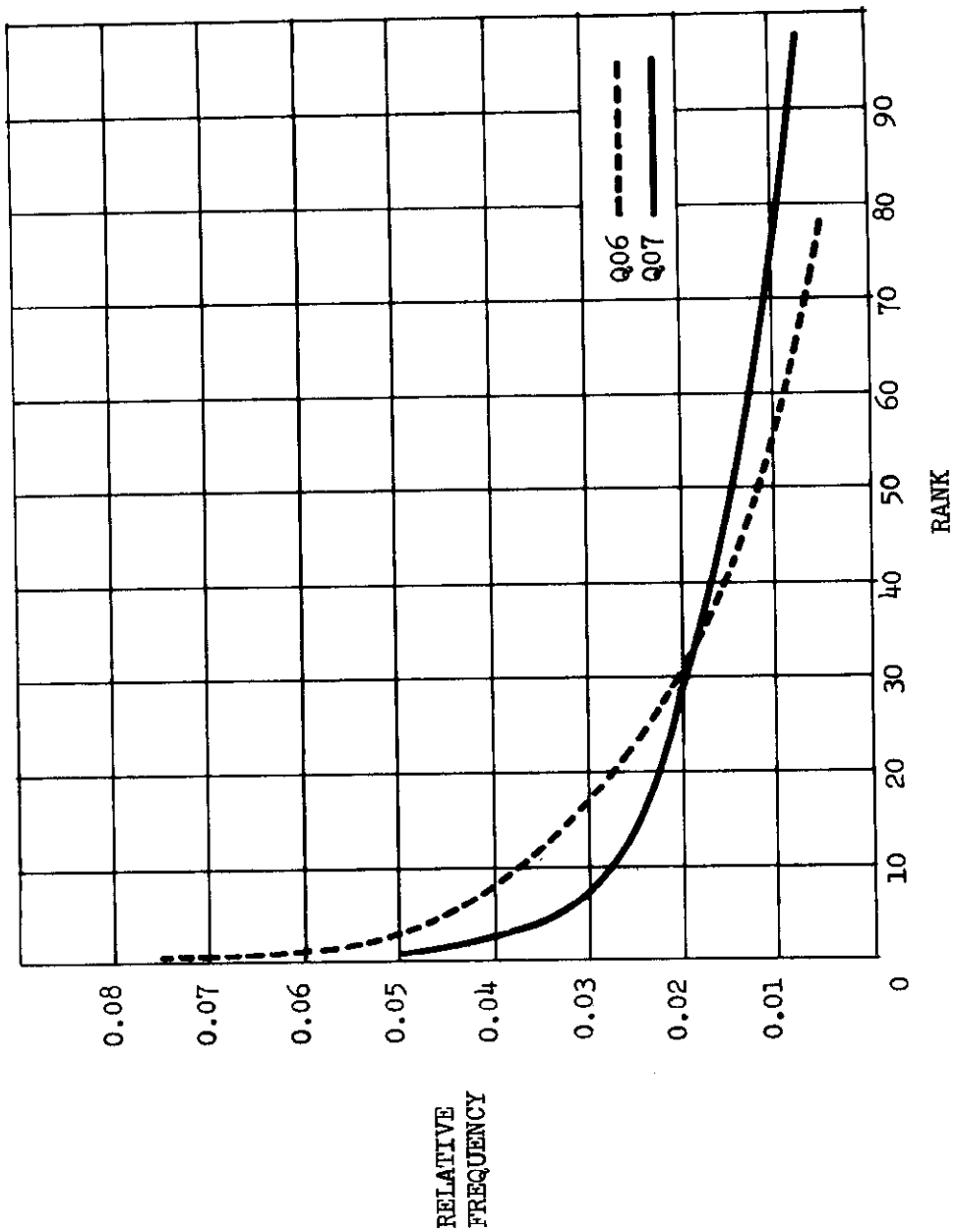
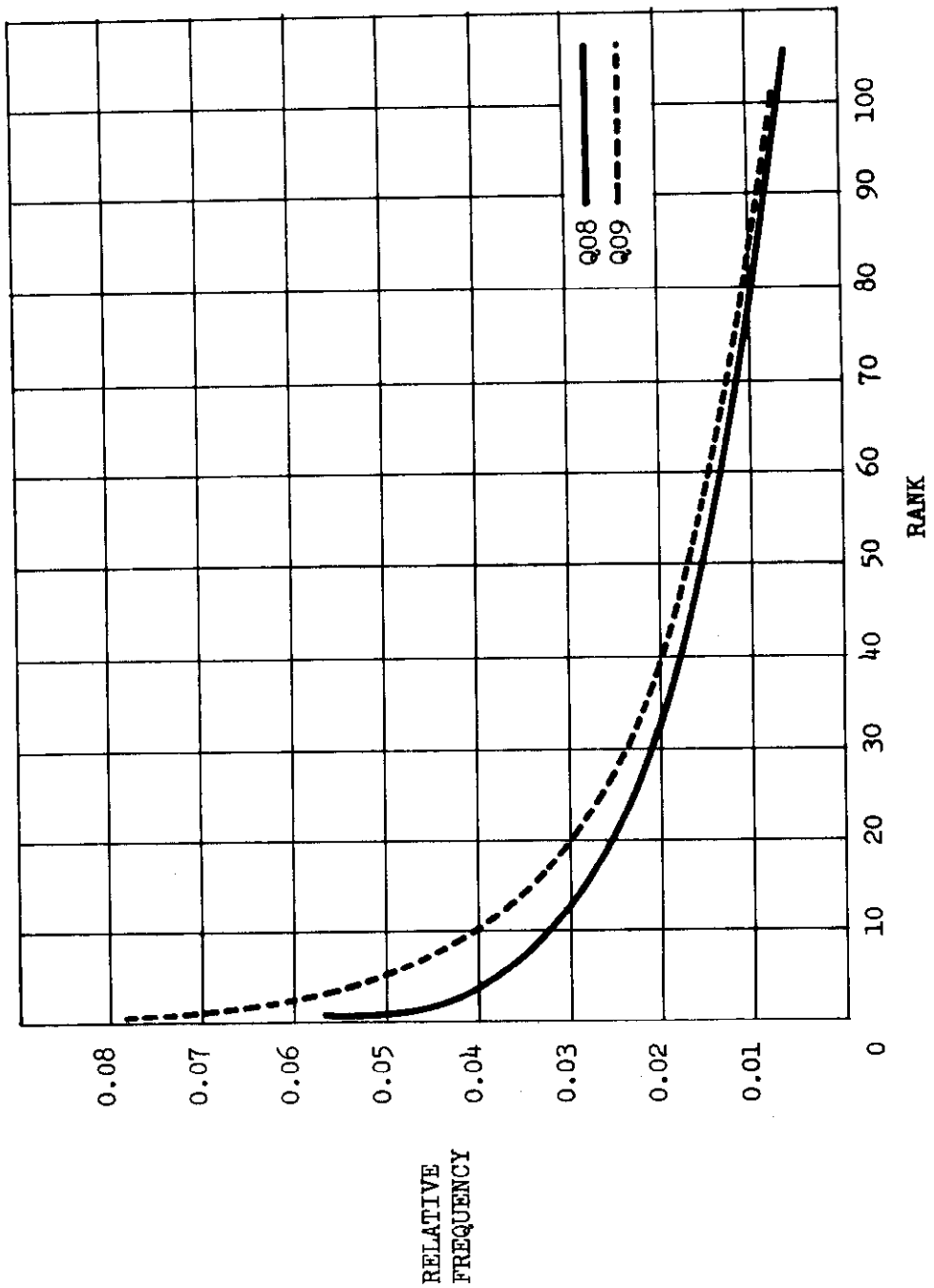


Figure 19

Files Q08-Q09 Rank Frequency Distribution



is calculated using a normal random number generator. Then the required number of queries in the first subset is generated. Before the second subset of queries is generated, the number of queries in that subset is computed using the normal random number generator. The process is repeated until 75 queries have been generated. Thus the number of subsets in a query file depends on the number of queries in each subset and is not a controlled variable as is the limit on the total number of queries to generate in a file.

In files Q10 and Q11 the parameter that is varied is the standard deviation of the number of words in the query. When the standard deviation of the query length is varied, the result is a consistently high relative frequency for terms of similar rank for file Q11. (See Figure 20). File Q11 has a larger standard deviation of query length than file Q10.

Figure 21 shows the rank frequency pattern for files Q12 and Q13. File Q12 has the largest number of word tokens and types in it of any query file. By contrast, Q13 is smallest in both categories. Insofar as the rank frequencies are concerned, Q13 has the most highly skewed distribution of any file. In this file, a small number of terms account for a large proportion of the total term usage. In contrast to the abrupt ending of Q13's distribution, the rank distribution of the terms in Q12 suggests a large number of terms are used infrequently in this file.

In contrast to Figure 21, Figure 22 shows a very similar rank frequency pattern for files Q14 and Q15. The only change between Q14 and Q15 is the value of the standard deviation of the number of queries per subset.

Figure 20

Files Q10-Q11 Rank Frequency Distribution

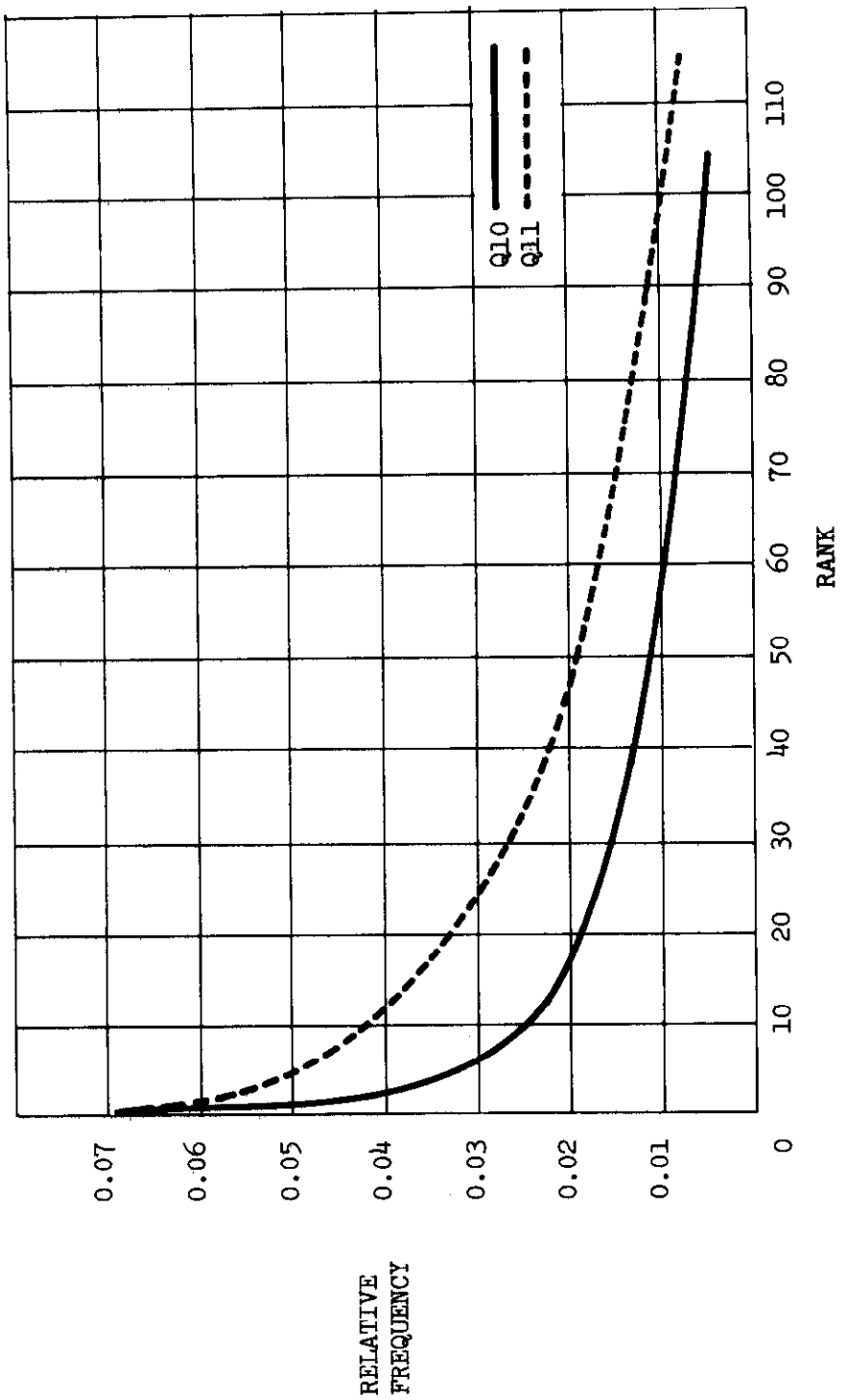


Figure 21

Files Q12-Q13 Rank Frequency Distribution

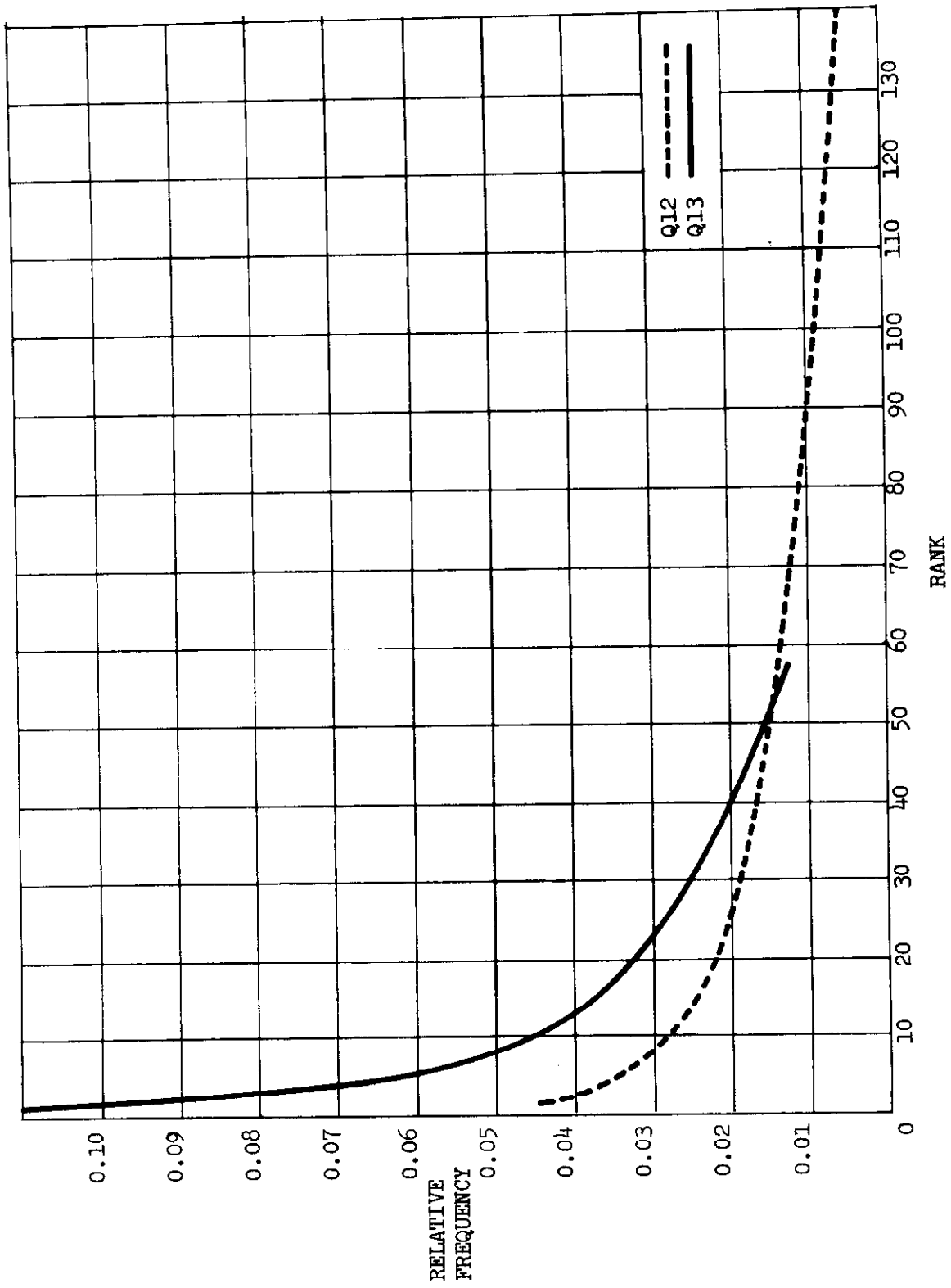
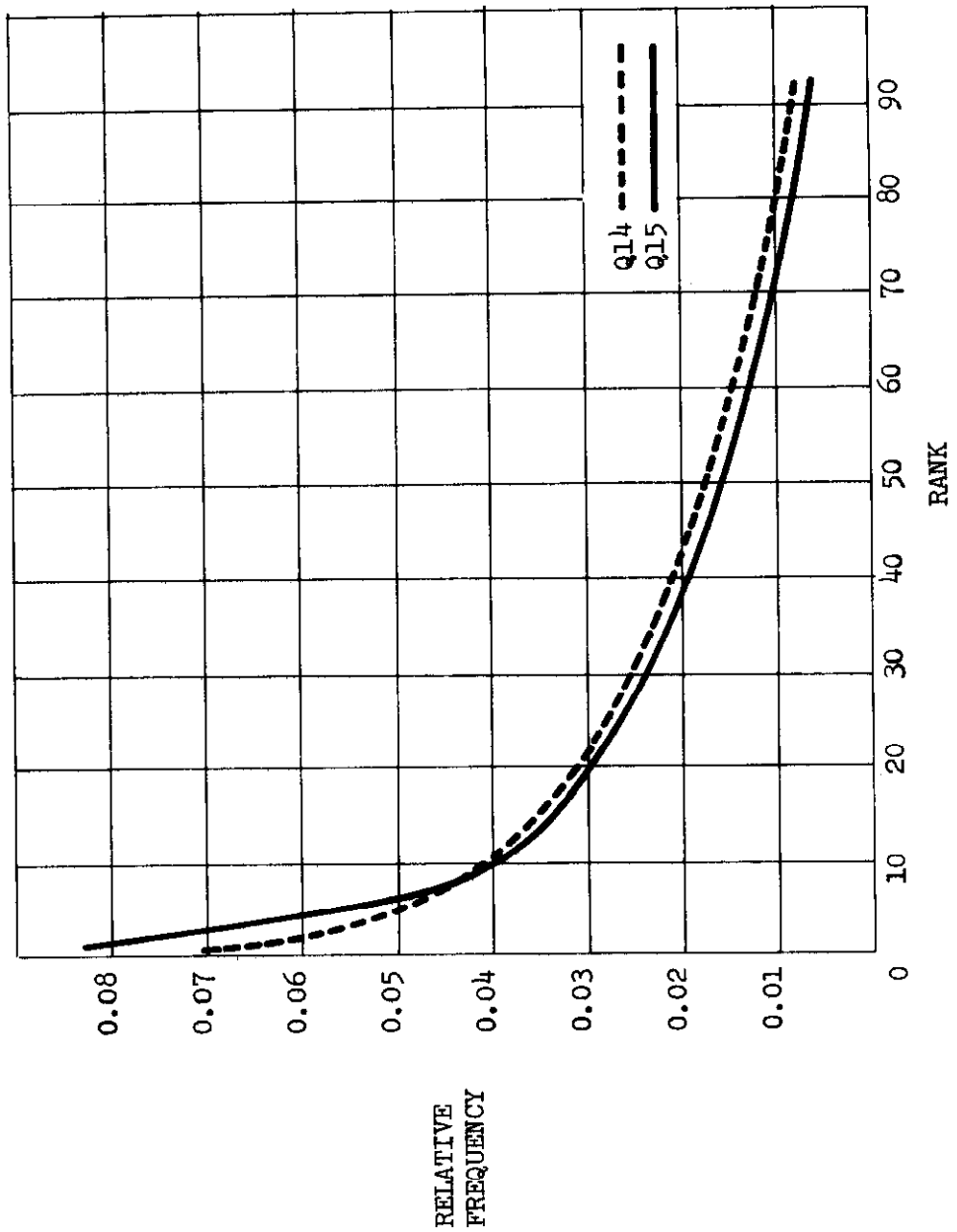


Figure 22

Files Q14-Q15 Rank Frequency Distribution



In query files Q16 and Q17 the variable that is changed is the mean number of queries per subset. Figure 23 plots the rank frequency distribution for these files. Within a query subset the base query is used to generate derivative queries. The process uses transition probabilities to determine if words will be transferred from the base query to the derivative query. The greater the number of queries in a query subset, the greater will be the probability that a word that appears in one query of the subset will appear in another. Also, the greater the likelihood that there will be more words that occur more frequently than if there are a small number of queries per subset in a file. Thus the aggregate rank distribution will be higher the greater the number of subsets in a file. This is confirmed in Figure 23 where the rank frequency of terms in file Q16 is considerably above the rank frequency of terms in Q17.

Figure 24 shows the frequency distributions for the three query files that have two variable interactions involved in their generation. Files Q18, Q19 and Q20 compare a high and low threshold and low starting fraction given a high and normal set of transition probabilities. The only feature that distinguishes these graphs from the single variable experiments of Q03, Q04 and Q06 is the extreme skewness of Q20. This phenomenon is attributed to the high transition probabilities in Q20. The same pattern can be observed in Figure 18 for file Q06. The rank patterns of Q18 and Q19 are very similar to one another. Even though Q19 has high transition probabilities, the effect on the rankings is negated by the low threshold-normal starting proportion that is present.

Figure 23

Files Q16-Q17 Rank Frequency Distribution

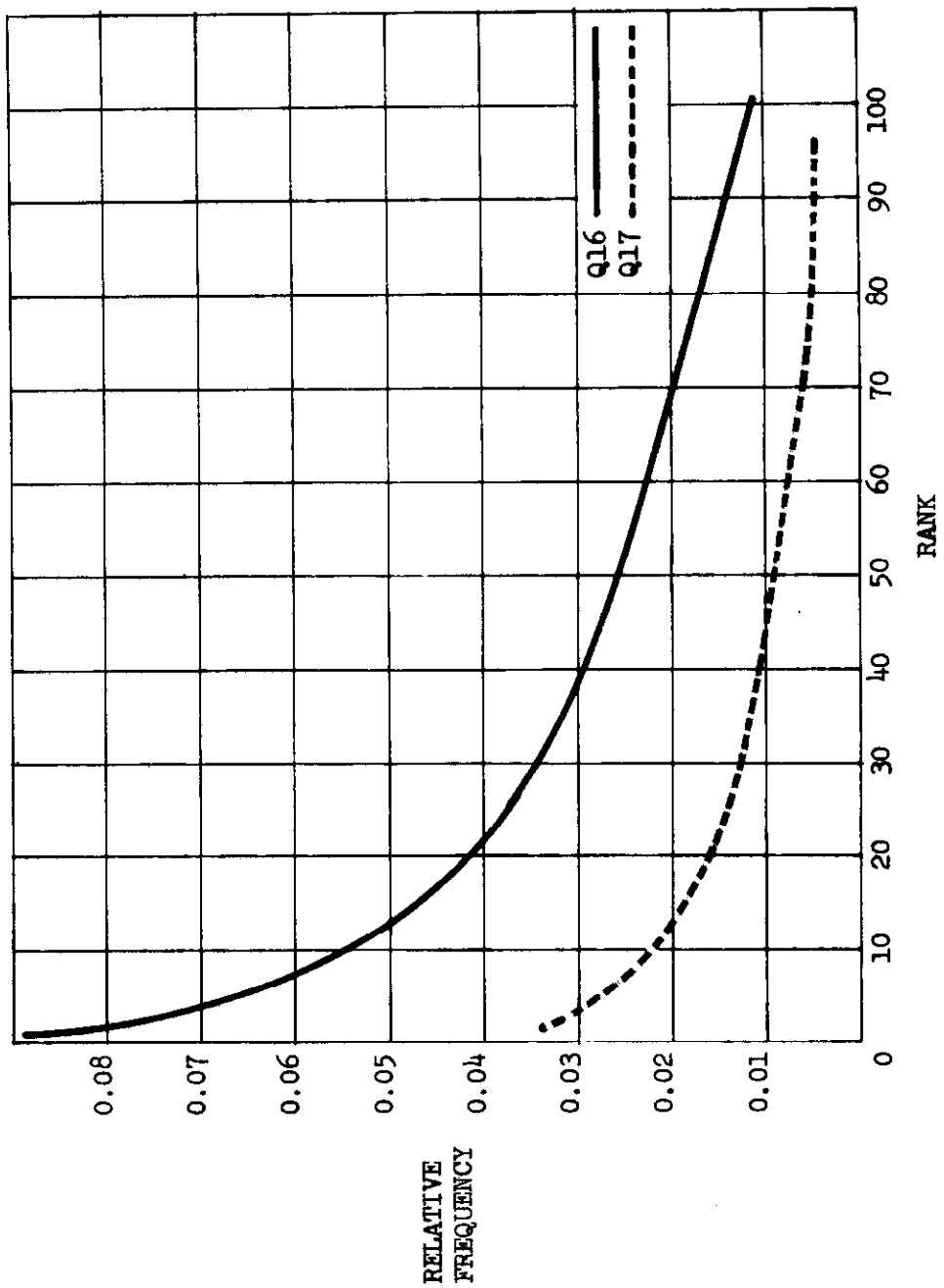


Figure 24

Files Q18-Q19-Q20 Rank Frequency Distribution

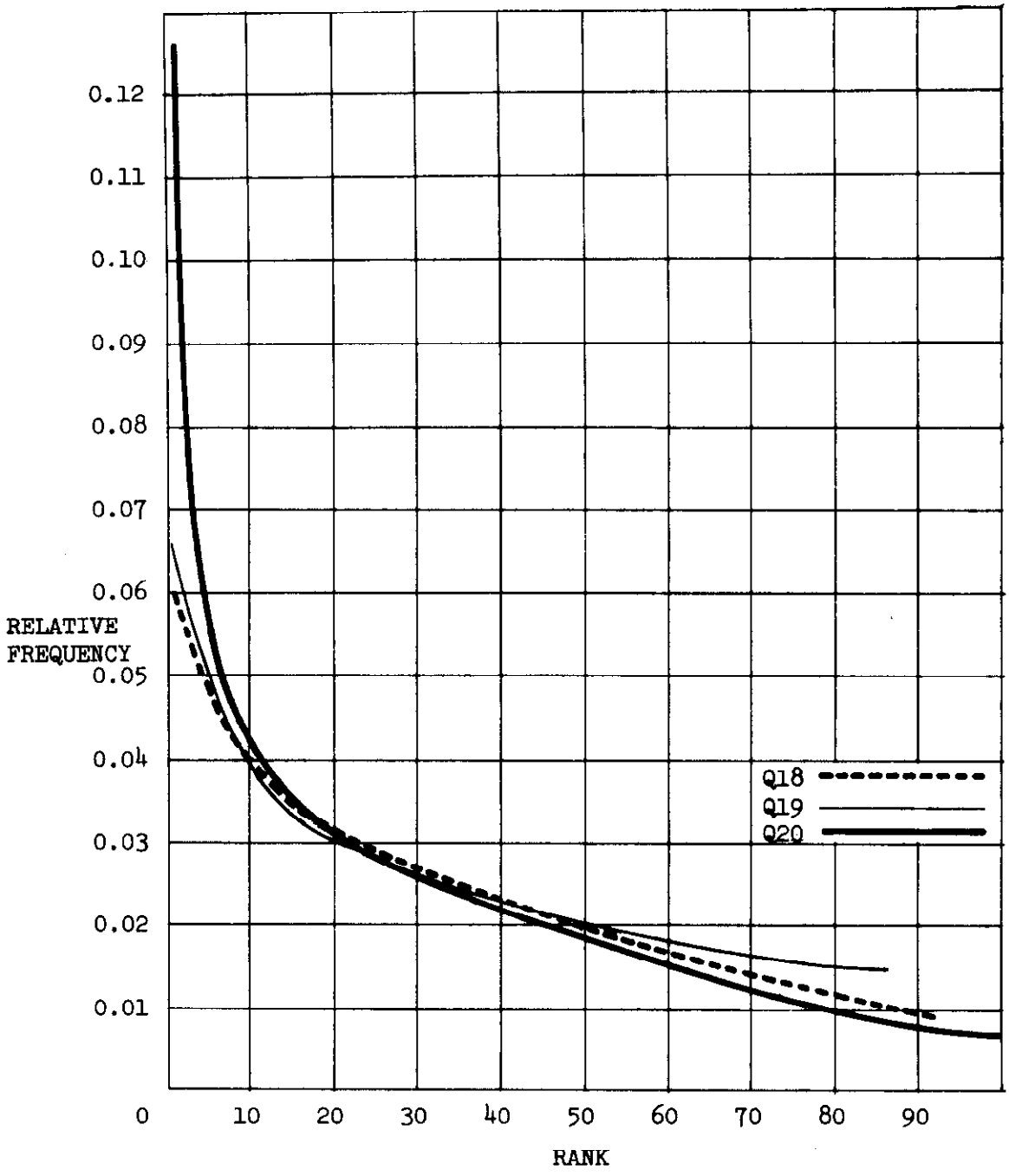
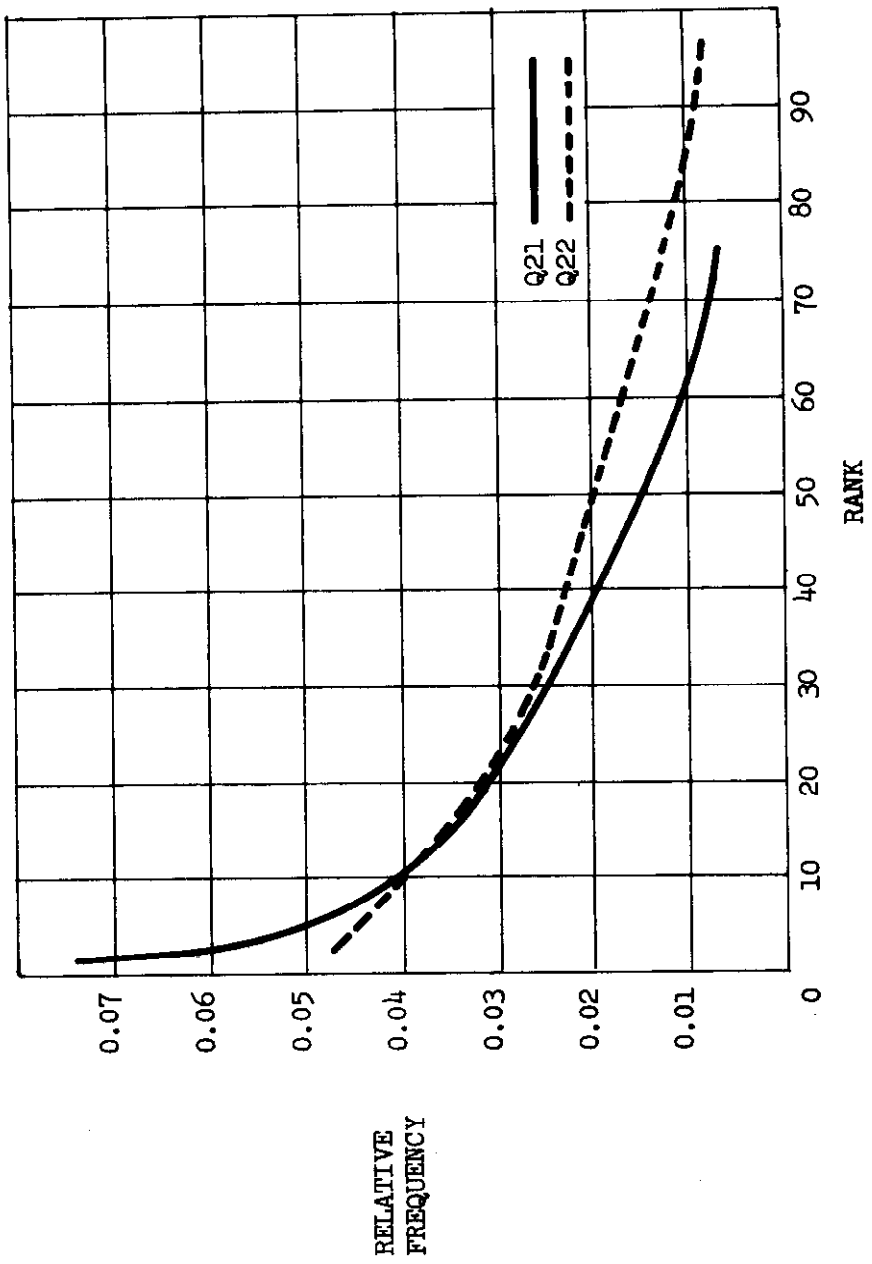


Figure 25

Files Q21-Q22 Rank Frequency Distribution



6.5 Experimental Results.

Evaluation of the retrieval system simulator involves two seperate issues. The first has to do with the adequacy of this particular simulation model and the adequacy of simulation as a technique for evaluating information retrieval systems. These issues are dealt with in Section 7.2. The second part of the evaluation of the retrieval system simulator has to do with evaluating the experimental results from actually simulating a document and query collection. The remainder of this section is devoted to an analysis and synthesis of the data from the simulation study.

A single simulation run, or experiment, using the retrieval system simulator, involves a series of steps. First a thesaurus is generated. Then a number of rules are used (which employ the thesaurus) to generate document representations. Next, another set of rules are used (also employing the thesaurus) to generate queries. Finally, using a retrieval rule, the extent of the match between the queries and the document representations is recorded.

The experiments that are described in this section use one thesaurus (T01), one document file (D01), and twenty-two query files (Q01-Q22). Each experiment involves comparing the document representations in file D01 with the queries in one of the query files. Thus there are twenty-two experiments that are performed. In each of the experiments the same retrieval rule was used to compare queries to document representations. This retrieval rule was an overlap rule which measured the number of terms in common to the query and the representation.

Each of the experimental runs was evaluated in two ways. The primary method for measuring the performance of an experiment was by counting the number of document representations matching a given query. This comparison was done for all queries in a query file and all document representations in the document file. Each run was evaluated by counting the number of searches that resulted in a match of one word, two words, three words, etc. between the queries and the document representations. This information was gathered as a result of applying the overlap retrieval rule to a comparison of document representations and queries.

In the document file D01 there are several different document representations associated with each document. A search of the document file involved comparing a query to all the document representations in the document file. The other method used to evaluate an experimental run was to determine whether there were certain document representations that were retrieved more frequently than others. For each experiment, the number of searches yielding a match with document representation number 1, number 2, number 3, etc. was recorded.

Normally when a retrieval system is evaluated, a user makes judgments about the relevance to his needs of the documents retrieved by the system. The evaluation of the simulated retrieval system did not include evaluation on the basis of relevance. The only criterion used for retrieval was whether a term in the query matched a term in the document representation. It is conceivable that a pseudo-relevance function could have been incorporated into the simulation model. The function would then predict when a document would be relevant to the user's needs. However, it is felt that not enough information is available to characterize

the process. In addition, modeling the aspects of the retrieval system which were considered in this study is by itself a sizeable undertaking and should provide valuable insights.

The tables that are presented later in this section summarize the quantity and proportion of searches that resulted in a match between a document representation and a query. Two points need clarification. It will be recalled that the document file D01 consists of a total of 150 documents comprising 622 representations, distributed as shown in Table 7 row 2. Each of the query files has 75 queries in it. The total number of comparisons that is made for each run is 46,650 (i.e. 622 x 75). In the reported statistics that follow, the total number of searches is always the same - 46,650.

The second point that needs amplification is that of a threshold. When the terms in a query are compared to the terms in a document representation, either there are no terms in common to the query and document representation, or a certain number of terms are common to both. A user formulating a query may require that a certain number of terms match between the query and the surrogate. This quantity is called a threshold. Only document representations that meet or exceed the threshold match requirement are retrieved. In the simulation results that follow the number and proportion of searches that result in a match at each threshold level is recorded.

6.5.1 Evaluation Using the Overlap Rule.

Tables 11 and 12 summarize the results of each of the experiments or runs. (An experiment involves the comparison of a query file to the document file. Experiment E01 compares query file Q01 with document file D01. E07 compares Q07 with D01, etc.) In Table 11 the number of searches for each experiment that resulted in a match between a query and a document representation is recorded. In addition, the number of searches that resulted in no match between query and document representation is recorded. Table 12 presents the same data, only expressed in proportions. For example, Table 11 indicates that in experiment E17 38,463 of 46,650 searches resulted in no matches between the queries in file Q17 and the representations in file D01. That is 0.825 of the searches had zero matches. (See Table 12, run 17.) Similarly, of the remaining searches that involve a hit, 6,916 (or 0.148) found one term in common between query and representation; 1,059 (or 0.023) showed two terms matching; and 185 (or 0.004) found three in common; etc.

Aside from two exceptions (E12 and E13) both the number and proportion of searches resulting in no matches between queries and representations are very similar for all runs. In general, approximately 83% of all searches result in no matches. Since the proportion of searches resulting in no match constitutes such a large quantity, Table 13 was constructed in order to elaborate on the threshold pattern for those searches for which there was a match.

The quantities in Table 13 are derived from Table 11. For each row in Table 11 the number of searches resulting in no match between query

Table 11
Number of Searches Resulting in a Match between
a Query and a Document Representation

Run number	Number of searches resulting in no match between query and doc. representation	Number of searches resulting in a match between a query and a document representation at threshold level							
		1	2	3	4	5	6	7	>7
1	38,306	6985	1140	189	28	2	0	0	0
2	39,162	6302	984	173	23	4	2	0	0
3	38,933	6640	891	166	17	2	1	0	0
4	38,849	6673	950	154	21	3	0	0	0
5	39,745	5838	912	131	18	5	1	0	0
6	38,774	6601	1106	157	12	0	0	0	0
7	38,987	6512	957	167	22	5	0	0	0
8	38,551	6809	1094	162	27	6	0	1	0
9	38,613	6741	1058	193	36	7	2	0	0
10	37,172	7966	1327	166	19	0	0	0	0
11	37,340	7598	1378	247	72	10	3	2	0
12	34,228	9667	2097	497	123	30	6	2	0
13	42,082	4180	352	29	4	2	1	0	0
14	38,935	6522	1015	149	25	4	0	0	0
15	38,794	6595	1034	183	37	6	1	0	0
16	38,520	6628	1233	229	38	2	0	0	0
17	38,463	6916	1059	185	25	2	0	0	0
18	38,681	6796	1018	142	11	2	0	0	0
19	38,531	6756	1134	202	25	1	1	0	0
20	39,443	6008	1012	163	21	2	1	0	0
21	38,778	6876	867	114	15	0	0	0	0
22	38,507	6971	1008	145	15	4	0	0	0

Table 12
Proportion of Searches Resulting in a Match between a Query and a Document Representation

Run number	Proportion of searches resulting in no match between query and doc. representation	Proportion of searches resulting in a match between a query and a document representation at threshold level							
		1	2	3	4	5	6	7	>7
1	0.821	0.150	0.024	0.004	0.001	0.000	0.000	0.000	0.000
2	0.839	0.135	0.021	0.004	0.000	0.000	0.000	0.000	0.000
3	0.835	0.142	0.019	0.004	0.000	0.000	0.000	0.000	0.000
4	0.833	0.143	0.020	0.003	0.000	0.000	0.000	0.000	0.000
5	0.852	0.125	0.020	0.003	0.000	0.000	0.000	0.000	0.000
6	0.831	0.142	0.024	0.003	0.000	0.000	0.000	0.000	0.000
7	0.836	0.140	0.021	0.004	0.000	0.000	0.000	0.000	0.000
8	0.826	0.146	0.023	0.003	0.001	0.000	0.000	0.000	0.000
9	0.828	0.145	0.023	0.004	0.001	0.000	0.000	0.000	0.000
10	0.797	0.171	0.028	0.004	0.000	0.000	0.000	0.000	0.000
11	0.800	0.163	0.030	0.005	0.002	0.000	0.000	0.000	0.000
12	0.734	0.207	0.045	0.011	0.003	0.001	0.000	0.000	0.000
13	0.902	0.090	0.008	0.001	0.000	0.000	0.000	0.000	0.000
14	0.835	0.140	0.022	0.003	0.001	0.000	0.000	0.000	0.000
15	0.832	0.141	0.022	0.004	0.001	0.000	0.000	0.000	0.000
16	0.826	0.142	0.026	0.005	0.001	0.000	0.000	0.000	0.000
17	0.825	0.148	0.023	0.004	0.001	0.000	0.000	0.000	0.000
18	0.829	0.146	0.022	0.003	0.000	0.000	0.000	0.000	0.000
19	0.826	0.145	0.024	0.004	0.001	0.000	0.000	0.000	0.000
20	0.846	0.129	0.022	0.003	0.000	0.000	0.000	0.000	0.000
21	0.831	0.147	0.019	0.002	0.000	0.000	0.000	0.000	0.000
22	0.825	0.149	0.022	0.003	0.000	0.000	0.000	0.000	0.000

and document representation is subtracted from 46,650. The resulting quantity (Table 14 column 7) is divided into each threshold value for the selected row in Table 11. Table 13 shows that for all runs, approximately 84% of the searches found only one term in common between query and surrogate; 13.3% found two terms in common; and 2.2% found three terms in common.

6.5.2 Evaluation Based on Analysis of Document Representations.

The cost model in Chapter 4 suggested that there is a cost of creating, storing and retrieving a document representation. In Section 7.3 suggestions are made for using the cost of a representation as a guide to selecting which representation, among a number of alternatives, should be used against which to compare a query. As a step toward the evaluation of this cost approach to representation selection, the results of each simulation experiment were evaluated by recording the number of searches that resulted in a match between a query and each of the five document representations.

Another motivation for this type of analysis has to do with the composition of a document file. In a document file it would be expected that there would be a number of document representations associated with each document. That is, in addition to the bibliographic description of the document, an abstract, a set of index terms, etc. would also be present. However, it is not likely that all possible representations would be present for every document in the file. For example, in a document file, document A may only have associated with it a title

Table 13
Proportion of Searches Resulting in a Match between a Query and a Document Representation (Excluding Searches Yielding No Matches)

Run number	Proportion of searches resulting in a match between a query and a document representation at threshold level							
	1	2	3	4	5	6	7	>7
1	0.837	0.137	0.023	0.003	0.000	0.000	0.000	0.000
2	0.842	0.131	0.023	0.003	0.001	0.000	0.000	0.000
3	0.860	0.115	0.022	0.002	0.000	0.000	0.000	0.000
4	0.855	0.122	0.020	0.003	0.000	0.000	0.000	0.000
5	0.845	0.132	0.019	0.003	0.001	0.000	0.000	0.000
6	0.838	0.140	0.020	0.002	0.000	0.000	0.000	0.000
7	0.850	0.125	0.022	0.003	0.001	0.000	0.000	0.000
8	0.841	0.135	0.020	0.003	0.001	0.000	0.000	0.000
9	0.839	0.132	0.024	0.004	0.001	0.000	0.000	0.000
10	0.840	0.140	0.018	0.002	0.000	0.000	0.000	0.000
11	0.816	0.148	0.027	0.008	0.001	0.000	0.000	0.000
12	0.778	0.169	0.040	0.010	0.002	0.000	0.000	0.000
13	0.915	0.077	0.006	0.001	0.000	0.000	0.000	0.000
14	0.845	0.132	0.019	0.003	0.001	0.000	0.000	0.000
15	0.839	0.132	0.023	0.005	0.001	0.000	0.000	0.000
16	0.815	0.152	0.028	0.005	0.000	0.000	0.000	0.000
17	0.845	0.129	0.023	0.003	0.000	0.000	0.000	0.000
18	0.853	0.128	0.018	0.001	0.000	0.000	0.000	0.000
19	0.832	0.140	0.025	0.003	0.000	0.000	0.000	0.000
20	0.834	0.140	0.023	0.003	0.000	0.000	0.000	0.000
21	0.873	0.110	0.014	0.002	0.000	0.000	0.000	0.000
22	0.856	0.124	0.018	0.002	0.000	0.000	0.000	0.000

representation and an abstract. Document B may only have associated with it a title and a set of index terms. For document A an index term set does not exist. For document B an abstract does not exist. Likewise, not every information retrieval system would have all document representations in one file. For example, one retrieval system may have a file which consists only of document abstracts. Another retrieval system may have a file which is composed only of index term surrogates.

The retrieval system simulator generates a number of different representations for each document. By monitoring the effect of query characteristic changes on the number of searches resulting in a match between query and document representation, it should be possible to draw conclusions about which query characteristics to modify to interrogate a file containing only certain document representations. For the 22 experiments this data is summarized in Tables 14 and 15.

Table 15 is derived from Table 14 by dividing each element in Table 14 column 2 by 10,650, which is the number of searches performed against representation number one (142 document representations times 75 queries). (See Table 7.) Similarly, elements in columns 3 through 7 of Table 14 are divided by 10,650, 9,600, 9,225, 6,525, and 46,650, respectively, to obtain values for Table 15.

6.5.3 Summary of Experimental Results.

In Table 16 and Figure 26 an overall summary of the results of the experiments is presented. Both the table and the figure are derived from Table 15. Table 16 is divided into pairs of columns. If it is desired

Table 14
Number of Searches Matching a Specific
Document Representation

Run number	Number of searches resulting in a match between a query and document representation number					Total number of searches resulting in a match
	1	2	3	4	5	
1	3905	2279	1078	777	305	8344
2	3502	2007	990	751	238	7488
3	3607	2057	1075	693	285	7717
4	3631	2124	1044	745	257	7801
5	3200	1858	901	695	251	6905
6	3735	2106	1085	695	255	7876
7	3604	2082	1041	682	254	7663
8	3827	2223	1043	730	276	8099
9	3718	2202	1097	742	278	8037
10	4466	2400	1276	971	365	9478
11	4205	2499	1328	932	346	9301
12	5528	3411	1815	1192	476	12422
13	2296	1183	575	346	168	4568
14	3546	2139	1005	738	287	7715
15	3641	2152	1029	699	335	7856
16	3802	2187	1052	754	335	8130
17	3881	2148	1120	738	300	8187
18	3789	2099	1068	751	262	7969
19	3688	2231	1143	760	297	8119
20	3343	1916	943	739	266	7207
21	3708	2144	1012	748	260	7872
22	3773	2224	1129	718	299	8143

Table 15
Proportion of Searches Matching a Specific Document Representation

Run number	Proportion of searches resulting in a match between a query and document representation number					Total proportion of searches resulting in match
	1	2	3	4	5	
1	0.367	0.214	0.112	0.084	0.047	0.179
2	0.329	0.188	0.103	0.081	0.036	0.161
3	0.339	0.193	0.112	0.075	0.044	0.165
4	0.341	0.199	0.109	0.081	0.039	0.167
5	0.300	0.174	0.094	0.075	0.038	0.148
6	0.351	0.198	0.113	0.075	0.039	0.169
7	0.338	0.195	0.108	0.074	0.039	0.164
8	0.359	0.209	0.109	0.079	0.042	0.174
9	0.349	0.207	0.114	0.080	0.043	0.172
10	0.419	0.225	0.133	0.105	0.056	0.203
11	0.395	0.235	0.138	0.101	0.053	0.200
12	0.519	0.320	0.189	0.129	0.073	0.266
13	0.216	0.111	0.060	0.038	0.026	0.098
14	0.333	0.201	0.105	0.080	0.044	0.165
15	0.342	0.202	0.107	0.076	0.051	0.168
16	0.357	0.205	0.110	0.082	0.051	0.174
17	0.364	0.202	0.117	0.080	0.046	0.175
18	0.356	0.197	0.111	0.081	0.040	0.171
19	0.346	0.209	0.119	0.082	0.046	0.174
20	0.314	0.180	0.098	0.080	0.041	0.154
21	0.348	0.201	0.105	0.081	0.040	0.169
22	0.354	0.209	0.118	0.078	0.046	0.175

Table 16
Ranking of Experimental Runs
Based on Proportion of Searches Matching a
Specific Document Representation

Document representation 1		Document representation 2		Document representation 3		Document representation 4		Document representation 5		Overall ranking	
Rank	Run	Rank	Run	Rank	Run	Rank	Run	Rank	Run	Rank	Run
1	12	1	12	1	12	1	12	1	12	1	12
2	10	2	11	2	11	2	11	2	11	2	10
3	11	3	10	3	10	3	10	3	10	3	11
4	1	4	1	4	19	4	1	4	15	4	1
5	17	5	8	5	22	5	16	4	16	5	17
6	8	5	19	6	17	6	19	5	1	5	22
7	16	5	22	7	9	7	2	6	17	6	8
8	18	6	9	8	6	7	4	6	19	6	16
9	22	7	16	9	1	7	18	6	22	6	19
10	6	8	15	9	3	7	21	7	3	7	9
11	9	8	17	10	18	8	9	7	14	8	18
12	21	9	14	11	16	8	14	8	9	9	6
13	19	9	21	12	4	8	17	9	8	9	21
14	15	10	4	12	8	8	20	10	20	10	15
15	4	11	6	13	7	9	8	11	18	11	4
16	3	12	18	14	15	10	22	11	21	12	3
17	7	13	7	15	14	11	15	12	4	12	14
18	14	14	3	15	21	12	3	12	6	13	7
19	2	15	2	16	2	12	5	12	7	14	2
20	20	16	20	17	20	12	6	13	5	15	20
21	5	17	5	18	5	13	7	14	2	16	5
22	13	18	13	19	13	14	13	15	13	17	13

to determine which experiment resulted in the largest number of matches between document representation number 2 and a query file, then the pair of columns labeled 'Document representation 2' would be consulted in Table 16. The table indicates that experiment number 12 resulted in the largest number of matches for representation number 2 and thus ranked first. The table also shows that experiment number 13 resulted in the smallest number of matches between the representation and the query files, and thus ranked last.

Since there are tie values in Table 15, there are tie rankings in Table 16. For example, using representation number 2, experimental runs number 8, 19, and 22 all had rank 5. In cases where ties occur, the run numbers for tie rankings are listed in ascending order within the rank. The right hand pair of columns in Table 16 display the overall ranking of the experiments based on the total number of searches in a run that resulted in a match.

Figures 26(a) through 26(d) present the same data as Table 16 except in graph form. The left hand column of the figures show the initial rank of each experiment. In the second column is the number of the experiment having the specified rank when ordered by the results for representation number 1. As one proceeds across the page to the right and follows the same line, the rank of the experiment for each of the surrogates and the total is displayed. For example, in Figure 26(a) the experiment having rank 7 for the first representation is run 16 (E16). The rank of E16 for representation 2 is also 7. For representation 3 the rank drops to 11; for representation 4 it is up to rank 5; for representation 5 it is rank 4; and the overall ranking of the experiment is 6. Note that both

Figure 26(a)

Ranking of Experimental Runs

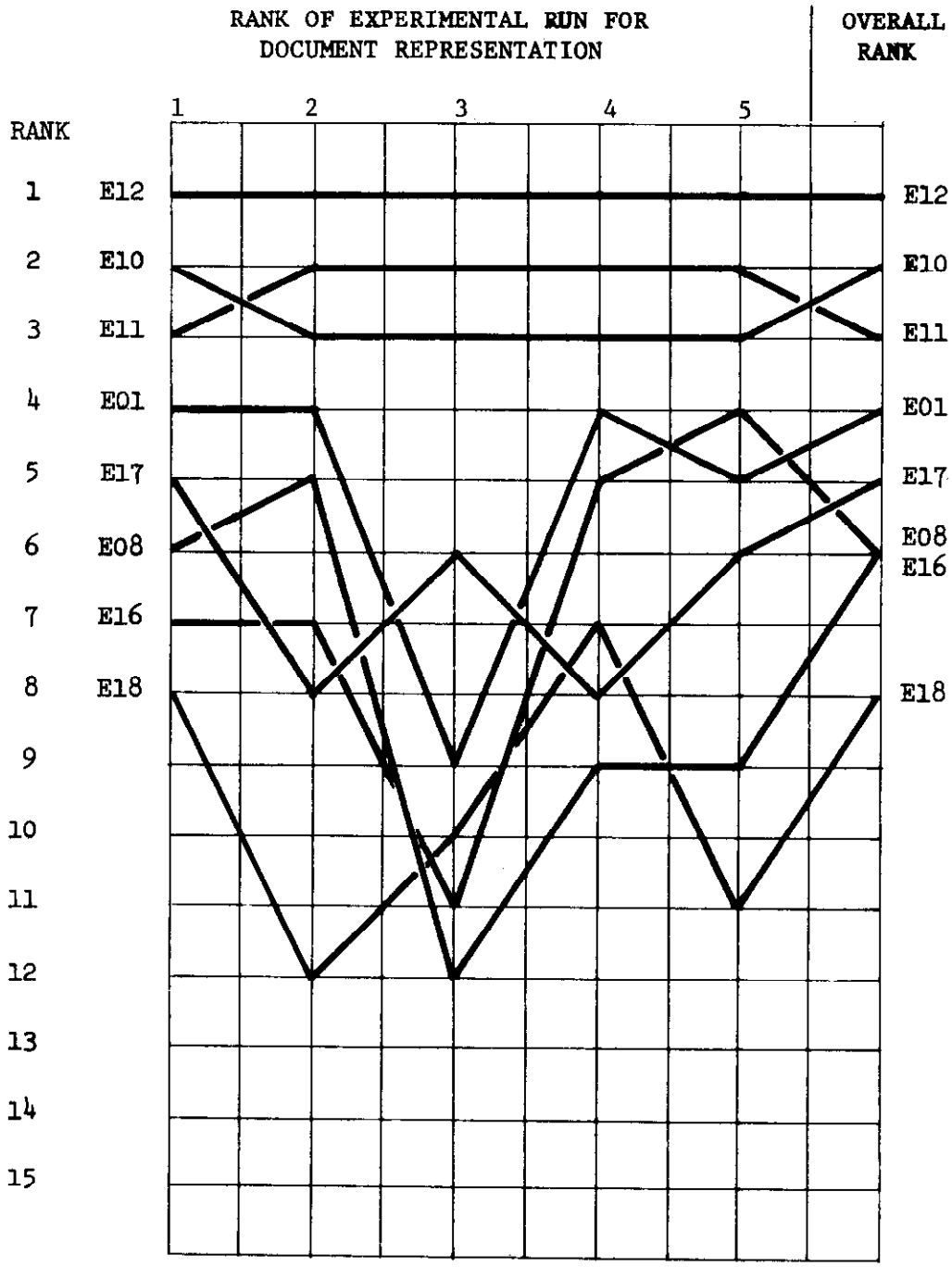


Figure 26(b)

Ranking of Experimental Runs

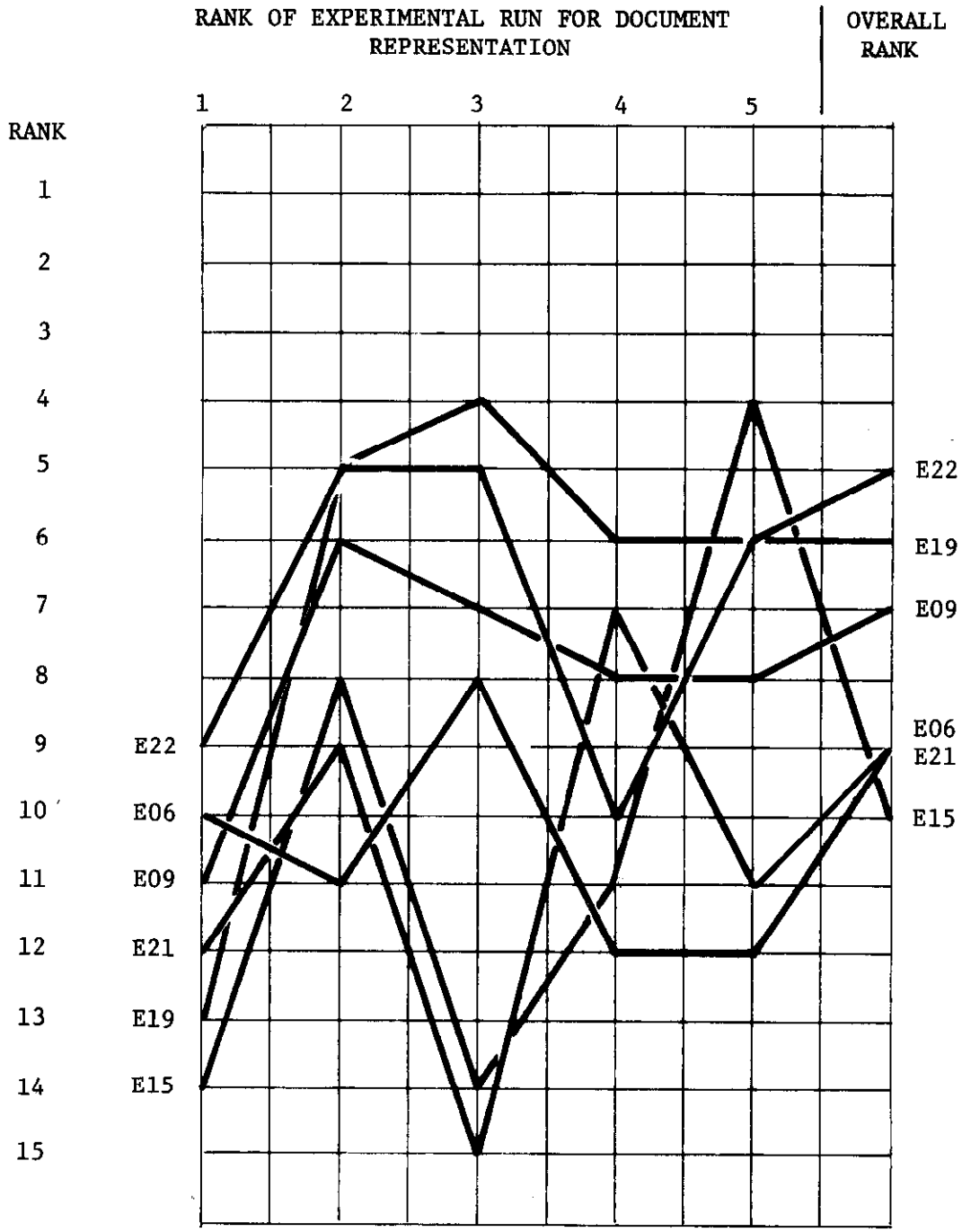


Figure 26(c)

Ranking of Experimental Runs

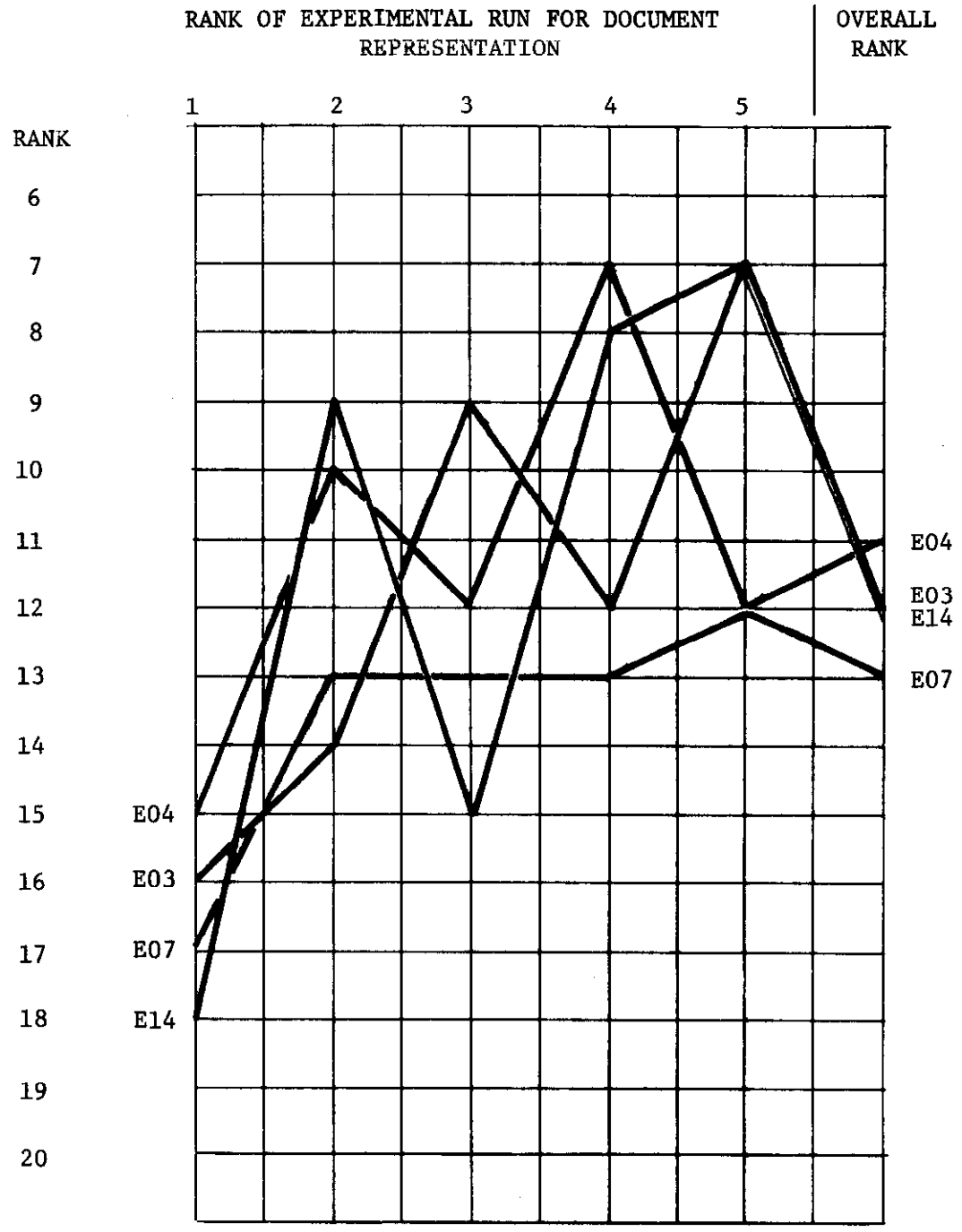


Figure 26(d)

Ranking of Experimental Runs

E08 and E16 have final rank 1. Figure 26 is divided into four parts for visual ease in following the rank patterns of each of the experiments.

Given the information in Table 16 and Figure 26, it is now possible to draw some conclusions regarding which parameter changes in the query files cause the greatest change in the number of matches between query and document representation. There are two criteria that can be used to determine whether one experiment produces better results than another. It may be decided that a better query file is one which results in a larger number of matches between document representations and queries. Alternatively it could be decided that a better query file is one which produces a minimum number of matches. In the evaluation that follows, the issue is not decided one way or another. It is believed that there will be situations in which a large number of matches will be desired and also situations in which the opposite will be true.

The single factor that caused the largest number of matches between queries and document representations is an increase in the number of words in a query. Experiment 12 (file Q12 with D01) is uniformly ranked highest for all surrogates. (In the retrieval system simulator, a query consists of a boolean disjunction of words.) Thus the more words that are added to a query, the more matches will result. Experiment 13 shows that the smaller the number of words in a pseudo-query, the fewer searches will result in a match.

In experiment 10 and experiment 11 the parameter that is changed is the standard deviation of the number of words in the query. In file Q10 the standard deviation of the query length is 1, and in Q11 it is 4. In both cases the mean query length is 5. (See Table 8.) Table 10 indicates that the <u>effect</u> of both of these changes is an increase in the mean query

length of the generated files Q10 and Q11 over the average query length for all files. Both experiments 10 and 11 result in consistently high rankings. It is presumed that the high rankings are due to the increased number of words in the queries in these files.

Experiment 1 produces high rankings for all cases except representation 3. One explanation for this high ranking is extremely interesting. There are a number of features that the document and query generation process have in common. They both use the concept of a starting fraction of terms, and they both use threshold and transition probabilities. Table 6 and Table 8 show that for those parameters that are common to both, there is a close similarity in values. While some of the rules and parameters are similar in the document and query generation routines, the process is a random one. That is, random variables are generated independently in each routine, and the words that are selected for initial inclusion in a document representation or query are selected randomly. Aside from the fact that the routines are independent, and random variables are employed, the experiment that produced a very high ranking was the one in which the document and query parameters were the most similar.

Runs E16 and E17 have as their only difference the mean number of subsets in each file. For Q16 there are 34 subsets and for Q17 there are 14 subsets. Both runs produce relatively high rankings. It is concluded that the mean number of subsets in a file does not materially affect the number of searches resulting in a match. Just as good a ranking could be obtained with a mid-range value of the mean number of subsets in a file. For example see the ranking for E01.

In query files Q08 and Q09 a change is made in the rule used for selecting a base query. In Q08 a query generated chronologically later in the sequence in a subset has a greater chance of being chosen as a base query. For Q09 a query generated earlier in the sequence will be more likely chosen as a base query. The simulation results indicate that experiment E08 has a higher rank (except for representation 3) than E09. Thus in the simulation model, in order to increase the number of matches between query and document representation, rules similar to the ones used to generate Q08 should be used rather than the rules used to generate Q09.

Taking another tack in the analysis, it is useful to determine if there are certain query file parameter changes that result in a high experimental ranking for only certain document representations. The motivation for this type of analysis is presented in Section 6.5.2. Experiments E02 and E20 are good examples of such a situation. In Q02 the parameter that is being changed is a high starting proportion of terms. The ranking is uniformly low except in the case of representation number 4. In Q20 high transition probabilities and a low starting proportion are present. Here again, the ranking based on representation 4 is high while the ranking based on all other document representations is low.

The two runs that produced consistently low rankings are E05 (in which Q05 has high threshold values) and E13 (in which Q13 has the shortest query length of any file). The pattern that is present in Figure 26(d) of the curves following the same slope is due to the fact that there are tie rankings for some of the surrogates. If the tie runs

were given a unique rank number, the lines for E05 and E13 would move horizontally across the page.

In summary, the experimental results from the simulation runs do not provide conclusive evidence of the superiority of one method or set of parameters for generating pseudo-query files. The data also indicates that there are, in general, very small differences between the results of one experiment and another.

Chapter 7

Summary and Conclusions

7. Summary and Conclusions

This dissertation has been concerned with exploring the structure of information retrieval systems and in particular with developing new methods for the evaluation of retrieval systems. Retrieval systems can develop in two ways. First it is possible to develop systems based on a theory which implies a complete understanding of the information acquisition processes involved (e.g. information transfer, the meaning of information, etc.). Alternatively, in the absence of such a theory, systems can be built and used as experimental tools in order to evaluate tentative hypotheses about the retrieval process.

In the absence of a theory of how retrieval systems should work, an ad hoc approach to designing systems has been pursued. The retrieval systems that are currently in use all have a number of components or sub-systems within them. These components, which have been previously discussed in Chapter 2, include modules for the analysis of the content of documents, rules for retrieving documents from a file, languages for communication between the user and the system, and methods for organizing files of information.

There are a large number of alternative methods that can be used to construct a retrieval system. An important question that must be analyzed is how to decide which of the alternatives is best. Traditionally such analysis has been done using the so-called measures of retrieval effectiveness. It was suggested in this dissertation that measures of retrieval effectiveness do not adequately evaluate the entire system and that a more comprehensive approach to the evaluation problem is in order. Two approaches have been presented for retrieval system

evaluation: a cost model and a simulation model.

7.1 Cost Model Evaluation.

The cost model that was developed divides the activities involved in the retrieval system operation in several ways. The first division involves an allocation of effort for a given search between the user of the retrieval system and the system itself. That is, either the user can spend time and effort in correctly specifying his query, understanding what kind of material is in the document file, how terms are related in the document file, etc., or a dialog between the user and the system can take place in which this information is established by negotiation. The negotiation process shifts some of the effort from the user to the system. Thus there is a trade off between the cost to the user and the cost to the system for the search. In addition to the division between user and system effort, the model divides the total time during which an interaction is taking place into three parts: pre-search activity, search activity and post-search activity. During the pre-search phase the user negotiates the query with the system; during the search phase the user waits while the system searches the file; and during the post-search phase the system displays the output for the user.

As with all models, the cost model of a literature searching system is a simplified description of the real situation. There are a number of deficiencies in the model. The performance measure that is used in determining the optimal allocation of effort between the user and the system is simplified. The measure only considers performance

as a function of user and system time. In all probability, a performance measure is much more complex than this cost model assumes. Another deficiency is that the model has not yet been verified with operating data. Aside from these problems it is believed that the framework that the model presents is a useful way of evaluating retrieval systems as well as a meaningful method for arriving at an optimal allocation between user and system resources.

7.2 Simulation Model Evaluation.

The second proposal that was made for evaluating retrieval systems was the use of simulation. A simulation model was developed to provide a framework for evaluation of retrieval systems. The simulation routines generate pseudo-documents and pseudo-queries to provide a data base to evaluate retrieval techniques. Pseudo-documents and pseudo-queries are used in the evaluation process rather than real documents and queries in order to exercise control over the characteristics of the documents and queries that are used in evaluating a specific retrieval system component. In this dissertation, the retrieval system simulator was used to analyze the way in which the quantity of output varied as a result of making changes in the way in which pseudo-queries were generated. These changes included such things as varying the proportion of terms that were randomly selected for inclusion in a query, varying the probability that a word that appeared in one query would appear in another query, and varying the number of words in a query.

Simulation has been used in many situations where analytic

solutions to problems can not be formulated. In concluding this dissertation it is important to ask whether simulation can be used to evaluate information retrieval systems. And it is also important to determine the adequacy of the simulation model described in Chapter 5 as a tool for retrieval system analysis.

There are a number of criteria that can be used to judge the adequacy of an evaluation technique. Specifically, the evaluation technique should be reliable in that it provides stable, dependable and accurate estimates of performance and valid in that it measures what it is that one desires should be measured. In addition, the methodology should be as comprehensive as possible. The technique should also be able to give clues as to how to change the system in order to improve performance. Further, a desirable characteristic of an evaluation tool is its ability to analyze a system at a minimum cost to the investigator, with a minimum investment in time for the analysis, and with maximum reliability in the results that are obtained from the analysis.

The simulation model as developed in Chapter 5 can not yet be considered as a good evaluative tool primarily because it is not yet comprehensive in its scope. The model does, however, provide a framework from which further development work can be performed.

Several other deficiencies of the model exist which prevent it being used without reservation. Most of these deficiencies occur because the process being modeled is not well understood. The first problem has to do with the rules that are used to generate documents and queries. The rules that are used in the simulation program are incomplete in that they do not take into account everything that is known about documents and queries, such as the fact that different kinds

of words convey different kinds and shades of meaning. The rules for document and query generation are also inadequate because no empirical studies have been conducted to establish that the relations between words can be characterized as they are in the model (i.e. by making probabilistic statements about whether words will be included or excluded from a document representation or a query . Thus given our present state of knowledge about the formation of documents and queries, the model makes only a first approximation at characterizing a very complex process.

A further deficiency of the simulation model is the method by which the results of the simulation are evaluated. The only method of evaluation that is used is to monitor the number of document representations that are found to match a query. Obviously the user of a retrieval system is concerned with more than just the quantity of material retrieved. The user is also concerned with the relevance of the material to his information need. Thus a serious deficiency of the simulation model is that it does not consider the issue of relevance. One reason why relevance is not considered is that there is no adequate theory of what the characteristics are of a function from which one could predict the relevance of a document to a user's need. Without such a theory it is very difficult to simulate the process. A further reason for the omission of a relevance function is the belief that it is possible to gain some insights into the retrieval system evaluation problem by only examining the quantity of material retrieved and not considering relevance. (The evidence for this belief is presented in Chapter 6.)

A third area in which the simulation model is inadquate is in the lack of alternative retrieval rules. The only retrieval rule that has

been implemented is an overlap measure of the extent to which a document representation and a query overlap. It would be useful and not at all difficult to implement other rules to see their effect on retrieval results.

Given these deficiencies it is possible to draw some general conclusions about this particular simulation model and about the use of simulation for evaluating retrieval systems. The simulation model of a retrieval system which is presented here, although not comprehensive and although limited because of the above deficiencies, was used to evaluate the way in which the quantity of output varied as a result of changing the rules used to generate various query files. The experimental design that was used to test the effect of changes in the query generation process allowed inferences to be made about the importance of various query characteristics. Evaluation of this limited model allowed prediction of ways in which the quantity of material retrieved varied relative to query characteristics.

The current simulation model does not permit predictions about the performance of retrieval systems in general. It does, however, provide a methodological framework from which a more comprehensive and complete model can be constructed. From the experience derived from the development of the retrieval system simulator two facts should be made explicit: the time required to develop this very simple model is great (more than a year) and the costs of developing the model are great both in terms of computer program development and program execution. (See Appendix 2.) In summary, then, the use of simulation as an evaluative tool appears to have great potential, but it should be realized that the time and cost to develop such a tool will be great.

The current model is considered as a limited but useful tool for retrieval system evaluation.

7.3 Future Research.

One of the more important goals in the future development of retrieval systems is to make the systems more adaptive to the needs of the user. [72]. There are a number of ways that this goal can be accomplished. A promising approach is to incorporate into retrieval systems methods for learning and feedback to improve system performance. [45].

It is believed that the cost model of a retrieval system presented in Chapter 4 and the simulation model of Chapter 5 can be integrated to provide an environment within which an adaptive search system for information retrieval can be evaluated. The cost model currently performs two functions. It determines an optimal allocation between system and user effort based on a performance measure and the cost of both the system's and the user's time. In addition, the model determines the total cost for searching and storing the documents in the file. For a given query or query subject category it would be possible to record in a matrix the cost to store and retrieve a specific document representation using a specific retrieval rule. Recording could be done for all possible retrieval rules. Given the matrix of cost quantities and a query, the retrieval system simulator could be designed to select a specific document representation to compare the query against based on the values in the cost matrix. The representation/retrieval rule

combination picked to compare the query against would be the one having the lowest cost entry in a given row of the cost matrix.

Once the user of the system had reviewed the documents retrieved by the literature searching system and determined the relevance of the documents to his information need, this information about relevance could be incorporated into the cost matrix in order to modify the cost entries. The effect of such a modification would be to create a new entry in each cell of the document representation/retrieval rule matrix reflecting both the cost and the benefit of a particular combination. The matrix could be continually modified to reflect the changing assessment by the user of a particular system strategy.

Appendix 1

Measures of Association

Appendix 1

Measures of Association

A common problem that is faced in many disciplines is to measure the similarity of objects. In the field of information retrieval the problem is to measure the similarity of the content of documents. If it is desired to use clustering techniques to group similar documents together, a measure of similarity must be computed in order to correctly assign a new document to the group to which it is most similar. If a query representation is being compared to a number of document representations, there needs to be some method of measuring the degree of match between each query representation-document representation pair. If associative searching is to be employed, an association matrix must be previously constructed. This requires that the extent to which terms in a document collection co-occur with one another be computed.

There are a number of different methods that can be used to compute similarity. It is possible to simply measure spatial distance between objects to determine similarity. The type of measure that is used will depend on the characteristics of objects and the way they are represented. Two possible measures for this are Euclidian distance and Hamming distance. A second category of similarity measures is correlation coefficients. The reader is referred to standard statistical texts such as [34] and [39] for a discussion of these measures.

The third type of similarity measure is the group employed most frequently in information retrieval systems. They are known as measures

of association. One of the most difficult problems facing a system designer is which of the many measures to employ in a retrieval system. To this date there is no clear answer. The work of Jones and Curtice [61], Kuhns [70], and Sokal and Sneath [122] offers some promise.

Sokal and Sneath have suggested several possible methods for evaluating the measures. [122]. It is possible to examine each association measure to determine its numerical bounds as each element in the measure goes to a limit. It is also possible to compute the expected value of each measure. Still other methods of evaluation include determining whether a weight can be attached to the presence of a particular representation used to compute similarity. Or alternatively it is possible to evaluate the measures on the basis of whether they take into account the dissimilarity as well as the similarity of objects. In Figure 27 a notation is presented which will be used to express in a standard form a number of association measures. The 2 x 2 table shows two 'properties' - property A and property B. Kuhns suggests a number of interpretations of a property, one of which is the following. [70, p. 33]. When a set of index terms is assigned to a document, the terms become properties of the document. Information about the presence or absence (or weight) of a pair of terms in all documents of a collection can be used to calculate the association between those terms.[6] The figure shows that each of the two properties can either be present or absent. Consider the case of index terms assigned to a document. For a collection of documents, there would be a total of 'a+b' documents in

---

6. Researchers in the field of numerical taxonomy refer to these 'properties' as operational taxonomic units. [122].

which index term A was present and 'a+c' documents in which index term B was present, out of a total of n documents. (See Figure 27.)

Kuhns has summarized a number of association measures, and they are presented in Table 17 using the standardized notation of Figure 27. [70]. In the formulas of Table 17 the quantity δ is given by

$$\delta = a - [(a+b)(a+c)] / n .$$

Sokal and Sneath also have summarized a number of measures. [122, p. 129-130]. Using the standardized notation, the formulas are presented in Table 18. In the table the association measures are classified according to whether or not the presence or absence of a match is accounted for. In addition the measures are classified according to how matching is weighted in the denominator of the formula. The name of the originator of the measure is shown above the measure in the table. Table 19 summarizes a number of other measures of association that have been suggested.

Figure 27

Standard Notation for Association Measures

		Property		
		B	Not B	
Property	A	a	b	a+b
	Not A	c	d	c+d
		a+c	b+d	n

Table 17

Summary of Association Measures - Kuhns

Symbol	Name	Formula
S	Area of seperation	$2\delta/n$
R	Rectangular distance	$\delta/\max[(a+b), (a+c)]$
P	Proportion of overlap	$\delta/[[1 - \frac{a}{(a+c)+(a+b)}][(a+c)+(a+b)/n]]$
W	Conditional probability on weak evidence	$\delta/\min[(a+b), (a+c)]$
U	First probability difference	$\delta/\max[(a+b)(1 - \frac{a+b}{n}), (a+c)(1 - \frac{a+c}{n})]$
V	Second probability difference	$\delta/\min[(a+b)(1 - \frac{a+b}{n}), (a+c)(1 - \frac{a+c}{n})]$
G	Angle between vectors	$\delta/\sqrt{(a+b)(a+c)}$
E	Modified proportion of overlap	$2\delta/[(a+b) + (a+c)]$
L	Linear Correlation	$\delta/\sqrt{(a+b)(a+c)(1 - \frac{a+b}{n})(1 - \frac{a+c}{n})}$
Y	Yule Coefficient of colligation	$n\delta/(ad+bc)^2$
Q	Yule auxiliary quantity	$n\delta/(ad+bc)$
I	Index of independence	$n\delta/[(a+b)(a+c)]$

Table 18

Summary of Association Measures - Sokal and Sneath

Denominator	Negative Matches in Numerator: Excluded	Negative Matches in Numerator: Included
Matched and unmatched pairs are equally weighted	(Jaccard, Sneath) $\frac{a}{a+b+c}$ (Russel and Rao) $\frac{a}{n}$	(Sokal and Michener) $\frac{a+d}{n}$ -
Matched pairs carry twice the weight of unmatched pairs	(Dice, Sørensen) $\frac{2a}{2a+b+c}$	$\frac{2(a+d)}{a+d+n}$
Unmatched pairs carry twice the weight of matched pairs	$\frac{a}{a+2(c+b)}$	(Rogers and Tanimoto) $\frac{a+d}{b+c+n}$
Unmatched pairs only	(Kulczynski) $\frac{a}{b+c}$	$\frac{a+d}{b+c}$
Marginal totals	(Kulczynski) $\frac{1}{2}[(a/a+c)+(a/a+b)]$	$\frac{1}{4}[(a/a+c)+(a/a+b)+(d/b+d)+(d/c+d)]$
Marginal totals	(Ochiani) $a/\sqrt{(a+c)+(a+b)}$	$ab/\sqrt{(a+c)(a+b)(b+d)(c+d)}$

Adapted from [122, pp. 129-130].

Table 19

Association Measures

Measure	Reference	
$\dfrac{a - \frac{(a+b)(a+c)}{n}}{\sqrt{\frac{(a+b)(a+c)}{n}}}$	Dennis	[33]
$\dfrac{a}{(a+b)+(a+c)-a}$	Doyle	[35]
$\dfrac{an}{(a+b)(a+c)}$	Giuliano	[46]
$\dfrac{a}{a+b}$	Maron and Kuhns	[86]
$\dfrac{a}{a+c}$	Maron and Kuhns	[86]
$a - \dfrac{(a+b)(a+c)}{n}$	Maron and Kuhns	[86]
$\log_{10} \dfrac{(\lvert an-(a+b)(a+c)\rvert - \frac{1}{2}n)^2 \, n}{(a+b)(a+c)[n-(a+b)][n-(a+c)]}$	Stiles	[126]

Appendix 2

The Simulation Programs

Appendix 2

The Simulation Programs

The computer programs used for the simulation experiments in this paper were written in PL/1 Version 4 using the Operating System (OS) on an IBM System/360 computer. Program development was performed on a 360/40. Production runs were made on a 360/40 and 360/91. Table 20 lists the approximate number of source statements for each program along with the number of routines comprising the program.

In Table 21 the mean execution time for each program and mean input/output count is given, along with the number of observations used to calculate the mean value. In the table, the timings for the query analysis phase of program execution are shown in three parts corresponding to three phases used in the analysis.

As noted above, program execution was performed on two different computers. It was also performed at two different computer centers. Due to different accounting methods it may not be possible to compare computing performance based on the figures supplied. It should be possible to make order of magnitude estimates of the differences in computing speed between the two machines.

All the programs are parameterized so that only control cards need be changed from one run to another.

Table 20

Program Size

Program name	Number of routines comprising the program	Approximate number of source statements
Thesaurus generation	7	400
Association matrix analysis	1	120
Document generation	11	600
Document analysis	3	350
Query generation	12	520
Query analysis	3	240
Search	1	110
Evaluation	1	450

Table 21

Program Execution Timings

Program name	Number of runs	Machine	Average execution time min:sec.hund.	Average input/ output count
Thesaurus generation	1	360/40	22:46.30	790
Association matrix analysis	1	360/40	3:57.20	956
Document generation	1	360/40	23:49.40	3514
Document analysis	1	360/40	7:34.70	16444
Query generation	15	360/40	2:23.00	987
Query generation	9	360/91	4.58	195
Query Anal-1	15	360/40	22.88	1780
Query Anal-1	9	360/91	0.82	202
Query Anal-2	13	360/40	43.28	2609
Query Anal-2	9	360/91	1.83	679
Query Anal-3	13	360/40	21.80	2823
Query Anal-3	8	360/91	0.98	657
Search	1	360/40	42:26.40	758
Search	23	360/91	77.36	867
Evaluation	1	360/40	7:03.50	5907
Evaluation	23	360/91	9.37	823

Bibliography

Bibliography

1. Abraham, C.T. "Graph Theoretic Techniques for the Organization of Linked Data," In: Manfred Kochen (Ed.) Some Problems in Information Science, Scarecrow Press, New York, 1965, pp. 229-251.

2. Abraham, C.T. "Techniques for Thesaurus Organization and Evaluation," In: Manfred Kochen (Ed.) Some Problems in Information Science, Scarecrow Press, New York, 1965, pp. 131-150.

3. Ackoff, Russell L. Scientific Method: Optimizing Applied Research Decisions, John Wiley, New York, 1962.

4. Baker, Frank B. "Information Retrieval Based Upon Latent Class Analysis," Journal of the ACM, 9:4(October 1962) 512-521.

5. Baker, Norman R. and Richard E. Nance. "The Use of Simulation in Studying Information Storage and Retrieval Systems," American Documentation, 19:4(October 1968) 363-370.

6. Ball, Geoffrey. A Comparison of Some Cluster-Seeking Techniques, Stanford Research Institute, Report RADC TR-66-514, Stanford, California, November 1966.

7. Bar-Hillel, Yehoshua. "Theoretical Aspects of the Mechanization of Literature Searching," In: Language and Information: Selected Essays on their Theory and Application, Addison-Wesley, Reading, Massachusetts, 1964, pp. 330-364.

8. Baxendale, Phyllis B. "Machine-Made Index for Technical Literature-An Experiment," IBM Journal of Research and Development, 2:4(October 1958) 354-361.

9. Baxendale, Phyllis B. and Dan C. Clarke. Documentation for an Economical Program for the Limited Parsing of English: Lexicon, Grammar, and Flowcharts, Research Report RJ-386, IBM San Jose Research Laboratory, San Jose, California, August 16, 1966.

10. Becker, Joseph and Robert M. Hayes. Information Storage and Retrieval: Tools, Elements, Theories, John Wiley, New York, 1963.

11. Berul, Lawrence H. "Document Retrieval," In: Carlos A. Cuadra (Ed.) Annual Review of Information Science and Technology, 4, Britannica, Chicago, 1969, pp. 203-227.

12. Black, Stanley W. "Library Economics," In: Douglas M. Knight and E. Shepley Nourse (Eds.) Libraries At Large: Tradition, Innovation, and the National Interest, R.R. Bowker Co., New York, 1969, pp. 590-599.

13. Blunt, Charles R. An Information Retrieval System Model, Technical Report 352.14-R-1, HRB-Singer, Inc., Science Park, State College, Pa., October 1965, AD 623 590.

14. Bobrow, Daniel G. "Syntactic Theories in Computer Implementations," In: Harold Borko (Ed.) Automated Language Processing, John Wiley, New York, 1967, pp. 215-251.

15. Borko, Harold. A Research Plan For Evaluating The Effectiveness of Various Indexing Systems, FN 5649/000/01, System Development Corp., Santa Monica, Calif., July 10, 1961, AD 278 624.

16. Borodin, A., L. Kerr and F. Lewis. "Query Splitting in Relevance Feedback Systems," Information Storage and Retrieval, Report ISR-14, Cornell University, Ithaca, New York, October 1968, pp. XII-1 to XII-20.

17. Bourne, Charles P. Methods of Information Handling, John Wiley, New York, 1963.

18. Bourne, Charles P. and Donald F. Ford. "Cost Analysis and Simulation Procedures for the Evaluation of Large Information Systems, American Documentation, 15:2(April 1964) 142-149.

19. Buchholz, Werner. "File Organization and Addressing," IBM Systems Journal, 2(June 1963) 86-111.

20. Bush, Robert R. and Frederick Mosteller. Stochastic Models for Learning, John Wiley, New York, 1955.

21. Carnap, Rudolf. Logical Foundations of Probability, Second Edition, University of Chicago Press, Chicago, 1962.

22. Chapin, Ned. "A Comparison of File Organization Techniques," Proceedings of 24 th National Conference, Association For Computing Machinery, No. P-69, 1969, pp. 273-283.

23. Churchman, C. West. "An Analysis of the Concept of Simulation," In: Austin C. Hoggatt and Frederick E. Balderston (Eds.) Symposium on Simulation Models: Methodology and Applications to the Behavioral Sciences, South-Western Publishing Co., Cincinnati, Ohio, 1963, pp. 1-12.

24. Churchman, C. West. The Systems Approach, Delacorte Press, New York, 1968.

25. Cleverdon, Cyril W. "The Testing and Evaluation of the Operating Efficiency of the Intellectual Stages of Information Retrieval Systems," In: Pauline Atherton (Ed.) Classification Research: Proceedings of the Second International Study, Elsinore, Denmark, 14-18 September 1964, Munksgaard, Copenhagen, 1965, pp. 445-465. (International Federation for Documentation Publication No. 370).

26. Cooper, William S. "Expected Search Length: A Single Measure of Retrieval Effectiveness Based on the Weak Ordering Action of Retrieval Systems," American Documentation, 19:1(January 1968) 30-41.

27. Cuadra, Carlos A. and Robert V. Katter. Experimental Studies of Relevance Judgments: Final Report, 3 vols, TM-3520/001/00, TM-3520/002/00 and TM-3520/003/00, System Development Corp., Santa Monica, Calif., June 30, 1967.

28. Curtice, Robert M. and Paul E. Jones. "Distributional Constraints and the Automatic Selection of an Indexing Vocabulary," In: American Documentation Institute Proceedings of the American Documentation Institute Annual Meeting: Levels of Interaction Between Man and Information, Volume 4, Thompson Book Co., Washington, D.C., 1967, pp. 152-156.

29. Dale, A.G. Retrieval System Experimentation and Evaluation at LRC, Linguistics Research Center, University of Texas, Austin, Texas, July 1965, PB 168 284.

30. Dale, A.G. and N. Dale. Clumping Techniques and Associative Retrieval, Linguistics Research Center, University of Texas, Austin, Texas, March 1964, PB 166 121.

31. Dale, A.G. and N. Dale. "Some Clumping Experiments for Associative Document Retrieval," American Documentation, 16:1(January 1965) 5-9.

32. Damerau, Fred J. "An Experiment in Automatic Indexing," American Documentation, 16:4(October 1965) 283-289.

33. Dennis, Sally F. "The Construction of a Thesaurus Automatically from a Sample of Text," In: [125], pp. 61-148.

34. Dixon, Wilfrid J. and Frank J. Massey, Jr. Introduction to Statistical Analysis, Third Edition, McGraw-Hill Book Co., New York, 1969.

35. Doyle, Lauren B. "Indexing and Abstracting by Association," American Documentation, 13:4(October 1962) 378-390.

36. Earl, L.L. Annual Report: Automatic Indexing and Abstracting Part 1, M-21-66-1, Lockheed Missles and Space Co., March 1966.

37. Edmundson, H.P. "New Methods in Automatic Extracting," Journal of the ACM, 16:2(April 1969) 264-285.

38. Edmundson, H.P., V.A. Oswald, Jr., and R.E. Wyllys. Automatic Indexing and Abstracting of the Contents of Documents, Planning Research Corp., Los Angeles, Calif., October 31, 1959, AD 231 606.

39. Ezekiel, Mordecai and Karl A. Fox. Methods of Correlation and Regression Analysis: Linear and Curvilinear, Third Edition, John Wiley, New York, 1959.

40. Fairthorne, Robert A. "Basic Parameters of Retrieval Tests," In: American Documentation Institute Proceedings of the 27 th Annual Meeting: Parameters of Information Science, Volume I, Spartan Books, Washington, D.C., 1964, pp. 343-345.

41. Feigenbaum, Edward A. and Julian Feldman (Eds.). Computers and Thought, McGraw-Hill Book Co., New York, 1963.

42. Fels, E.M. "Evaluation of the Performance of an Information-Retrieval System by Modified Mooers Plan," American Documentation, 14:1(January 1963) 28-34.

43. Forrester, Jay W. Industrial Dynamics, MIT Press, Cambridge, Mass., 1961.

44. Fried, J.B., B.C. Landry, D.M. Liston, Jr., B.P. Price, R.C. Van Buskirk and D.M. Wachsberger. Index Simulation Feasibility and Automatic Document Classification, Technical Report 68-4, Computer and Information Science Research Center, Ohio State University, Columbus, Ohio, October 1968, PB 182 597.

45. Fu, K.S. Learning Control Systems: Review and Outlook, Report TR-EE 69-41, School of Electrical Engineering, Purdue Univ., Lafayette, Indiana, October 1969.

46. Giuliano, Vincent E. "The Interpretation of Word Associations," In: [125], pp. 25-32.

47. Giuliano, Vincent E. and Paul E. Jones. "Linear Associative Information Retrieval," In: Paul W. Howerton and David C. Weeks (Eds.) Vistas In Information Handling, Volume I, Spartan Books, Washington, D.C., 1963, pp. 30-54.

48. Goffman, William and Vaun A. Newill. "A Methodology for Test and Evaluation of Information Retrieval Systems," Information Storage and Retrieval, 3:1(August 1966) 19-25.

49. Good, I.J. "The Decision-Theory Approach to the Evaluation of Information-Retrieval Systems," Information Storage and Retrieval, 3:2(April 1967) 31-34.

50. Gotlieb, C.C. and S. Kumar. "Semantic Clustering of Index Terms," Journal of the ACM, 15:4(October 1968) 493-513.

51. Gurk, Herbert M. and Jack Minker. "Storage Requirements for Information Handling Centers," Journal of the ACM, 17:1(January 1970) 65-77.

52. Haas, Warren J. "Columbia University Libraries: A Description of a Project to Study the Research Library as an Economic System," In: Association of Research Libraries Minutes of the Sixty-Third Meeting, Chicago, January 26, 1964, pp. 40-46.

53. Herdan, Gustav. Quantitative Linguistics, Butterworth, London, 1964.

54. Hertz, D.B., B.E. Wynne, Jr., J.J. Corrigan, K.H. Schaffir, L.A. Moody, and S.B. Littauer. Research Study of Criteria and Procedures for Evaluating Scientific Information Retrieval Systems, Contract NSF-C218, Arthur Anderson and Co., New York, March 1962, AD 273 115.

55. Hinojosa, Jorge. Analysis and Design of File Structures, Using Indexed Sequential Organization, Technical Memo No. 6, Institute of Library Research, University of Calif., Berkeley, June 16, 1969.

56. Ide, Eleanor. "New Experiments in Relevance Feedback," Information Storage and Retrieval, Report ISR-14, Cornell University, Department of Computer Science, Ithaca, New York, October 1968, pp. VIII-1 to VIII-30.

57. IBM Corporation. Bibliography on Simulation, Form No. 320-0924-0, White Plains, New York, 1966.

58. IBM Corporation. Introduction to IBM System/360 Direct Access Storage Devices and Organization Methods, Form No. C20-1649, White Plains, New York, 1966.

59. Irwin, J.O. "The Place of Mathematics in Medical and Biological Statistics," Journal of the Royal Statistical Society, Series A, 126:1(1963) 1-41, Discussion pp. 42-44.

60. Jain, Aridaman K. A Statistical Study of Book Use, PhD Dissertation, Purdue University, Lafayette, Indiana, 1967, PB 176 525.

61. Jones, Paul E. and Robert M. Curtice. "A Framework for Comparing Term Association Measure," American Documentation, 18:3(July 1967) 153-161.

62. Jones, Paul E., Vincent E. Giuliano and Robert M. Curtice. Papers on Automatic Language Processing: Selected Collection Statistics and Data Analysis, Report ESD-TR-67-202, Volume I, Arthur D. Little, Inc., Cambridge, Mass., February 1967.

63. Kasher, Asa. "Data-Retrieval by Computer: A Critical Survey," In: Manfred Kochen (Ed.) The Growth of Knowledge: Readings on Organization and Retrieval of Information, John Wiley, New York, 1967, pp. 292-324.

64. Katter, Robert V. "Design and Evaluation of Information Systems," In: Carlos A. Cuadra (Ed.) Annual Review of Information Science and Technology, 4, Britannica, Chicago, 1969, pp. 31-70.

65. Katter, Robert V. "The Influence of Scale Form on Relevance Judgments," Information Storage and Retrieval, 4:1(March 1968) 1-11.

66. Kay, Martin. A Parsing Program for Computational Grammars, Report RM-4283-PR, Rand Corp., Santa Monica, Calif., 1964.

67. Keith, Nathan R., Jr. "A General Evaluation Model for an Information Storage and Retrieval System," Journal of the American Society for Information Science, 21:4(July-August 1970) 237-239.

68. Kessler, M.M. "Bibliographic Coupling between Scientific Papers," American Documentation, 14:1(January 1963) 10-25.

69. Kučera, Henry and W. Nelson Francis. Computational Analysis of Present-Day American English, Brown University Press, Providence, Rhode Island, 1967.

70. Kuhns, J.L. "The Continuum of Coefficients of Association," In: [125], pp. 33-39.

71. Kuno, Susumo and A.G. Oettinger. "Multiple-Path Syntactic Analyzer," In: Cicely M. Popplewell (Ed.) Information Processing 1962: Proceedings of IFIP Congress 62, Munich, North-Holland Publishing Co., Amsterdam, 1963, pp. 306-312.

72. Kunz, W. and H. Rittel. "Zur Logik von Forschung und Dokumentation; Einige Strategien für den Entwurf 'Freundlicher' Informationssysteme für die Wissenschaftliche Forschung (I)," Naturwissenschaften, 55:8(1968) 358-361.

73. Lancaster, Frederick Wilfrid. Information Retrieval Systems: Characteristics, Testing, and Evaluation, John Wiley, New York, 1968.

74. Landau, Herbert B. "The Cost Analysis of Document Surrogation: A Literature Review," American Documentation, 20:4(October 1969) 302-310.

75. Leimkuhler, Ferdinand F. and Michael D. Cooper. Analytical Planning for University Libraries, Paper P-1, Office of the Vice President - Planning and Analysis, University of California, Berkeley, Calif., January 1970, ED 040 729.

76. Leimkuhler, Ferdinand F. and Michael D. Cooper. Cost Accounting and Analysis for University Libraries, Paper P-2, Office of the Vice President - Planning and Analysis, University of California, Berkeley, Calif., January 1970, ED 040 728.

77. Lefkovitz, David. File Structures for On-Line Systems, Spartan Books, New York, 1969.

78. Lewis, P.A.W., P.B. Baxendale, and J.L. Bennett. "Statistical Discrimination of the Synonymy/Antonymy Relationship Between Words," Journal of the ACM, 14:1(January 1967) 20-44.

79. Lister, Winston C. Least Cost Decision Rules for the Selection of Library Materials for Compact Storage, PhD Dissertation, Purdue University, Lafayette, Indiana, January 1967, PB 174 441.

80. Luhn, Hans P. "The Automatic Creation of Literature Abstracts," IBM Journal of Research and Development, 2:2(April 1958), 159-165.

81. Lyons, John. Introduction to Theoretical Linguistics, Cambridge University Press, London, 1968.

82. MacNaughton-Smith, P. Some Statistical and Other Numerical Techniques for Classifying Individuals, Her Majesty's Stationery Office, London, 1965. (Studies in the Causes of Delinquency and the Treatment of Offenders, No. 6).

83. MacQueen, James B. Some Methods for Classification and Analysis of Multivariate Observations, Working Paper No. 96, Western Management Science Institute, University of California, Los Angeles, March 1966.

84. March, James G. and Herbert A. Simon. Organizations, John Wiley, New York, 1958.

85. Maron, M.E. "Automatic Indexing: An Experimental Inquiry," Journal of the ACM, 8:3(July 1961) 404-417.

86. Maron, M.E. and J.L. Kuhns. "On Relevance, Probabilistic Indexing and Information Retrieval," Journal of the ACM, 7:3(July 1960) 216-244.

87. Maron, M.E. and R.M. Shoffner. The Study of Context: An Overview, NSF Grant No. GN643, Institute of Library Research, University of California, Berkeley, January 1969.

88. Maron, M.E., A.J. Humphrey and J.C. Meredith. An Information Processing Laboratory for Education and Research in Library Science: Phase I, Institute of Library Research, University of California, Berkeley, July 1969.

89. Meadow, Charles T. The Analysis of Information Systems: A Programmer's Introduction to Information Retrieval, John Wiley, New York, 1967.

90. Minsky, Marvin (Ed.). Semantic Information Processing, MIT Press, Cambridge, Massachusetts, 1968.

91. Montgomery, Christine A. "Automated Language Processing," In: Carlos A. Cuadra (Ed.) Annual Review of Information Science and Technology, 4, Britannica, Chicago, 1969, pp. 145-174.

92. Montgomery, Christine A., R.M. Worthy, and G. Reitz. Optical Character Reader Applications Study, Final Technical Report covering period January 24, 1967 - August 24, 1968, Rome Air Development Center, Griffiss Air Force Base, New York, August 1968, AD 504 342L.

93. Morse, Philip M. Library Effectiveness: A Systems Approach, MIT Press, Cambridge, Massachusetts, 1968.

94. Nance, Richard E. Strategic Simulation of a Library/User/Funder System, PhD Dissertation, Purdue University, Lafayette, Indiana, June 1968.

95. Naylor, Thomas H., Joseph L. Balintfy, Donald S. Burdick and Kong Chu. Computer Simulation Techniques, John Wiley, New York, 1968.

96. Needham, R.M. "A Method for Using Computers in Information Classification," In: Cicely M. Popplewell (Ed.) Information Processing 1962: Proceedings of IFIP Congress 62, North-Holland Publishing Co., Amsterdam, 1963, pp. 284-287.

97. Needham, R.M. "Applications of the Theory of Clumps," Mechanical Translation, 8:3 and 4(June and October 1965) 113-127.

98. Parker-Rhodes, A.F. and R.M. Needham. The Theory of Clumps: A New Concept of Classification and Selection, Cambridge Language Research Unit, Cambridge, England, Report ML 126, February 1960, PB 166 804.

99. Patrick, Ruth J. and Michael D. Cooper. "A User Study," In: Laura Gould, Deborah D. Barrett, and Ralph M. Shoffner An Experimental Inquiry into Context Information Processing, NSF Grant No. GN 643, Institute of Library Research, University of California, Berkeley, January 1969, pp. 79-91.

100. Penner, Rudolf J. "The Practice of Charging Users for Information Services: A State of the Art Report," Journal of the American Society for Information Science, 21:1(January-February 1970) 67-74.

101. Perry, James W., Allen Kent, and Madeline M. Berry. Machine Literature Searching, Western Reserve University Press, Cleveland, Ohio, Interscience Publishers, New York, 1956.

102. Prywes, N.S. and H.J. Gray. "The Organization of a Multilist-Type Associative Memory," IEEE Transactions on Communications and Electronics, No. 68(September 1963) 488-492.

103. Rees, Alan M. The Evaluation of Retrieval Systems, Comparative Systems Laboratory Technical Report CSL: TR-5, Center for Documentation and Communication Research, Western Reserve Univ., Cleveland, Ohio, July 1965.

104. Reilly, Kevin D. User Determination of Library Request Presentation: A Simulation, NSF Grant GN-422, Institute of Library Research, University of California, Los Angeles, March 31, 1968.

105. Rettenmayer, John W. The Effect of File Ordering on Retrieval Cost, PhD Dissertation, University of California, Los Angeles, Western Management Science Institute Working Paper No. 156, December 1969.

106. Riddle, W., T. Horwitz, and R. Dietz. "Relevance Feedback in an Information Retrieval System," Information Storage and Retrieval, Report ISR-11, Cornell University, Department of Computer Science, Ithaca, New York, June 1966, pp. VI-1 to VI-71.

107. Rocchio, Joseph J., Jr. "Document Retrieval Systems - Optimization and Evaluation," PhD Dissertation, Information Storage and Retrieval, Report ISR-10, Harvard University, Cambridge, Mass., March 1966, PB 170 702.

108. Rogers, David J. and Taffee T. Tanimoto. "A Computer Program for Classifying Plants," Science, 132:3434(October 21, 1960) 1115-1118.

109. Rubenstein, Herbert and John B. Goodenough. "Contextual Correlates of Synonymy," Communications of the ACM, 8:10(October 1965) 627-633.

110. Salton, Gerard. "Automated Language Processing," In: Carlos A. Cuadra (Ed.) Annual Review of Information Science and Technology, 3, Britannica, Chicago, 1968, pp. 169-199.

111. Salton, Gerard. Automatic Information Organization and Retrieval, McGraw-Hill, New York, 1968.

112. Salton, Gerard. "The Evaluation of Automatic Retrieval Procedures - Selected Test Results Using the SMART System," American Documentation, 16:3(July 1965) 209-222.

113. Senko, Michael E. "File Organization and Management Information Systems," In: Carlos A. Cuadra (Ed.) Annual Review of Information Science and Technology, 4, Britannica, Chicago, 1969, pp. 111-143.

114. Senko, Michael E. Formatted File Organization Techniques: Final Report, Contract AF 30(602)-4088 to Rome Air Development Center, IBM Corp., Thomas J. Watson Research Center, Yorktown Heights, New York, May 16, 1967.

115. Senko, Michael E., Vincent Y. Lum, and Philip J. Owens. "A File Organization Evaluation Model (FOREM)," In: A.J.H. Morell (Ed.) Information Processing 68, Proceedings of IFIP Congress 1968, Volume I, North-Holland Publishing Co., Amsterdam, 1969, pp. 514-519.

116. Shapiro, Robert M. et. al. A Handbook on File Structuring, Applied Data Research, Inc., New York, September 1969, AD 697 025.

117. Shubik, Martin. "Simulation of the Industry and the Firm," The American Economic Review, L:5(December 1960) 908-919.

118. Simmons, Robert F. "Natural Language Question-Answering Systems: 1969," Communications of the ACM, 13:1(January 1970) 15-30.

119. Simon, Herbet A. Administrative Behavior: A Study of Decision-Making Processes in Administrative Organization, Macmillan, New York, 1957.

120. Smith, Steven F. and Ralph M. Shoffner. A Comparative Study of Mechanized Search Languages, NSF Grant No. GN643, Institute of Library Research, University of California, Berkeley, January 1969.

121. Sokal, Robert R. "Numerical Taxonomy," Scientific American, 215:6(December 1966) 106-116.

122. Sokal, Robert R. and Peter H.A. Sneath. Principles of Numerical Taxonomy, W.H. Freeman, San Francisco, 1963.

123. Sparck Jones, Karen. "Experiments in Semantic Classification," Machine Translation, 8:3 and 4(June and October 1965) 97-112.

124. Stevens, Mary E. Automatic Indexing: A State-of-the-Art Report, National Bureau of Standards Monograph 91, U.S. Government Printing Office, Washington, D.C., March 30, 1965.

125. Stevens, Mary E., Vincent E. Giuliano and Laurence B. Heilprin (Eds.). Statistical Association Methods for Mechanized Documentation, National Bureau of Standards Miscellaneous Publication 269, Washington, D.C., December 15, 1965.

126. Stiles, H. Edmund. "The Association Factor in Information Retrieval," Journal of the ACM, 8:2(April 1961) 271-279.

127. Stone, Don C. Word Statistics in the Generation of Semantic Tools for Information Systems, University of Pennsylvania, Philadelphia, Pa., December 1967, AD 664 915.

128. Swanson, Don R. "Searching Natural Language Text by Computer," Science, 132:3434(October 21, 1960) 1099-1104.

129. Swets, John A. "Effectiveness of Information Retrieval Methods," American Documentation, 20:1(January 1969) 72-89.

130. Swets, John A. "Information Retrieval Systems," Science, 141:3577 (July 19, 1963) 245-250.

131. Tague, Jean. "Association Trails," In: Allen Kent and Harold Lancour (Eds.) Encyclopedia of Library and Information Science, Volume II, Marcel Dekker, New York, pp. 55-81.

132. Tanimoto, Taffee T. An Elementary Mathematical Theory of Classification and Prediction, IBM Corp., New York, New York, November 17, 1958.

133. United States Educational Resources Information Center. Thesaurus of ERIC Descriptors, Second Edition, April 1969, U.S. Government Printing Office, Washington, D.C., 1969.

134. United States National Science Foundation. Summary of Study Conference on Evaluation of Document Searching Systems and Procedures, Washington, D.C., February 10, 1965.

135. Van Horn, Richard L. "Validation of Simulation Results," Management Science, 17:5(January 1971) 247-258.

136. Verhoeff, J., W. Goffman, and Jack Belzer. "Inefficiency of the Use of Boolean Functions for Information Retrieval Systems," Communications of the ACM, 4:12(December 1961) 557-558 and 594.

137. Ward, Joe H., Jr. and Marion E. Hook. "Application of a Hierarchical Grouping Procedure to a Problem of Grouping Profiles," Educational and Psychological Measurement, 23:1(Spring 1963) 69-81.

138. Williams, Gordon. Library Cost Models: Owning versus Borrowing Serial Publications, NSF Grant No. GN532, November 1968, PB 182 304.

139. Williams, John H., Jr. "A Discriminant Method for Automatically Classifying Documents," AFIPS Conference Proceedings: 1963 Fall Joint Computer Conference, 24, Spartan Books, Baltimore, Maryland, 1963, pp. 161-166.

140. Wilson, Patrick. Two Kinds of Power: An Essay on Bibliographical Control, University of California Press, Berkeley and Los Angeles, 1968.

141. Wyllys, Ronald E. "Extracting and Abstracting by Computer," In: Harold Borko (Ed.) Automated Language Processing, John Wiley, New York, 1967, pp. 127-179.

142. USA Standard for a Format for Bibliographic Information Interchange on Magnetic Tape," Journal of Library Automation, 2:2(June 1969) 53-65.